Visual Odometry Using Heterogeneous Cameras for Simultaneous Localization and Mapping for Autonomous Vehicles

Abanob Soliman

université PARIS-SACLAY

Visual Odometry Using Heterogeneous Cameras for Simultaneous Localization and Mapping for Autonomous Vehicles

Odométrie Visuelle par Association de Caméras Hétérogènes. Application à la Localisation et à la Cartographie Simultanée des Véhicules Autonomes

Thèse de doctorat de l'Université Paris-Saclay

École doctorale n°580 Sciences et Technologies de l'Information et de la Communication (STIC)
Spécialité de doctorat: Sciences du Traitement du Signal et des Images

Graduate school: Sciences de l'Ingénierie et des Systèmes
Référent: Université d'Évry Val d'Essonne

Thèse préparée dans l'unité de recherche **IBISC (Université Paris-Saclay, Univ Evry)**, sous la direction de **Samia BOUCHAFA-BRUNEAU**, Professeure des Universités, la co-direction de **Dro Désiré SIDIBÉ**, Professeur des Universités, le co-encadrement de **Fabien BONARDI**, Maître de Conférences.

Thèse soutenue à Paris-Saclay, le 5 Octobre 2023, par

Abanob SOLIMAN

Composition du Jury
Membres du jury avec voix délibérative

Vincent FRÉMONT Professeur des Universités, Ecole Centrale de Nantes	Président
Rémi BOUTTEAU Professeur des Universités, Université de Rouen Normandie	Rapporteur & Examinateur
Cédric DEMONCEAUX Professeur des Universités, Université de Bourgogne	Rapporteur & Examinateur
Michèle GOUIFFÈS Maîtresse de Conférences, HDR, Polytech Université Paris-Saclay	Examinatrice

THESE DE DOCTORAT

NNT : 2023UPAST119

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Odométrie visuelle par association de caméras hétérogènes. Application à la localisation et à la cartographie simultanée des véhicules autonomes

Mots clés: Odométrie visuelle, Mise en correspondance d'images, Caméras événementielles, Stéréovision

Résumé: Cette thèse de doctorat aborde les défis de la fusion de capteurs et de la localisation et de la cartographie simultanées (SLAM) pour les systèmes autonomes, en se concentrant spécifiquement sur les véhicules terrestres autonomes (AGV) et les micro-véhicules aériens (MAV) naviguant dans des environnements dynamiques et à grande échelle. La thèse présente une gamme de solutions innovantes pour améliorer la performance et la fiabilité des systèmes SLAM à travers cinq chapitres méthodologiques.
Le chapitre d'introduction établit la motivation de la recherche, en soulignant les défis et les limitations de l'odométrie visuelle utilisant des caméras hétérogènes. Il décrit également la structure de la thèse et fournit une analyse approfondie de la littérature pertinente. Le deuxième chapitre présente IBISCape, une référence simulée pour valider les systèmes SLAM haute fidélité basés sur le simulateur CARLA. Le troisième chapitre présente une nouvelle méthode basée sur l'optimisation pour calibrer une configuration visuelle-inertielle RGB-D-IMU, validée par des expériences approfondies sur des séquences réelles et simulées. Le quatrième chapitre propose une approche d'estimation d'état optimale linéaire pour les MAV afin d'obtenir une localisation de haute précision avec un retard minimal du système.
Le cinquième chapitre présente le système DH-PTAM pour un suivi et une cartographie parallèles robustes dans des environnements dynamiques utilisant des images stéréo et des flux d'événements. Le sixième chapitre explore de nouvelles frontières dans le domaine du SLAM dense à l'aide de caméras Event, présentant une nouvelle approche de bout en bout pour les événements hybrides et le système SLAM dense à nuages de points. Le septième et dernier chapitre résume les contributions et les principaux résultats de la thèse, en mettant l'accent sur les progrès réalisés dans la fusion de capteurs hétérogènes multimodaux pour les systèmes autonomes naviguant dans des environnements dynamiques et à grande échelle. Les travaux futurs comprennent l'étude du potentiel d'intégration de capteurs de navigation inertielle et l'exploration de composants supplémentaires d'apprentissage en profondeur pour améliorer la robustesse et la précision de la fermeture de boucle.

Title: Visual odometry using heterogeneous cameras for simultaneous localization and mapping for autonomous vehicles

Keywords: Visual odometry, Image matching, Event cameras, Stereovision

Abstract: This Ph.D. thesis addresses the challenges of sensor fusion and Simultaneous Localization And Mapping (SLAM) for autonomous systems, specifically focusing on Autonomous Ground Vehicles (AGVs) and Micro Aerial Vehicles (MAVs) navigating large-scale and dynamic environments. The thesis presents a range of innovative solutions to enhance the performance and reliability of SLAM systems through five methodological chapters.
The introductory chapter establishes the research motivation, highlighting the challenges and limitations of visual odometry using heterogeneous cameras. It also outlines the thesis structure and extensively reviews relevant literature. The second chapter introduces IBISCape, a simulated benchmark for validating high-fidelity SLAM systems based on the CARLA simulator. The third chapter presents a novel optimization-based method for calibrating an RGB-D-IMU visual-inertial setup, validated through extensive experiments on real-world and simulated sequences. The fourth chapter proposes a linear optimal state estimation approach for MAVs to achieve high-accuracy localization with minimal system delay.
The fifth chapter introduces the DH-PTAM system for robust parallel tracking and mapping in dynamic environments using stereo images and event streams. The sixth chapter explores new frontiers in the field of dense SLAM using Event cameras, presenting a novel end-to-end approach for hybrid events and point clouds dense SLAM system. The seventh and final chapter summarizes the thesis's contributions and main findings, emphasizing the advancements made in multi-modal heterogeneous sensor fusion for autonomous systems navigating large-scale and dynamic environments. Future work includes investigating the potential of integrating inertial navigation sensors and exploring additional deep-learning components for improving loop-closure robustness and accuracy.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France

To my family, whose love, encouragement, and unwavering support have been my driving force and inspiration throughout my academic journey ...

Acknowledgements

As I present the experimental, mathematical, and scientific contributions in this thesis, I would like to take a moment to express my deepest gratitude to those who have been instrumental in bringing this work to completion. My sincerest appreciation goes to Professor **Samia Bouchafa-Bruneau**, my thesis directress, whose unwavering support and guidance have been a continuous source of inspiration throughout my academic journey. I am also immensely grateful for the invaluable assistance and advice provided by Professor **Dro Désiré Sidibé**, my thesis co-director. Furthermore, I extend my heartfelt appreciation to Associate Professor **Fabien Bonardi**, my thesis co-supervisor, for his insightful feedback and steadfast encouragement. Additionally, I would like to convey my gratitude to my former colleague and friend, Dr. **Mahmoud Z. Khairallah**, for the stimulating exchanges and collaborative studies we have shared.

Furthermore, I am indebted to the University of Evry (Paris-Saclay) faculty for their exceptional teaching and dedication to imparting knowledge. The rich academic environment provided by the institution has nurtured my intellectual growth and fostered a deep passion for learning. The wide range of courses offered and the stimulating discussions in the classrooms have expanded my horizons and broadened my understanding of the subject matter.

During my Ph.D. journey, I had the privilege of utilizing my teaching skills as part of the institution's teaching mission. I want to extend my gratitude to the professors who entrusted me with teaching responsibilities and provided me with opportunities to engage with students. These experiences reinforced my understanding of the subject matter and allowed me to cultivate essential communication and pedagogical skills. The interactions with the students have been immensely rewarding, and I am grateful for the trust placed in me to contribute to their educational journey.

I would also like to acknowledge my fellow graduate students, whose support, camaraderie, and shared experiences have made the academic journey more enjoyable and inspiring. Your insights, discussions, and encouragement have been invaluable, and I am grateful for the friendships we have formed.

Last but certainly not least, I want to express my deepest gratitude to my family and friends for their unwavering support, understanding, and love throughout this challenging journey. Their constant encouragement and belief in my abilities have been a constant source of strength, and I am incredibly fortunate to have them by my side.

Synthèse

Cette thèse de doctorat présente des solutions innovantes pour relever les défis de la fusion de données multi-capteurs, de la localisation et de la cartographie simultanées (SLAM) pour les systèmes autonomes, en se concentrant spécifiquement sur les véhicules terrestres autonomes (AGV) et les micro-véhicules aériens (MAV) naviguant dans des environnements dynamiques et à grande échelle. La thèse comprend cinq chapitres méthodologiques, chacun apportant une solution unique pour améliorer la performance et la fiabilité des systèmes SLAM.

Le chapitre d'introduction établit la motivation de la recherche en mettant l'accent sur les défis et les limitations actuelles de l'odométrie visuelle utilisant des caméras hétérogènes. La philosophie de la recherche consiste à relever ces défis grâce à une approche innovante qui combine les caractéristiques visuelles extraites des caméras RVB, de profondeur et d'événements pour estimer la pose du capteur. Cette introduction décrit la structure de la thèse et ses différents chapitres, qui comprennent une revue de la littérature, l'extraction et la correspondance des caractéristiques visuelles, l'étalonnage du capteur, l'estimation de l'état hybride, ainsi que le suivi et la cartographie robustes. La thèse vise à faire progresser le domaine de l'odométrie visuelle en introduisant de nouvelles approches qui exploitent les forces des caméras hétérogènes pour surmonter les limites des méthodes traditionnelles. En fournissant une base solide pour les chapitres suivants, l'introduction prépare le terrain pour la contribution globale de la thèse au domaine.

Le deuxième chapitre présente IBISCape, un jeu de données simulé pour valider les systèmes SLAM de haute fidélité qui inclut des API de synchronisation et d'acquisition de données pour la télémétrie de capteurs hétérogènes, la segmentation de la scène de vérité terrain, les cartes de profondeur et l'égo-mouvement du véhicule. Construit à l'aide du simulateur CARLA, qui utilise Unreal Engine pour simuler des scènes hautement dynamiques, le jeu de données comprend 43 sous-ensembles pour l'évaluation de la fiabilité. Ce chapitre propose des cibles d'étalonnage innovantes pour les cartes CARLA et une couche de prétraitement pour l'intégration des événements des capteurs DVS dans n'importe quel système Visual-SLAM basé sur des images. Les derniers systèmes SLAM visuels (RVB, profondeur, événement), inertiels et LiDAR de pointe sont évalués de manière approfondie sur diverses séquences IBISCape collectées dans des environnements dynamiques simulés à grande échelle.

Le troisième chapitre présente une nouvelle méthode basée sur l'optimisation pour l'étalonnage intrinsèque et extrinsèque d'une configuration visuelle-inertielle RGB-D-IMU à l'aide d'un algorithme d'intialisation assisté par GPS. La méthode fournit des estimations initiales fiables pour les paramètres intrinsèques et la trajectoire de la caméra

RVB sur la base d'une méthode d'odométrie visuelle (VO) basée sur le flot optique tout en optimisant les paramètres spatio-temporels tels que la pose de la cible, le nuage de points 3D et les biais de l'IMU en arrière-plan. La méthode est validée par de nombreux résultats expérimentaux sur des séquences réelles et simulées.

Le quatrième chapitre propose une approche linéaire d'estimation optimale de l'état pour les MAV afin d'obtenir une localisation de haute précision avec un retard minimal du système. L'approche comprend une technique de fusion de capteurs basée sur l'optimisation et le filtrage découplés qui permet d'obtenir une précision d'estimation élevée et une complexité minimale du système. Le système utilise des environnements intérieurs et extérieurs réels pour des études de localisation de MAV afin de valider et de tester les résultats de la méthode proposée.

Le cinquième chapitre présente le système DH-PTAM pour un suivi et une cartographie parallèles robustes dans des environnements dynamiques à l'aide d'images stéréo et de flux d'événements. Le système combine les forces des capteurs visuels multimodaux hétérogènes et utilise l'extraction et la description de caractéristiques basées sur l'apprentissage profond pour l'estimation afin d'améliorer la robustesse. Les expériences démontrent que DH-PTAM surpasse les méthodes SLAM visuelles-inertielles de pointe, en particulier dans des scénarios difficiles tels que les mouvements rapides, la HDR et les occlusions. Le système proposé fournit une solution évolutive et précise pour la reconstruction 3D et l'estimation de la pose et offre une API Python basée sur la recherche et disponible publiquement sur GitHub pour d'autres recherches et développements.

Le sixième chapitre explore de nouvelles frontières dans le domaine du SLAM dense à l'aide de caméras évènementielles. Le pipeline proposé est construit sur la bibliothèque open3D pour l'optimisation des graphes de poses avec un simple paradigme de fermeture de boucle basé uniquement sur les nuages de points estimés basés sur les événements. L'alignement des nuages de points et l'estimation de la pose relative sont effectués à l'aide de la méthode efficace de l'état de l'art Teaser++ au lieu de la méthode ICP traditionnelle. Enfin, une validation de la preuve de concept est effectuée sur DSEC, un benchmark public du monde réel.

Le septième chapitre de cette thèse de doctorat résume les contributions et les principaux résultats de la recherche présentée dans les chapitres précédents. Cette thèse propose plusieurs solutions nouvelles qui contribuent à faire avancer la recherche dans le domaine de la fusion de capteurs hétérogènes multimodaux appliquée à la navigation de systèmes autonomes dans des environnements dynamiques à grande échelle. Les repères proposés, les cibles d'étalonnage et les couches de prétraitement offrent une validation fiable des systèmes SLAM. Les algorithmes de calibration et de SLAM proposés permettent une estimation plus précise et plus robuste de la pose. Les techniques de fusion de capteurs proposées permettent une localisation de haute précision avec un retard minimal du système. Le système DH-PTAM proposé constitue une solution évolutive et précise pour la reconstruction 3D et l'estimation de la pose dans des scénarios difficiles. Les travaux futurs comprennent l'étude du potentiel d'intégration de capteurs de navigation inertielle et l'étude de l'intégration de composants d'apprentissage profond supplémentaires pour améliorer la robustesse et la précision de la fermeture de boucle.

Abstract

This Ph.D. thesis presents innovative solutions to tackle the challenges of sensor fusion and Simultaneous Localization And Mapping (SLAM) for autonomous systems, specifically focusing on Autonomous Ground Vehicles (AGVs) and Micro Aerial Vehicles (MAVs) navigating large-scale and dynamic environments. The thesis comprises five methodological chapters, each contributing a unique solution to enhance the performance and reliability of SLAM systems.

The introductory chapter establishes the research motivation by emphasizing the current challenges and limitations in visual odometry using heterogeneous cameras. The research philosophy is to address these challenges through an innovative approach that combines visual features extracted from RGB, depth, and event cameras to estimate the sensor's pose. The chapter outlines the thesis structure and its various chapters, which encompass a literature review, visual feature extraction and matching, sensor calibration, hybrid state estimation, and robust tracking and mapping. The thesis aims to advance the field of visual odometry by introducing novel approaches that exploit the strengths of heterogeneous cameras to overcome traditional method limitations. By providing a solid foundation for subsequent chapters, the introduction prepares the stage for the thesis's overall contribution to the field.

The second chapter presents IBISCape, a simulated benchmark for validating high-fidelity SLAM systems that includes data synchronization and acquisition APIs for telemetry from heterogeneous sensors, ground truth scene segmentation, depth maps, and vehicle ego-motion. Built upon the CARLA simulator, which employs Unreal Engine to simulate highly dynamic scenes, the benchmark comprises 43 datasets for reliability assessment. The chapter proposes innovative calibration targets for CARLA maps and a pre-processing layer for integrating DVS sensor events in any frame-based Visual-SLAM system. The latest state-of-the-art Visual (RGB, Depth, Event)-Inertial-LiDAR SLAM systems are extensively evaluated on various IBISCape sequences collected in simulated large-scale dynamic environments.

The third chapter introduces a novel optimization-based method for intrinsic and extrinsic calibration of an RGB-D-IMU visual-inertial setup with a GPS-aided optimizer bootstrapping algorithm. The method delivers reliable initial estimates for the RGB camera intrinsics and trajectory based on an optical flow Visual Odometry (VO) method while optimizing spatio-temporal parameters such as the target's pose, 3D point cloud, and IMU biases in the back-end. The method is validated through extensive experimental results on real-world and simulated sequences.

The fourth chapter proposes a linear optimal state estimation approach for MAVs to achieve high-accuracy lo-

calization with minimal system delay. The approach includes a decoupled optimization- and filtering-based sensor fusion technique that achieves high estimation accuracy and minimal system complexity. The system uses real-world indoor and outdoor settings for MAV localization studies to validate and test the proposed method's findings.

The fifth chapter introduces the DH-PTAM system for robust parallel tracking and mapping in dynamic environments using stereo images and event streams. The system combines the strengths of heterogeneous multi-modal visual sensors and employs deep learning-based feature extraction and description for estimation to enhance robustness. Experiments demonstrate that DH-PTAM outperforms state-of-the-art visual-inertial SLAM methods, particularly in challenging scenarios such as fast motion, HDR, and occlusions. The proposed system provides a scalable and accurate solution for 3D reconstruction and pose estimation and offers a research-based Python API publicly available on GitHub for further research and development.

The sixth chapter explores new frontiers in the field of dense SLAM using Event cameras. The proposed pipeline is built on the open3D library for pose graph optimization with a simple loop-closure paradigm based only on the estimated event-based point clouds. Point cloud alignment and relative pose recovery are performed using the state-of-the-art efficient method Teaser++ instead of the traditional ICP method. Lastly, a proof of concept evaluation is performed on DSEC, a real-world public benchmark.

The seventh chapter of this Ph.D. thesis summarizes the contributions and main findings of the research presented in the preceding chapters. This thesis proposes several novel solutions that contribute to advancing multi-modal heterogeneous sensor fusion research applied to autonomous systems' navigation in large-scale and dynamic environments. The proposed benchmarks, calibration targets, and pre-processing layers offer reliable validation of SLAM systems. The proposed calibration and SLAM algorithms enable more accurate and robust pose estimation. The proposed sensor fusion techniques achieve high-accuracy localization with minimal system delay. The proposed DH-PTAM system provides a scalable and accurate solution for 3D reconstruction and pose estimation in challenging scenarios. Future work includes investigating the potential of integrating inertial navigation sensors and exploring the integration of additional deep-learning components for improving loop-closure robustness and accuracy.

Contents

List of Figures

List of Tables

1 Introduction

Abstract

This thesis explores the development of simultaneous localization and mapping (SLAM) systems using multiple heterogeneous visual sensors to increase robustness and accuracy in challenging environments. It proposes novel integration methods for sensor data into a comprehensive SLAM chain and investigates the benefits of incorporating unconventional sensors like depth and event cameras. The work revolves around the idea of using data fusion techniques like filtering and optimization to enhance SLAM systems' performance. The thesis provides a structured solution to the scientific gap in sensor fusion and SLAM for autonomous systems. This includes a benchmark for SLAM system validation, a novel calibration method for pose estimation, a decoupled optimization and filtering-based sensor fusion technique, a multi-modal visual sensor-based robust pose estimation and 3D reconstruction system, and a dense SLAM system using event cameras.

"Nothing in life is to be feared, it is only to be understood. Now is the time to understand more, so that we may fear less."

Marie Curie

1.1 Motivation

The field of simultaneous localization and mapping (SLAM) has been a subject of active research in recent years, as it has significant applications in various fields, including robotics, augmented reality, and autonomous driving. Despite the popularity of using visual sensors for SLAM, several challenges persist in complex indoor or outdoor environments, such as smoke, darkness, illumination variations, and seasonal changes (see Figure 1.1).

Figure 1.1: Some challenging driving situations for vision-based SLAM Systems (source: Google).

The motivation of this thesis is to explore the potential benefits of incorporating multiple heterogeneous visual sensors in SLAM systems to address these challenges. Specifically, this research aims to develop novel methods for integrating different types of visual sensors into a complete SLAM chain by utilizing calibration, synchronization, and generic matching techniques. Furthermore, this thesis seeks to overcome limitations in visual odometry and feature matching approaches and adapt them to different types of visual sensors.

The ultimate goal of this research is to provide new insights into developing more effective and robust SLAM systems that can have potential applications in various fields. By incorporating unconventional sensors based on their spectral sensitivity and caption technology, this thesis seeks to contribute to the development of SLAM systems that can overcome the limitations of conventional approaches in challenging environments. The inclusion of IMU and GPS sensors in this research can also improve the accuracy and robustness of SLAM systems.

Overall, this thesis aims to build upon existing research and propose novel methods for integrating and utilizing data from multiple heterogeneous visual sensors in SLAM systems. By doing so, this research can contribute to the development of more effective and robust SLAM systems with potential applications in various fields.

1.2 Philosophy

The philosophy underlying this PhD thesis is centered on the belief that the integration of multiple heterogeneous visual sensors can provide significant benefits in the development of simultaneous localization and mapping systems in complex indoor or outdoor environments. The thesis explores the benefits of incorporating unconventional visual sensors, such as those with different spectral sensitivities (e.g., depth cameras) and caption technology (e.g., event

Figure 1.2: A simulated challenging driving scenario with high intensity fog, rain and darkness as seen from left to right by: RGB camera, Depth sensor, and Event camera (source: IBISCape [1]).

cameras). This is based on the idea that such sensors can capture complementary information that enhances the accuracy and robustness of the localization and mapping processes (see Figure 1.2).

To achieve the integration of multiple heterogeneous visual sensors, the thesis explores different fusion techniques such as filtering and optimization. The philosophy is based on the belief that these techniques can enhance the accuracy and robustness of the localization and mapping processes, as they can reduce the effects of noise and incomplete data.

Overall, the thesis philosophy emphasizes the importance of developing a common representation space for the primitives extracted from different visual sensors, taking into account the heterogeneity and incompleteness of the data. This is based on the understanding that the integration of multiple heterogeneous visual sensors can result in data that is noisy, incomplete, and challenging to work with.

1.3 Thesis Outline

What is the scientific gap in the field of sensor fusion and Simultaneous Localization And Mapping (SLAM) for autonomous systems, particularly in visual odometry using heterogeneous cameras, and how does this thesis address this gap through its outlined chapters and proposed solutions?

This thesis identifies the scientific gap as the current challenges and limitations in visual odometry using heterogeneous cameras, which hinder the performance and reliability of SLAM systems in Autonomous Ground Vehicles (AGVs) and Micro Aerial Vehicles (MAVs) navigating large-scale and dynamic environments. To address this gap and towards a complete multi-modal heterogeneous sensor fusion framework (see Figure 1.3), the thesis proposes the following structure and solutions:

- **Chapter 1: Introduction**

  Establishes the research motivation and philosophy by emphasizing the scientific gap and outlining an inno-

vative approach to address it.

- **Chapter 2: IBISCape: Simulated Benchmark for High-Fidelity SLAM Systems**
  Presents a benchmark for validating SLAM systems, including data synchronization, calibration targets, and pre-processing layers that address challenges related to heterogeneous sensor data.
- **Chapter 3: Optimization-based Method for Intrinsic and Extrinsic Calibration of an RGB-D-IMU Visual-Inertial Setup**
  Introduces a novel calibration method for improving the accuracy and robustness of pose estimation in visual odometry systems.
- **Chapter 4: Linear Optimal State Estimation Approach for MAVs**
  Proposes a decoupled optimization- and filtering-based sensor fusion technique that enhances localization accuracy while minimizing system delay.
- **Chapter 5: DH-PTAM: Robust Parallel Tracking and Mapping in Dynamic Environments**
  Develops a system that leverages heterogeneous multi-modal visual sensors and deep learning-based feature extraction for robust pose estimation and 3D reconstruction.
- **Chapter 6: Dense SLAM using Event Cameras**
  Explores new frontiers in dense SLAM, presenting an end-to-end approach for hybrid events-images dense SLAM system that further addresses the scientific gap.
- **Chapter 7: Conclusion & Perspectives**
  Summarizes the contributions and findings, emphasizing how the proposed solutions collectively address the identified scientific gap in the field of sensor fusion and SLAM for autonomous systems.

1.4 Scientific and Experimental Contributions

This thesis addresses the scientific gap in multi-modal sensor fusion for simultaneous localization and mapping (SLAM) systems in complex indoor or outdoor environments. The proposed methods integrate multiple heterogeneous visual sensors, including unconventional sensors based on their spectral sensitivity and caption technology, and provide reliable calibration and synchronization methods. The proposed methods also address adapting hybrid classical and learning-based features to cameras with different spectral sensitivities and combining time, space, luminance, and motion criteria to establish new events correspondences for matching problems. Additionally, all the produced publications propose common representation spaces for the new primitives extracted from different visual sensors, considering the data's heterogeneity and incompleteness. These contributions can inspire future research on efficient multi-modal calibration and SLAM algorithms based on the fusion of heterogeneous sensors with

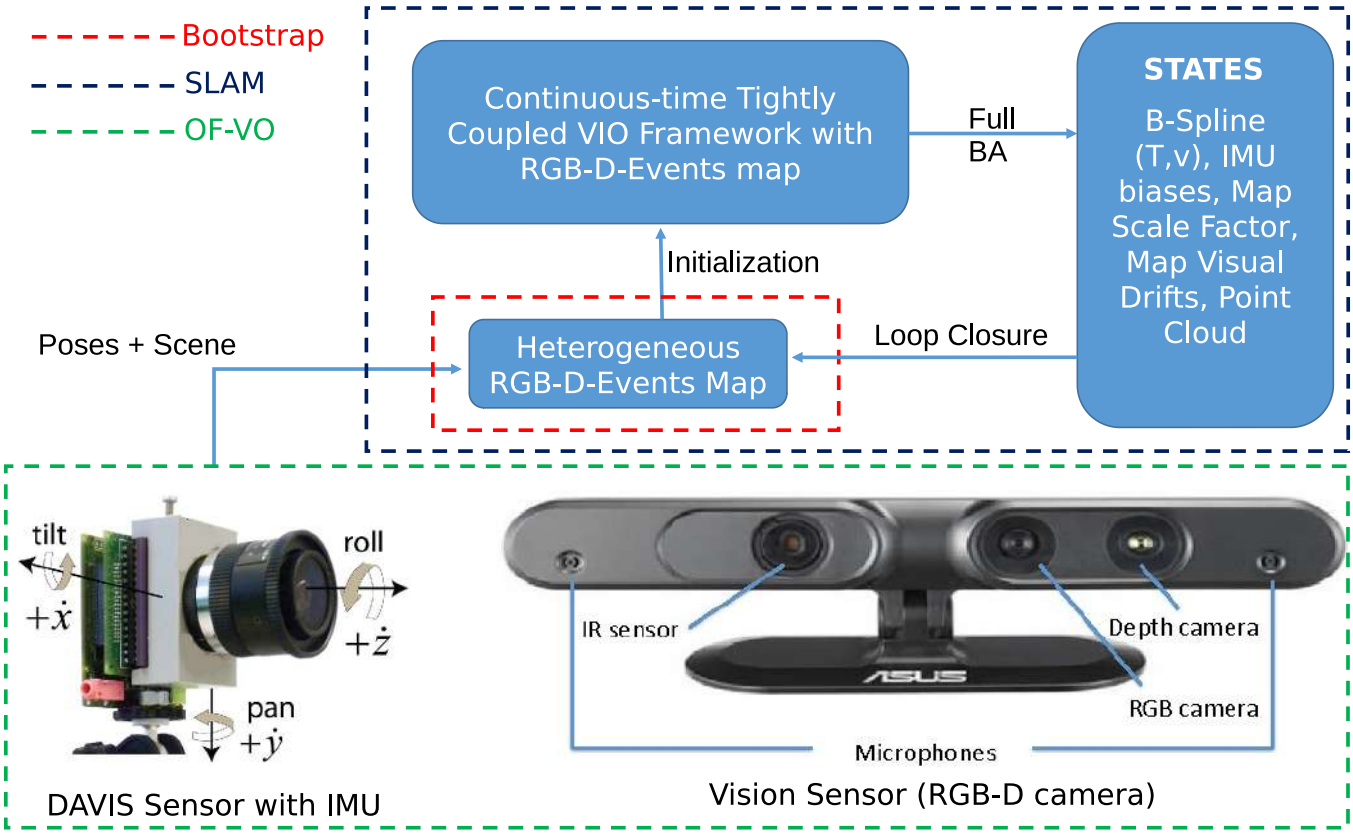


Figure 1.3: A general conceptional schematic for the intended heterogeneous SLAM system. VIO: Visual-Inertial Odometry. B-spline (T,v): the spline manifold nodes control parameters; pose (T(R|t)) and velocity (v). OF-VO: Optical Flow - Visual Odometry. DAVIS sensor: Dynamic and Active VISion Sensor (Event camera). BA: Bundle-Adjustment.

different caption and spectral technologies for reliable continuous-time 3D scene mapping. The thesis contributions can be summed up as follows:

- The first contribution of this thesis is the **IBISCape simulated benchmark**, including telemetry from heterogeneous sensors, ground truth scene segmentation, depth maps, and vehicle ego-motion, for Autonomous Ground Vehicles (AGVs) reliability assessment. It also introduces a novel pre-processing layer for DVS sensor events in any frame-based Visual-SLAM system. This thesis also extensively evaluates the state-of-the-art Visual/Inertial/LiDAR SLAM systems on various sequences in simulated large-scale dynamic environments.
- The second contribution of this thesis is a novel optimization-based method for intrinsic and extrinsic calibration of an RGB-D-IMU visual-inertial setup with a GPS-aided optimizer bootstrapping algorithm. This contribution provides reliable initial estimates for the RGB camera intrinsics and trajectory based on an optical flow Visual Odometry (VO) method. It also includes experiments on real-world and realistically high-quality simulated sequences to validate the proposed calibration algorithm and estimate each sensor's contribution in the multi-modal setup on the vehicle's pose estimation accuracy.
- The third contribution of this thesis is a hybrid optimization/filtering optimal state estimation approach for GPS-aided Micro Aerial Vehicles (MAVs) localization in large-scale landscapes. The proposed strategy shows how the vision sensor can quickly bootstrap a pose and recover from various drifts that affect vision-based algorithms. This contribution provides extensive quantitative and qualitative analyses utilizing real-world and large-scale MAV sequences that demonstrate the proposed technique's higher performance compared to the most recent state-of-the-art algorithms in terms of trajectory estimation accuracy and system latency.
- The fourth contribution presents the DH-PTAM system for robust parallel tracking and mapping in dynamic environments using stereo images and event streams. The proposed system builds upon the principles of S-PTAM and extends it with a deep learning-based approach to handle the sparse and noisy nature of event-based sensors while leveraging the rich information provided by fusion frames. This work provides extensive experiments on both small-scale and large-scale real-world sequences of publicly available benchmarks, demonstrating superior performance compared to state-of-the-art methods in terms of robustness and accuracy in adverse conditions.
- Finally, theoretical and conceptual modeling of a dense event-based SLAM system is presented, paving the way for a novel hybrid dense multi-modal sensor fusion algorithm pushing the limits of visual SLAM systems to new horizons.

These are the scientific collaborations and research articles published during this thesis preparation period:

1. Book chapters:

- **Soliman, Abanob**, Fabien Bonardi, Désiré Sidibé, and Samia Bouchafa. "**HCALIB: A Hybrid RGB-D-IMU Pose Estimation and Calibration Method.**" *Communications in Computer and Information Science, Springer 2023* - Under Review

2. Journal articles:

- **Soliman, Abanob**, Fabien Bonardi, Désiré Sidibé, and Samia Bouchafa. "**IBISCape: A Simulated Benchmark for multi-modal SLAM Systems Evaluation in Large-scale Dynamic Environments.**" *Journal of Intelligent & Robotic Systems* 106, no. 3 (2022): 53. `https://doi.org/10.1007/s10846-022-01753-7`
- **Soliman, Abanob**, Hicham Hadj-Abdelkader, Fabien Bonardi, Samia Bouchafa, and Désiré Sidibé. "**MAV Localization in Large-Scale Environments: A Decoupled Optimization/Filtering Approach**" *Sensors* 23, no. 1(2023): 516. `https://doi.org/10.3390/s23010516`
- **Soliman, Abanob**, Fabien Bonardi, Désiré Sidibé, and Samia Bouchafa. "**DH-PTAM: A Deep Hybrid Stereo Events-Frames Parallel Tracking And Mapping System.**" arXiv preprint arXiv:2306.01891 - Under Submission

3. International Conferences:

- **Soliman, A.**; Bonardi, F.; Sidibé, D. and Bouchafa, S. (2023). **Robust RGB-D-IMU Calibration Method Applied to GPS-Aided Pose Estimation.** In Proceedings of the *18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications* - Volume 4: VISAPP, ISBN 978-989-758-634-7; ISSN 2184-4321, pages 83-94. `https://doi.org/10.5220/0011656800003417`
- Khairallah, M.; **Soliman, A.**; Bonardi, F.; Roussel, D. and Bouchafa, S. (2023). **Flow-Based Visual-Inertial Odometry for Neuromorphic Vision Sensors Using non-Linear Optimization with Online Calibration.** In Proceedings of the *18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications* - Volume 5: VISAPP, ISBN 978-989-758-634-7; ISSN 2184-4321, pages 963-973. `https://doi.org/10.5220/0011660400003417`

4. National Conferences:

- **Soliman, A.**; Bonardi, F.; Sidibé, D. and Bouchafa, S. (2023). **HCALIB : Méthode hybride d'étalonnage et d'estimation de pose RVB-D-IMU.** In Proceedings of *ORASIS 2023 - 19èmes Journées francophones des jeunes chercheurs en vision par ordinateur, Centre National de la Recherche Scientifique [CNRS]*.

2 Heterogeneous SLAM Benchmarking

Abstract

This chapter focuses on introducing IBISCape, a benchmark designed to validate high-fidelity Simultaneous Localization and Mapping (SLAM) systems. The benchmark provides simulated data synchronization and acquisition Application Programming Interfaces (APIs) for capturing telemetry from different types of sensors, such as ground truth scene segmentation, depth maps, and vehicle ego-motion. The foundation of IBISCape is built upon the CARLA simulator, which utilizes the powerful Unreal Engine to create highly dynamic scenes. Within this benchmark, there are 43 datasets available for assessing the reliability of SLAM systems. The chapter also presents innovative calibration targets specifically designed for CARLA maps, along with a preprocessing layer that enables the integration of Dynamic Vision Sensor (DVS) sensor events into any frame-based Visual-SLAM system. To evaluate the performance of the latest state-of-the-art Visual (RGB, Depth, Event)-Inertial-LiDAR SLAM systems, extensive assessments are conducted using various IBISCape sequences obtained from simulated large-scale dynamic environments.

"The important thing is not to stop questioning.
Curiosity has its own reason for existing."

Albert Einstein

2.1 Introduction

Autonomous vehicles navigating in unknown and dynamic environments need to rely on accurate perception systems for real-time 3D mapping. These perception systems must function optimally in all weather conditions and situations. That enables the vehicle to make decisions for its passengers or the surrounding pedestrians and cars. To this objective, many novel technologies have been developed over the last decade. Some use vision sensors such as monocular Visual Odometry (VO) [2], which can suffer from estimations up to a scale factor. Innovative solutions to estimate this scale factor by fusion with another sensor like mono/stereo Visual-Inertial Odometry (VIO) [3, 4, 5] and RGB-D SLAM [6] to add depth information have been proposed. Other works use LiDAR [7] sensor that provides high precision point clouds mapping of the scene, or use the GPS [8] for localization using satellite signal triangulation.

Multi-modal datasets can enrich and broaden the research in the Simultaneous Localization and Mapping (SLAM) field, mainly applied to Autonomous Ground Vehicles (AGVs) navigation in large-scale and dynamic environments. These environments have specific characteristics, such as the dynamic range of the objects' intensities in the scene. For example: mapping an indoor small room with proper lighting can be of higher quality than mapping a road in a city (large-scale) at night with high intensity fog, rain, and wind (outdoors dynamic environment). The advantages of system multi-modality appear when depending on cameras with high dynamic range, such as the DAVIS sensor and regular low-cost cameras and sensors (IMU/GPS). This multi-modality leads to completing the data shortages during the scene mapping and AGV's localization.

Nowadays, multi-modal frameworks of sensors have proven to be attracting the attention of many researchers in robotics perception for different tasks such as calibration [9, 10] and odometry [11, 12]. That is due to the fact that heterogeneous sensors that perceive the environment allow the acquisition of complementary information data about the scene. Moreover, sensors multi-modality can also include redundancy such as stereo-DVS or stereo-RGB cameras configurations. Having redundancy in the system setup can improve both the precision and the quality of the collected scene landmarks. Furthermore, some sensors have a high temporal resolution and are sensitive to the scene intensity changes, such as the DAVIS sensor (Event Camera) [13]. While other sensors can efficiently detect and track landmarks and scene features in the 3D spatial domain, such as RGB-D cameras [14] and LiDAR [15].

Simulated datasets [16, 12, 17, 18, 19] provide the possibility to have sequences in various complex scenarios. Moreover, setting a hardware data acquisition framework with a specific configuration can be costly and time-consuming and is prone to multiple limitations such as the carrier (car, handheld, drone), weather conditions, sensors configuration, and synchronization. Furthermore, open sourcing the data acquisition APIs with configurable calibration targets can widen the research horizon in multi-modal calibration and sensors synchronization to reach reliable and easy algorithms to implement.

IBISCape main contributions to mitigate all these hardware configuration constraints and to facilitate the multi-

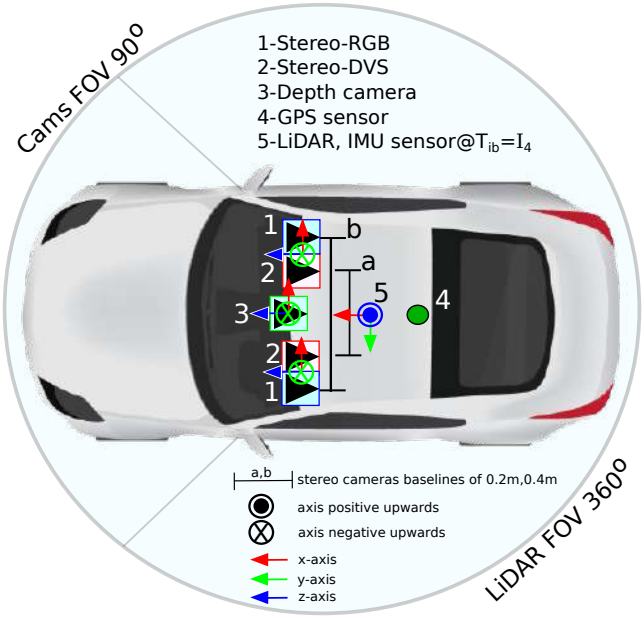


Figure 2.1: Full sensor setup CAD model (Top view). The GPS readings are axis-aligned with the ground truth (GT). The IMU sensor frame is the vehicle body frame of reference with an identity transformation between them $I_{4\times4}$.

modal data synchronization and acquisition process are:

- A benchmark of 43 sequences for multi-modal LiDAR/VI-SLAM applications, besides open-sourcing our multi-modal data acquisition APIs.
- A simulated core sensor suite of most visual-inertial sensors used in assessing visual SLAM systems, along with providing high resolution frames of variable quality depending on the dynamic level of the scene. The full sensor setup is represented in Figure 2.1.
- A solution to calibrate CARLA [20] RGB and DVS cameras with unknown distortion values.
- An advanced high quality 3-channel events pre-processing layer for frame-based Visual-SLAM systems based on the Event Spike Tensor (EST) representation method [21], that can outperform the latest state-of-the-art methods, especially in dynamic environments with adverse conditions.
- A comprehensive and extensive evaluation of state-of-the-art VI systems using IBISCape sequences collected in dynamically simulated large-scale environments, along with a fair comparison with the publicly available real world SLAM systems evaluation benchmarks.

This chapter is organized as follows: Section 2.2, discusses the advantages and novelty of our benchmark compared to the related datasets in the field of multi-modal visual localization, including the state-of-the-art V/VI/LiDAR SLAM algorithms. Section 2.3 explains the data acquisition APIs methodologies and the system calibration in details. Then, an extensive evaluation of the most recent Odometry/SLAM systems using 31 IBISCape SLAM sequences with multiple modalities is represented in Section 2.4. Finally, Section 2.5, provides concluding remarks about our

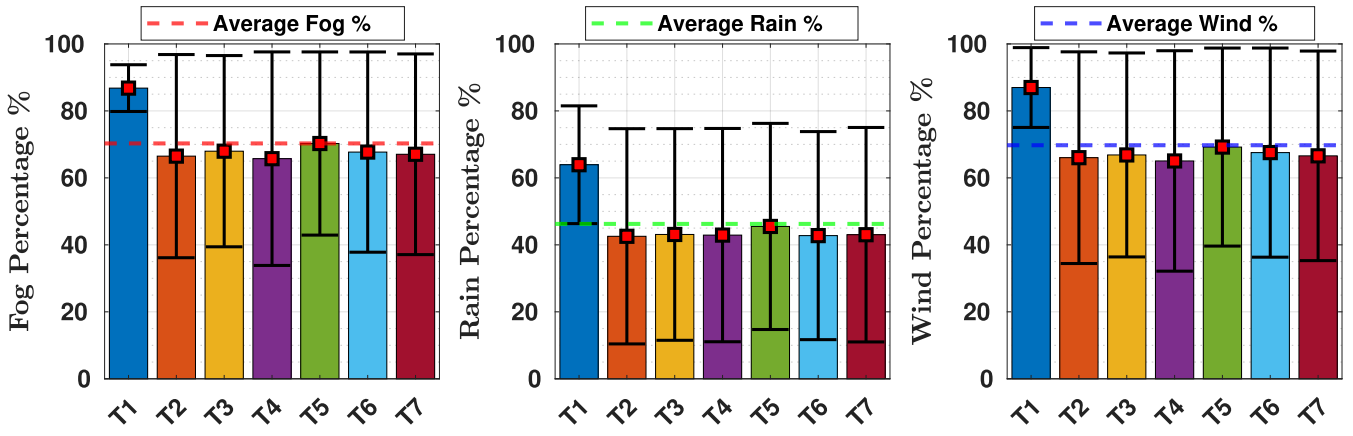


Figure 2.2: The average fog, rain, and wind percentages for sample IBISCape sequences simulated in dynamic weather. Ti is CARLA map of Town number (i). The percentage ranges can be set in IBISCape APIs within the weather simulation class.

work including evaluation observations that motivate and push the development process of new multi-modal SLAM techniques forward, especially in dynamic and large-scale environments based on new findings.

2.2 Related Works

2.2.1 Existing Datasets

The main goal of our benchmark's data acquisition APIs is to collect multi-modal sequences suitable for most robotics perception evaluation, including scene understanding, calibration, and complete SLAM systems. IBISCape APIs are highly configurable concerning the intrinsic and extrinsic setup of the sensors and include all CARLA sensors till the version (0.9.11).

Table 2.1 compares the recent SLAM systems evaluation benchmarks from the sensors types and configuration point of view along with the carrier and ground truth information. Compared to the most recent publicly available benchmarks, IBISCape includes all the sensors needed to evaluate all the state-of-the-art VIO algorithms in any desired configuration including data rates and mono/stereo setups.

Since IBISCape is a simulated benchmark, the GT data for the poses, vehicle controls, scene segmentation, and depth maps are rendered in high precision. This high precision GT data can significantly improve fitting the models of novel data driven VIO architectures that lacks this high quality training data with the real world datasets and hence improving the prediction accuracy.

Thanks to the realistic simulations that CARLA simulator can provide, it has become an important data acquisition environment that recently attracts the attention of many research works in the field on AGVs reliability assessment [17, 18] and simulation to real world transfer-learning techniques [19]. All these recent state-of-the-art works [17, 18, 19] are proposing CARLA-based simulated datasets that exactly emulate other real world datasets collected in

Table 2.1: Core Sensor Suite Comparison of Latest VIO Evaluation Benchmarks.

Benchmark Name	RGB	Depth	DVS[1]	LiDAR[,6]	IMU	GT	Carrier
Real World Platform							
TUM-RGBD [22]	Mono@30Hz	Mono@30Hz	-	-	Accel@500Hz	MoCap@300Hz	Handheld[3]
KITTI[4] [23]	Stereo@15Hz	-	-	1@10Hz,100m	1@100Hz	GPS	Car
Malaga Urban [24]	Stereo@20Hz	-	-	5@75Hz,30m	1@100Hz	GPS	Car
UMich NCLT [25]	Omni@5Hz	-	-	1@10Hz,100m	1@100Hz	GPS/IMU/LiDAR	Segway
EuRoC [26]	Stereo@20Hz	-	-	-	1@200Hz	Laser/Vicon	MAV
Zurich [27]	Mono@30Hz	-	-	-	1@100Hz	GPS	MAV
PennCOSYVIO [28]	Stereo@20Hz	-	-	-	3@200Hz	Markers	Handheld
TUM-VI [29]	Stereo@20Hz	-	-	-	1@200Hz	MoCap	Handheld
Oxford [30]	Stereo@16Hz	Mono@30Hz	-	-	1@500Hz	Vicon	Handheld
KAIST [31]	Stereo@10Hz	-	-	-	1@200Hz	GPS	Car
OIVIO [32]	Stereo@30Hz	-	-	-	1@100Hz	MoCap	Handheld
UZH-FPV [33]	Stereo@30Hz	-	APS/DVS/IMU	-	1@500Hz	Laser	UAV
UMA-VI [34]	Stereo@25Hz	-	-	-	1@250Hz	Camera	Handheld
Blackbird[2] [35]	Stereo@120Hz	Mono@60Hz	-	-	1@100Hz	MoCap	UAV
VCU-RVI [36]	Mono@30Hz	Mono@30Hz	-	-	1@100Hz	MoCap	Handheld[3]
TUM-VIE [37]	Stereo@20Hz	-	Stereo $\leq 10^9 e/s$	-	1@200Hz	MoCap	Helmet
Simulated Platform							
VIODE[2] [16]	Stereo@20Hz	-	-	-	1@200Hz	Simulation	UAV
EVENTSCAPE[2] [12]	Mono@25Hz	Mono@25Hz	Mono $\leq 10^6 e/s$	-	-	Simulation	Car
Paris-CARLA-3D[2] [17]	6@2Hz	-	-	1@10Hz,80m	-	Simulation	Car
KITTI-CARLA[2] [18]	Stereo@10Hz	-	-	1@10Hz,80m	-	Sim@1000Hz	Car
SynWoodScape[2,5] [19]	5@10Hz	5@10Hz	5@10Hz	1@10Hz	1@10Hz	Sim./GPS@10Hz	Car
IBISCape[2,6] (Ours)	Stereo@20Hz	Mono@20Hz	Stereo $\leq 10^7 e/s$	1@20Hz,100m	3@200Hz	Sim./GPS @200Hz	Car

[1] e/s is DVS events per second.

[2] Segmentation frames classify any visible object by displaying it in a different color according to its label (for example, pedestrians in a different color than cars). At the beginning of the simulation, each scene element is created with a tag. In the CARLA simulator, there are 23 segmentation tags with the possibility of adding new tags `https://carla.readthedocs.io/en/latest/tuto_D_create_semantic_tags/`.

[3] Some sequences where collected using a Robot for SLAM systems evaluation.

[4] Annotations for the dynamic objects in the scene are generated using scripts.

[5] No available DVS and LiDAR sensors detailed specifications.

[6] LiDAR sensor rotation rate [Hz] and range [m].

exactly similar situations (number of pedestrians or car types), environments (rural, urban, weather conditions), and sensor setup.

Although the recent works [17, 18, 19] provide high quality sequences with realistic simulations in CARLA, there still exist some shortages that are considered critical for a simulated dataset usability in SLAM systems evaluations summed-up as follows:

- The work of [19] simulates the fisheye camera model with a parametric on-sphere projection of the pixels acquired from a setup of multiple pinhole cameras. However, this work lacks addressing the effect of the actual radial-tangential distortions of CARLA RGB/DVS pinhole cameras.
- A more in-depth performance analysis of the V-SLAM algorithms using a simulated DAVIS sensor in the dynamic environments with adverse weather conditions is needed.
- Since CARLA is an outdoor environment simulator, the acquired data should imitate that of the real world platforms, as a result the SLAM systems evaluation results on these simulated outdoors datasets should be as close as possible to the evaluations performed using real world datasets.

As an overview of the capabilities of the IBISCape benchmark, we collect simulated sequences on a car equipped with most of the low-cost sensors that can be used in the field of robotics perception. This simulation is thoroughly controlled by an autopilot that navigates the car on traffic-aligned roads. Furthermore, weather and scene constituents, including other cars and pedestrians, can be autonomously controlled within our APIs, resulting in datasets that can contend with the real world benchmarks in the literature.

2.2.2 Dynamic Environment Simulation

[16] introduce the concept of dynamic scene simulation with moving vehicles. In the works [17, 18, 19], sequences are collected with some limited pre-defined weather conditions in CARLA. In our benchmark, we extrapolate the concept of dynamic scene simulation to an entire dynamic environment simulation. This simulation includes moving vehicles and pedestrians, as well as a weather class. The weather ticking function updates the weather states every CARLA world tick with a specific speed factor and update frequency. The weather states that can be controlled are clouds, rain, wind, fog, humidity intensity, and sun angles.

A particular observation from sample IBISCape sequences in Figure 2.2 is that our weather update algorithm generates dynamic weather with high intensity fog, rain, and wind with average percentages of 70%, 45%, and 70%, respectively. These dynamic weather conditions result in high trajectory estimation errors due to map loss using existing VIO algorithms. This observation is further verified in Section 2.4 where we compare the trajectory estimation accuracy in diverse weather conditions.

These weather challenges motivate the development of new VIO techniques based on the hybridization of heterogeneous multi-modal sensors to complete the shortages in the map lost during navigation. In Table 2.2, a brief

comparison regarding the scene dynamic class and the amount of information being processed is represented in the camera's frame resolution for all benchmarks represented in Table 2.1. The dynamic level indicators ([C] for Clear / [M] for Moderate / [D] for Dynamic) in Table 2.2, represent the severity of the [W]eather constituents such as: rain, fog, wind and lack of luminosity besides indicating the amount and speed of moving objects in the [S]cene such as other vehicles and walking pedestrians.

Table 2.2: Benchmarks Dynamic Scene Information.

Benchmark	RGB Resolution [px]	Level[1,2]
TUM-RGBD	$1\times640\times480$	C
KITTI	$2\times1384\times1032$	C
Malaga Urban	$2\times1024\times768$	M
UMich NCLT	$1\times1600\times1200$	D (W/S)
EuRoC	$2\times752\times480$	D (S)
Zurich	$1\times1024\times768$	M
PennCOSYVIO	$2\times752\times480$	C
TUM-VI	$2\times1024\times1024$	C
Oxford	$2\times1280\times960$	M
KAIST	$2\times1600\times1200$	M
OIVIO	$2\times1280\times720$	C
UZH-FPV	$2\times640\times480$	C
UMA-VI	$2\times1024\times768$	C
Blackbird	$2\times1024\times768$	C
VCU-RVI	$1\times640\times480$	D (S)
TUM-VIE	$2\times1024\times1024$	D (S)
VIODE	$2\times752\times480$	D (S)
EVENTSCAPE	$1\times512\times256$	C
Paris-CARLA-3D	$6\times2048\times2048$	M (W/S)
KITTI-CARLA	$2\times1392\times1024$	M (W/S)
SynWoodScape	$1\times1024\times1024$	M (W/S)
	$2\times1280\times966$	
	$2\times3264\times2448$	
IBISCape (Ours)	$2\times1024\times1024$	D (W/S)

[1]C: Clear, M: Moderate, D: Dynamic. [2]W: Weather, S: Scene.

2.2.3 Visual Odometry Techniques

The novel VI systems are divided into two prominent techniques: loosely and tightly coupled fusion methodologies [38]. In loosely coupled fusion [39], the camera is used as a black-box pose estimator [2], and an Extended Kalman Filter or an optimizer is applied to fuse the visual pose estimate with the pre-integrated noisy pose from IMU [40]. Whereas in tightly coupled fusion, the scene descriptors (feature points) from the camera are directly inserted to the filter or optimizer to be fused with the IMU readings of the accelerometer and gyroscope using a model that estimates the pose, visual scale, IMU biases, and also re-project the optimized features to build a precise map of the scene.

The tightly coupled VI systems can be approached using two architectures: filter-based like MSCKF [41] and

ROVIO [42], and optimization-based such as VINS-Mono [5], OKVIS [3], and recently ORB-SLAM3 [43] and BASALT [44]. In the work of [45], they compare all these VIO algorithms (except the recent works: ORB-SLAM3 and BASALT) in moderately constrained environments with respect to the dynamic level of the scene. They conclude that ROVIO and VINS-Mono are the best performing techniques concerning system latency, robustness, and accuracy.

In this chapter we focus on evaluating the most recent VI systems: BASALT and ORB-SLAM3 that share the same mapping layer concept based on ORB descriptors. However, their tracking architectures, IMU pre-integration methodologies, and loop-closing constraints are different. In Section 2.4.1, a qualitative performance analysis of BASALT and ORB-SLAM3 on multiple IBISCape SLAM sequences is performed.

Since the DAVIS camera is a visual sensor with the highest dynamic range and temporal resolution (up to 1MHz), it can be deemed one of the efficient sensors to deal with high speed robotics scenarios [46] where conventional cameras may fail. Event cameras work on an unconventional caption technology based on the asynchronous detection of image intensity changes through all pixels on the retina. Novel open-source event-based VO algorithms have been developed in the last few years, including: monocular tracking (EVO) [47], mapping (EMVS) [48], and stereo mapping and tracking (ESVO) [46] methods.

However, the current approaches have a computational complexity limitations based on the number of events and the frame resolution. Another DAVIS sensor limitation is the navigation in high rain, dense fog and dark outdoor environments. This limitation is recently studied in [49] by fusing RGB frames with DVS events in an object detection application. In this work, we propose a novel low complexity events-only pre-processing layer that outputs a high quality 3-channel event tensors that can outperform the data driven approach (E2VID) [50], especially in outdoors environments with adverse weather conditions.

The LiDAR sensor operates on an efficient ranging technology that measures the distance to target objects based on the time lapse between the emitted and received laser rays. LiDAR has a sensing range up to 200 meters and a Field Of View (FOV) up to 360°. Due to its operational technology and technical capabilities, the LiDAR can be deemed as the most reliable sensor for Odometry (LOAM) [51] and SLAM (MULLS) [52] tasks in large-scale dynamic environments.

2.3 Core Sensor Suite

Sensors in IBISCape APIs are highly configurable according to the intended mission, we have set an initial sensor configuration for our experiments that can be easily changed. This initial configuration of the IBISCape core sensor suite is given in Table 2.1.

Table 2.4 shows the distribution of IBISCape sequences with different sensor modalities and configurations in all dynamic environmental conditions. All datasets in every sensor suite are synchronized during acquisition and timestamped in nano-seconds for high precision. Moreover, during the sequence collection, the vehicle control

Figure 2.3: Excitation of the vehicle pitch and roll angles using bubble bumps for reliable calibration results using Kalibr.

Table 2.3: Simulated LiDAR Characteristics.

Parameter	Value set in CARLA
`channels`	64
`range`	100.0 [m]
`points_per_second`	5e6
`rotation_frequency`	20 [Hz]
`upper_fov`	$15.0°$
`lower_fov`	$-25.0°$
`horizontal_fov`	$360.0°$
`atmosphere_attenuation_rate`	0.004
`dropoff_general_rate`	0.45
`dropoff_intensity_limit`	0.8
`dropoff_zero_intensity`	0.4
`noise_stddev`	0.0

forces are saved as normalized vectors within the range [0, 1] and the steering angle in the range [-1, 1].

Simulated LiDAR intrinsics are given in Table 2.3, where `atmosphere_attenuation_rate` is a factor that defines the sensor wave length and atmospheric conditions. To ensure a better realistic LiDAR measurements, CARLA defines a random drop proportion of points with a general drop rate factor and a drop rate factor based on the point intensity.

These control commands are normalized with respect to their maximum attained value based on the chosen vehicle dynamics. One of the advantages of CARLA simulator is that we can tune the physical properties of the vehicle and its wheels.

Simulated GPS data is collected with all setups and synchronized with the GT pose. A text file with every framework explains its dataset files contents in detail. The data access manual for the 43 sequences and the acquisition APIs is given in details in the link in the extended data section in Appendix A.

2.3.1 Cameras Intrinsic & Extrinsic Calibration

One of the advantages of IBISCape benchmark is providing calibration targets for evaluating multi-modal calibration algorithms as well as SLAM systems performance analysis. The more erroneous the calibration parameters, the more incorrect pose is estimated. Although the intrinsic calibration parameters of CARLA cameras can be configured

Table 2.4: IBISCape Sequences & Sensor Setup.

	Acquisition Sensor Suite	Clear	Mod.	Dyn.
Calibration	IMU	2	-	-
	2xRGB+IMU (SVI)	2	-	-
	2xDVS+2xIMU (ESVI)	2	-	-
	RGB-D	2	-	-
	LiDAR+2xRGB	2	-	-
	Full Sensor Setup	2	-	-
SLAM	2xRGB+IMU (SVI)	2	2	3
	2xDVS+2xIMU (ESVI)	2	2	2
	RGB-D	2	2	2
	LiDAR+2xRGB	2	2	2
	Full Sensor Setup	2	2	2
	Total = 43	22	10	11

directly in the APIs, there is no direct way to set the lens distortion coefficients till version (0.9.11). Consequently, we propose introducing the first calibration targets (Checkerboard (7×7) and AprilGrid (6×6)) to one CARLA map (Town 3). Moreover, to excite all angles, especially the pitch and roll angles which are not easy to be simulated in a car, we introduce artificial bumps in the form of bubbles and waves, as shown in Figure 2.3.

Instead of simulating blinking LED lights that cannot be used in a multi-modal calibration framework that includes RGB frames, we use Kalibr [53] to calibrate the stereo RGB cameras and the stereo DVS sensors after performing a frame reconstruction from events using the generic framework E2CALIB [54] (sample in Figure 2.4). Since active illumination cannot be used to calibrate conventional cameras such as mono/stereo-RGB cameras, E2CALIB with the traditional calibration targets makes it possible to calibrate DVS sensors as any conventional camera. Hence, all cameras' intrinsic and extrinsic parameters in a multi-modal framework can be calibrated irrespective of their caption technology, i.e., frames or events.

All IBISCape cameras operate on a global shutter mode with a FOV of $90°$. Table 2.5 shows the specific DVS sensors parameters set during our simulations, including the positive/negative thresholds associated with an increment in brightness change along with their white noise standard deviation for positive/negative events.

Table 2.5: Simulated DVS Characteristics.

Parameter	Value set in CARLA
`+ve/-ve_threshold`	0.3
`sigma_+ve/-ve_threshold`	0.0
`refractory_period_ns`	0.0
`log_eps`	0.05

The known camera model is a pinhole model with unknown distortion parameters for RGB and DVS cameras. We calibrate our cameras using Kalibr **pinhole-radial-tangential** and **pinhole-equidistant** distortion models. The

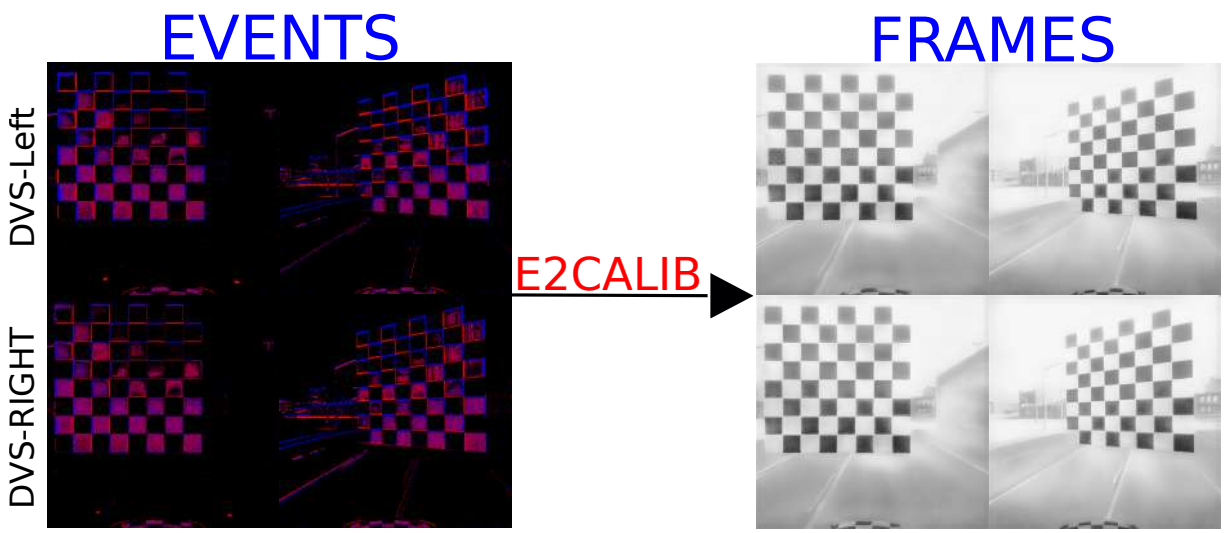


Figure 2.4: Raw events to reconstructed frames using E2CALIB.

calibration process is validated based on two criteria:

- The estimated stereo baselines (extrinsics) compared to the GT values set in our acquisition APIs (see Tables 2.6, 2.7).
- The quality of the optimization process that can be determined from the pixels re-projection errors and the number of optimization constraints (see Table 2.8).

Table 2.6: Stereo DVS sensors and RGB Cameras intrinsic parameters estimation using Kalibr. f_x and f_y, c_x and c_y are the focal lengths and principal point coordinates, respectively. k_1, k_2 and k_3, k_4 are the radial and tangential distortion coefficients, respectively. Calibration is performed using the **Checkerboard** target.

	Camera Model	f_x	f_y	c_x	c_y	k_1	k_2	k_3	k_4
DVS	cam0-radtan	517.07	517.59	506.41	513.27	-2.32e-3	7.12e-4	1.97e-4	-8.87e-4
	cam1-radtan	517.45	517.79	504.48	512.89	-8.34e-4	-1.08e-3	9.11e-5	-1.28e-3
	cam0-equi	375.85	373.29	573.79	513.44	-0.0122	1.9684	-3.8539	2.82
	cam1-equi	370.65	368.16	572.65	513.1	0.2912	0.2954	-0.2626	0.2344
GT	**cam0/cam1**	512.0	512.0	512.0	512.0	-	-	-	-
RGB	cam0-radtan	513.55	513.07	511.0	510.26	1.92e-3	-1.83e-3	-8.5e-4	2.1e-4
	cam1-radtan	512.51	512.87	512.0	512.1	-2.75e-3	3.16e-3	3.7e-4	-3.8e-4
	cam0-equi	511.11	511.18	511.24	511.0	0.3533	0.065	0.181	-0.058
	cam1-equi	512.41	512.31	512.0	512.34	0.3269	0.1084	0.1495	-0.0505
GT	**cam0/cam1**	512.0	512.0	512.0	512.0	-	-	-	-

Based on these two criteria and the obtained results, we can conclude that the **pinhole-radtan** camera-distortion model best fits both RGB and DVS cameras simulation in CARLA. This conclusion is due to its lowest re-projection errors and highest stereo baseline estimation accuracy.

We provide all the calibration configuration files and various ROS scripts to convert the raw dataset files to `rosbag` and `.h5` file formats for Kalibr and E2CALIB frameworks.

Table 2.7: Estimation quality is further validated by comparison to the stereo baselines set in CARLA. cam0 and cam1 are the left and right cameras, respectively.

	Camera Model	Stereo Baseline (t [m])
DVS	cam0-radtan cam1-radtan	q=[3.18e-4 -1.77e-3 3.17e-5 1] t=[**-0.1986** 0.0009 0.0131]
DVS	cam0-equi cam1-equi	q=[3.69e-4 -1.16e-3 -1.14e-4 1] t=[**-0.1902** 0.003 0.0115]
GT	**cam0/cam1**	q=[0 0 0 1], t=[**-0.2** 0 0]
RGB	cam0-radtan cam1-radtan	q=[-0.0021 1.45e-4 4e-5 1] t=[**-0.403** 0.0103 -0.004]
RGB	cam0-equi cam1-equi	q=[-.0014 -3.5e-4 -2e-5 1] t=[**-0.413** 0.005 0.01]
GT	**cam0/cam1**	q=[0 0 0 1], t=[**-0.4** 0 0]

Table 2.8: Re-projection errors & optimization constraints.

	Camera Model	Re-projection errors [px.]	Edges
DVS	cam0-radtan	[0.000132, -0.000016]	61397
DVS	cam1-radtan	[0.000163, -0.000009]	61397
DVS	cam0-equi	[-0.000740, 0.000008]	61397
DVS	cam1-equi	[-0.000703, 0.002294]	61397
RGB	cam0-radtan	[-0.000034, -0.000007]	29008
RGB	cam1-radtan	[0.000034, 0.000007]	29008
RGB	cam0-equi	[-0.000067, 0.000001]	29008
RGB	cam1-equi	[0.000064, 0.000007]	29008

2.3.2 Simulated IMU Calibration

IBISCape novel calibration methodology is based on fixing the high quality calibration target in the center of the frame and moving the vehicle towards it with a complete manual control. Furthermore, adding bumps in its way in the form of a big wave and spherical bubbles, can ensure the sufficient excitation of the inertial sensor for precise system (IMU+cameras) calibration.

In CARLA, IMU measurements are modeled as most low-cost real world IMUs containing a particular bias b and white gaussian noise n. Thus, the GT angular velocities ω and linear accelerations a in the IMU frame are modeled as

$$\omega_{GT} = \omega_{gyro} - b_g - n_g, \quad a_{GT} = a_{accel} - b_a - n_a. \tag{2.1}$$

The standard deviation $\sigma_{wa}, \sigma_{ba}, \sigma_{wg}, \sigma_{bg}$ values are given in Table 2.9, and Allan Deviation plots are given in Figure 2.5 calibrated by the IMU Still Calibration Tool in [34] using a 300 [hrs] of IMU simulated sequence.

In Table 2.9, IMU still calibration shows a remarkable difference between the GT values we set in CARLA and the estimated ones. This is an expected observation, since in a simulation environment the standard deviation and bias values set as GT are the dynamic IMU covariance values which can't be estimated by the static Allan deviation method [55].

Till CARLA version 0.9.11, the acceleration bias standard deviation value cannot be manually set within the simulation. As a result, an accurate and reliable IMU still calibration is essential to obtain simulated datasets with usable IMU measurements. We evaluate the IBISCape Stereo-Visual Inertial (SVI) sequences using the still calibration values for the IMU noises.

Table 2.9: Simulated IMU Still Calibration Results.

Parameter	CARLA	Calibrated
σ_{ba} $[m/s^2/\sqrt{Hz}]$	-	4.983e-3
σ_{wa} $[m/s^3/\sqrt{Hz}]$	7e-2	3.167e-6
σ_{bg} $[rad/s/\sqrt{Hz}]$	-	2.839e-4
σ_{wg} $[rad/s^2/\sqrt{Hz}]$	4e-3	1.916e-7

2.3.3 Inter-sensor Extrinsic Parameters

The CAD model of the GT extrinsic relation between all the sensors in a full sensor setup is represented in Figure 2.6. The axes shown on the camera's center-line are given for the visual sensors only: RGB, DVS, Depth, Segmentation. All IMUs axes conventions are similar to that shown on the IMU0 center-line. Axes color and direction conventions coincide precisely with the Top view CAD model in Figure 2.1.

There is no orientation change between cameras i.e. $\delta\theta = [0, 0, 0]$ and all cameras have the relative rotation

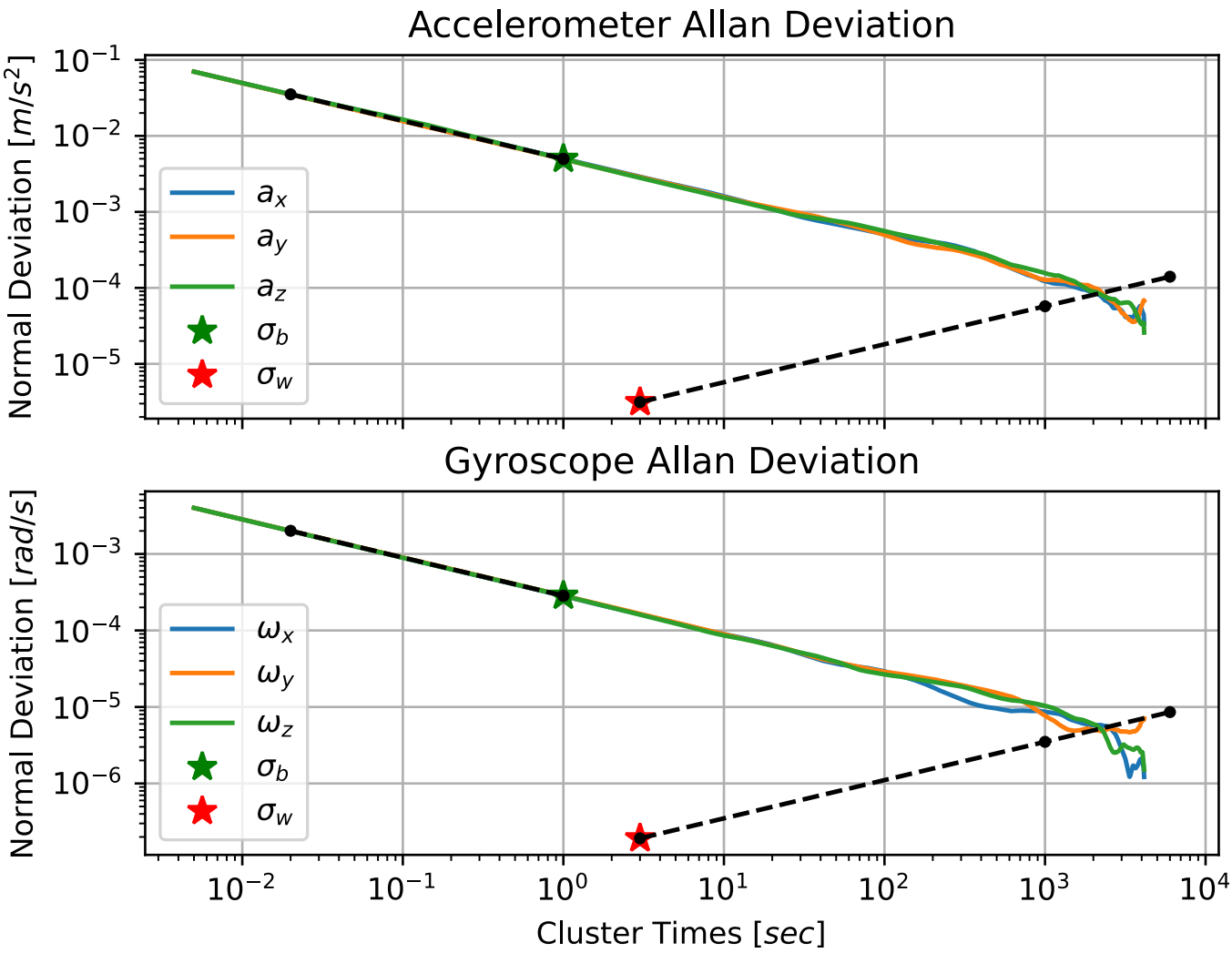


Figure 2.5: IMU log-log scaled plot of Allan-variances over the cluster time. We calculate the IMU noises σ_b and σ_w at cluster times 1 sec. and 3 sec. with slopes $\mp 1/2$ respectively.

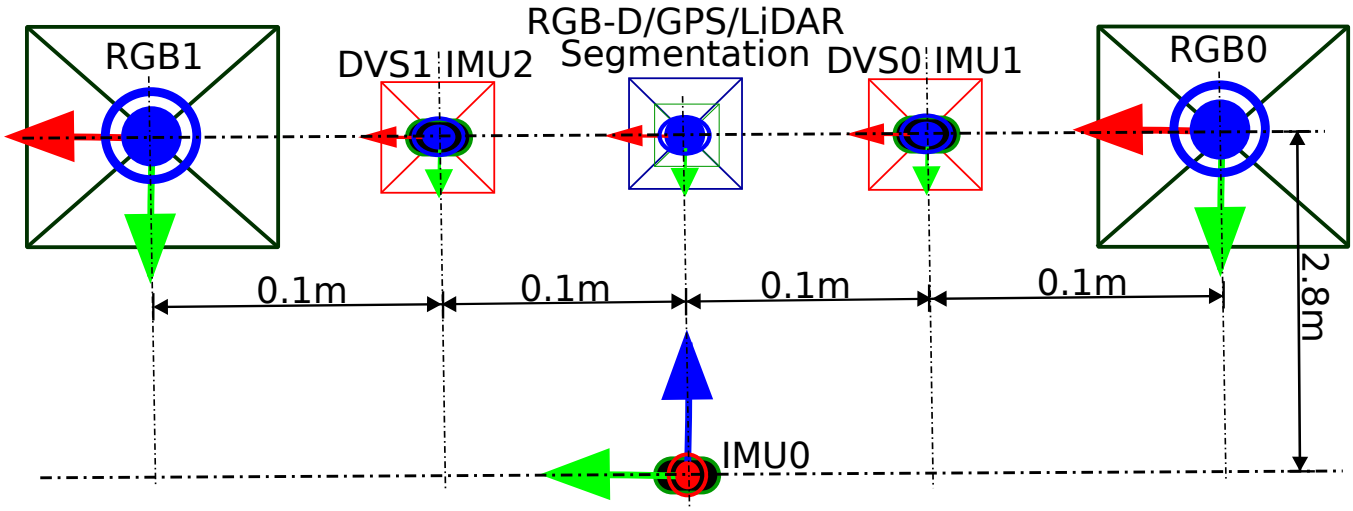


Figure 2.6: Full sensor setup CAD model (Front view).

$q_{cam_1}^{cam_2} = [0, 0, 0, 1]$. In Table 2.10, we give the exact GT values for each sensor location with respect to the IMU0 (body) axes.

In the RGB-D sensor setup, the simulated RGB and Depth cameras have a concentric configuration where both the focal centers are coincided. Moreover, IBISCape data acquisition APIs are written to be highly configurable with respect to the inter-sensor extrinsic parameters with the ease of adding and removing sensors.

Table 2.10: IBISCape Full Sensor Setup Extrinsics.

Sensor	X,Y,Z Translation to IMU0 [m]
Left RGB0	[0.0, 0.2, -2.8]
Right RGB1	[0.0, -0.2, -2.8]
left DVS0 + IMU1	[0.0, 0.1, -2.8]
Right DVS1 + IMU2	[0.0, -0.1, -2.8]
RGB-D cameras	[0.0, 0.0, -2.8]
GPS	[0.2, 0.0, -2.8]
LiDAR	[0.0, 0.0, -3.0]
GT Segmentation	[0.0, 0.0, -2.8]
GT Pose	[0.0, 0.0, 0.0]

2.4 Evaluation

2.4.1 Efficient VI Systems

We use our IBISCape sequences to evaluate state-of-the-art monocular and stereo VI-SLAM algorithms which are ORB-SLAM3 [43] and BASALT VIO [44]. Their choice is because they are the latest state-of-the-art SLAM (ORB-SLAM3) and VIO (BASALT) algorithms. Accordingly, their extensive evaluation on new large-scale and dynamic environment (scene and weather) IBISCape sequences can facilitate detecting their limitations and performance regarding their

accuracy and robustness.

BASALT uses a sparse set of FAST keypoints, tracks them between consecutive frames based on optical flow (KLT) [56], and uses a pyramidal resolution method to ensure reliable and robust tracking in large-scale displacements tracking. Two layers for local bundle adjustment and global pose graph optimization are implemented for precise localization, mapping, and loop-closing. Furthermore, partial marginalization non-linear factors are applied to remove the IMU and feature outlier measurements for constant latency localization.

ORB-SLAM3 is developed to withstand a prolonged duration of low visual information. When a map is disturbed, it initiates a new map that will be smoothly merged with previous maps when revisiting similarly mapped areas. That results in a robust system that operates in dynamic environments and is much more accurate and robust than previous approaches.

Both ORB-SLAM3 and BASALT relate to the optimization-based tightly-coupled fusion stereo VI systems. In Section 2.4.2.1, a detailed evaluation of their performance in large-scale dynamic environments is performed.

In Section 2.4.2.3, our stereo event cameras configuration is used to evaluate the latest open-source stereo DVS mapping and tracking method ESVO [46], along with the novel event-based mapping method EMVS [48]. In the work of E2VID [50], the authors evaluate their event-based frame reconstruction method in the application of monocular VIO, and their method has shown superior performance compared to the other frame-based and event/frame-based methods in comparison. However, these experiments are carried out on indoor sequences with ideal environmental conditions.

In our evaluations on IBISCape, inspired by the work of E2VID, we extrapolate these experiments to include stereo V-SLAM systems in outdoors dynamic environments. Then, we propose an alternative 3-channel event-based frame reconstruction layer that can outperform the quality of E2VID visually as shown in Figure 5.8,5.7 and numerically as given in Table 2.12.

In Section 2.4.2.5, an extensive in-depth evaluations of the latest LiDAR based Odometry/SLAM algorithms MULLS [52] and an advanced version of LOAM [51] in dynamic environments with adverse weather conditions is provided. All LiDAR SLAM sequences simulate multiple loop closure detection situations. Section 2.4.2.6 compares the evaluation process which is run on `31 IBISCape sequences` given in Table 2.11 simulated in various large-scale dynamic environments with the real world evaluations on the state-of-the-art benchmarks in literature.

2.4.2 Performance Analysis

To ease the comparison with the previous and future SLAM system benchmarks, the performance analysis is done using the two known SLAM systems evaluation metrics defined in [57]:

(i) The RMS of Absolute Trajectory Error (ATE) for all (n) estimated poses, and defined as:

Table 2.11: IBISCape Sequences Specifications.

Sequence	Specifications			
	Length [m]	Duration [sec]	Size[1]	Loop Closure
Full Setup				
`Clear-1`	214.6313	60.52	1211	-
`Clear-2`	251.0401	70.55	1412	-
`Moderate-1`	368.9815	71.08	1422	-
`Moderate-2`	104.5391	29.92	599	-
`Dynamic-1`	217.9678	70.24	1405	-
`Dynamic-2`	61.2707	23.38	468	-
SVI Setup				-
`Clear-1`	140.2081	70.16	1404	-
`Clear-2`	141.1631	71.45	1429	✓
`Moderate-1`	253.8933	64.40	1288	-
`Moderate-2`	330.6167	85.98	1719	-
`Dynamic-1`	248.6546	72.35	1448	-
`Dynamic-2`	289.0983	74.01	1480	-
`Accident`	23.6777	6.13	123	-
RGB-D Setup				
`Clear-1`	223.1038	74.95	1500	-
`Clear-2`	360.5324	89.55	1792	-
`Moderate-1`	209.1469	72.65	1454	-
`Moderate-2`	233.6294	70.00	1401	-
`Dynamic-1`	208.0217	65.50	1311	-
`Dynamic-2`	406.3022	75.65	1514	-
ESVI Setup				-
`Clear-1`	116.3213	23.98	4672	-
`Clear-2`	251.5679	60.61	12123	-
`Moderate-1`	264.8653	72.91	13980	-
`Moderate-2`	274.8627	61.13	4390	-
`Dynamic-1`	333.2455	71.54	13997	✓
`Dynamic-2`	15.0866	23.87	11771	-
LiDAR Setup				
`Clear-1`	38.2770	13.85	278	✓
`Clear-2`	64.8373	22.75	456	✓
`Moderate-1`	111.0011	37.50	751	✓
`Moderate-2`	235.0245	76.90	1539	✓
`Dynamic-1`	81.3070	28.90	579	✓
`Dynamic-2`	146.7856	52.25	1046	✓

$$\mathrm{ATE}(\hat{T}^{(1:n)}, T_{gt}^{(1:n)}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||t_i||^2}\ [m], \tag{2.2}$$

where $\hat{T}^{(1:n)}$, $T_{gt}^{(1:n)} \in \mathrm{SE}(3)$ are the estimated and ground truth trajectories, respectively. $t_i \in \mathrm{R}(3)$ is the translation vector of the absolute trajectory error E_i at time step i where $E_i(R_i, t_i) = T_{gt(i)}^{-1} T_{rel} \hat{T}_i \in \mathrm{SE}(3)$, and T_{rel} is rigid-body transformation corresponding to the least-squares solution that maps the $\hat{T}$ trajectory onto the T_{gt} trajectory calculated by optimization.

(ii) Relative Pose Error (RPE) at every i-th frame, and defined as:

$$\mathrm{RPE}(\hat{T}^{(1:n)}, T_{gt}^{(1:n)}) = ||\delta t_i|| \quad [m], \tag{2.3}$$

where δt_i is the translation vector of the relative pose error $e_i(\delta\theta_i, \delta t_i) = (T_{gt(i)}^{-1} T_{gt(i+\Delta)})^{-1} (\hat{T}(i)^{-1} \hat{T}(i+\Delta)) \in \mathfrak{se}(3)$ at time step i with a fixed time interval Δ for our local trajectory increments.

For the orientations, RPE values are given in degrees. We use the same formula after replacing the translation vector δt_i with the rotation part $\delta\theta_i$ in e_i by applying the vee operator to the skew-symmetric error matrix:

$$\mathrm{RPE}(\hat{T}^{(1:n)}, T_{gt}^{(1:n)}) = ||\lfloor \delta\theta_i \rfloor_\vee|| \quad [rad] \tag{2.4}$$

We discuss a thorough descriptive and analytical evaluation for the latest state-of-the-art SLAM systems in the following sub-sections. The descriptive and analytical studies for every sensor setup raise the confidence in the novelty and usability of the IBISCape benchmark, using the calibrated RGB and DVS cameras distortion parameters along with the IMU still calibration.

To ensure a fair evaluation process, all the data acquisition APIs and benchmarking experiments are executed on a 16 GB RAM laptop computer running 64-bit Ubuntu 20.04.3 LTS with AMD(R) Ryzen 7 4800h $\times 16$ cores 2.9 GHz processor and a Radeon RTX NV166 Renoir graphics card.

2.4.2.1 Stereo Visual-Inertial (SVI) Setup Evaluation

IBISCape Stereo Visual Inertial (SVI) sequences push one of the limits of the ORB-SLAM3 system as mentioned in [43], which is the IMU initialization of planar motion of vehicles like cars. In Figure 2.7(A), this limitation constraint was further tested using the `Dynamic 1` sequence with significantly dimmed light and rapid scene motions. The ORB-SLAM3 IMU initialization failed to start with the mapping layer. This failure has led to a significant trajectory drift due to the map loss. This IMU initialization failure problem is also observed in the `Dynamic 2` sequence with the BASALT system.

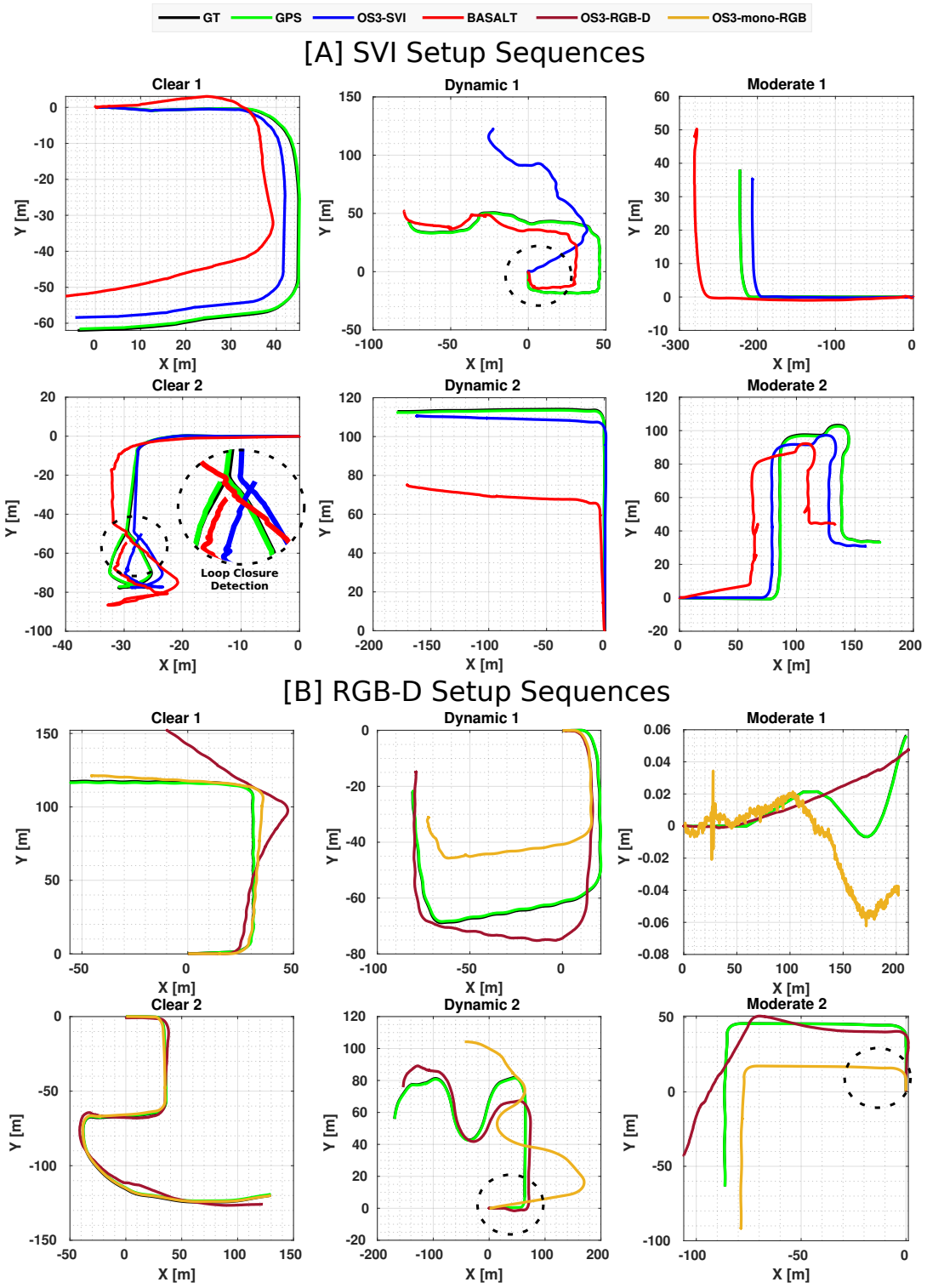


Figure 2.7: Trajectories estimated by ORB-SLAM3 and BASALT SLAM systems using IBISCape sequences with **SVI sensor setup** and **RGB-D sensor setup**, with comparison to their ground truth and GPS paths. For the set `(A) SVI SETUP`, ORB-SLAM3 Stereo Visual Inertial Odometry (SVI) algorithm performance is analysed and compared to the BASALT SVI algorithm. Whereas for the set `(B) RGB-D SETUP`, two ORB-SLAM3 algorithms: Monocular RGB and RGB-D SLAM systems, are assessed with respect to each other after estimation alignment with the GT and scale factor recovery using the GPS measurements. OS3: ORB-SLAM3.

In Table 2.12, the other sequences, `Clear 1,2, Moderate 1,2`, show superior performance for the trajectory estimation using the ORB-SLAM3 system over BASALT based on both overall ATE and incremental RPE values. IBISCape SVI sequences are provided in **raw** and **rosbag** formats, along with the evaluation configuration files `.json` and `.yaml` for BASALT and ORB-SLAM3.

Although sharing ORB keypoints for loop-closing in BASALT and scene descriptors in ORB-SLAM3, BASALT has shown superior accuracy and robustness regarding the visual-inertial sub-system than an early version of ORB-SLAM [44]. This better performance is due to the inertial layer of BASALT that utilizes recovered non-linear factors summarizing IMU and visual tracking on the higher layer of VIO.

However, the latest version of ORB-SLAM3 proved to be much more accurate than BASALT during evaluation on most of the IBISCape sequences, as shown in the performance analysis results in Table 2.12. Despite the superior performance of ORB-SLAM3 over BASALT, we note that the trajectory estimation is much faster in BASALT than in ORB-SLAM3. This evaluation observation validates the proposed comparison in Table (I) in [43].

2.4.2.2 RGB-D Setup Evaluation

One of the advantages of IBISCape sequences is the variety of its sensors' multi-modality. While SVI sequences can provide the scene depth information by stereo RGB cameras and augment the scale factor using the inertial measurements, IBISCape RGB-D sequences offer another sensor modality to measure the scene depth: the depth camera. After alignment with the GT and scale factor recovery using the GPS measurements, we evaluate two ORB-SLAM3 algorithms: the monocular RGB and the RGB-D SLAM systems. In Figure 2.7, it is evident that adding the depth information results in more accurate trajectory estimation with a minor map loss in dynamic weather.

We notice this map loss clearly with the mono-RGB using `Dynamic 1,2, Moderate 1,2` sequences. However, in clear weather sequences `Clear 1,2`, the monocular RGB SLAM can outperform the RGB-D SLAM as seen in Table 2.12 with respect to the ATE values. IBISCape RGB-D sequences are provided in **raw** format with the RGB and Depth cameras `association.txt` file for every sequence, along with the evaluation configuration `.yaml` files for ORB-SLAM3 RGB-D and mono-RGB systems.

2.4.2.3 Event Stereo Visual-Inertial (ESVI) Setup Evaluation

IBISCape event-based sequences address two corner case scenarios introduced to the event-based monocular/stereo VO algorithms. The first scenario is the planar motion in **large-scale** environments; this scenario leads to millions of events fired at locations in the scene that can be tens of meters away from each other. These environments consume much time to reconstruct a map, leading to significant processing time gaps between the tracking and mapping layers of the odometry algorithm.

As a result, the ESVO [46] experiments on IBISCape sequences fails during the trajectory estimation giving an error indicating inconsistency between the tracking and mapping layers, although maps initialize successfully. Accordingly,

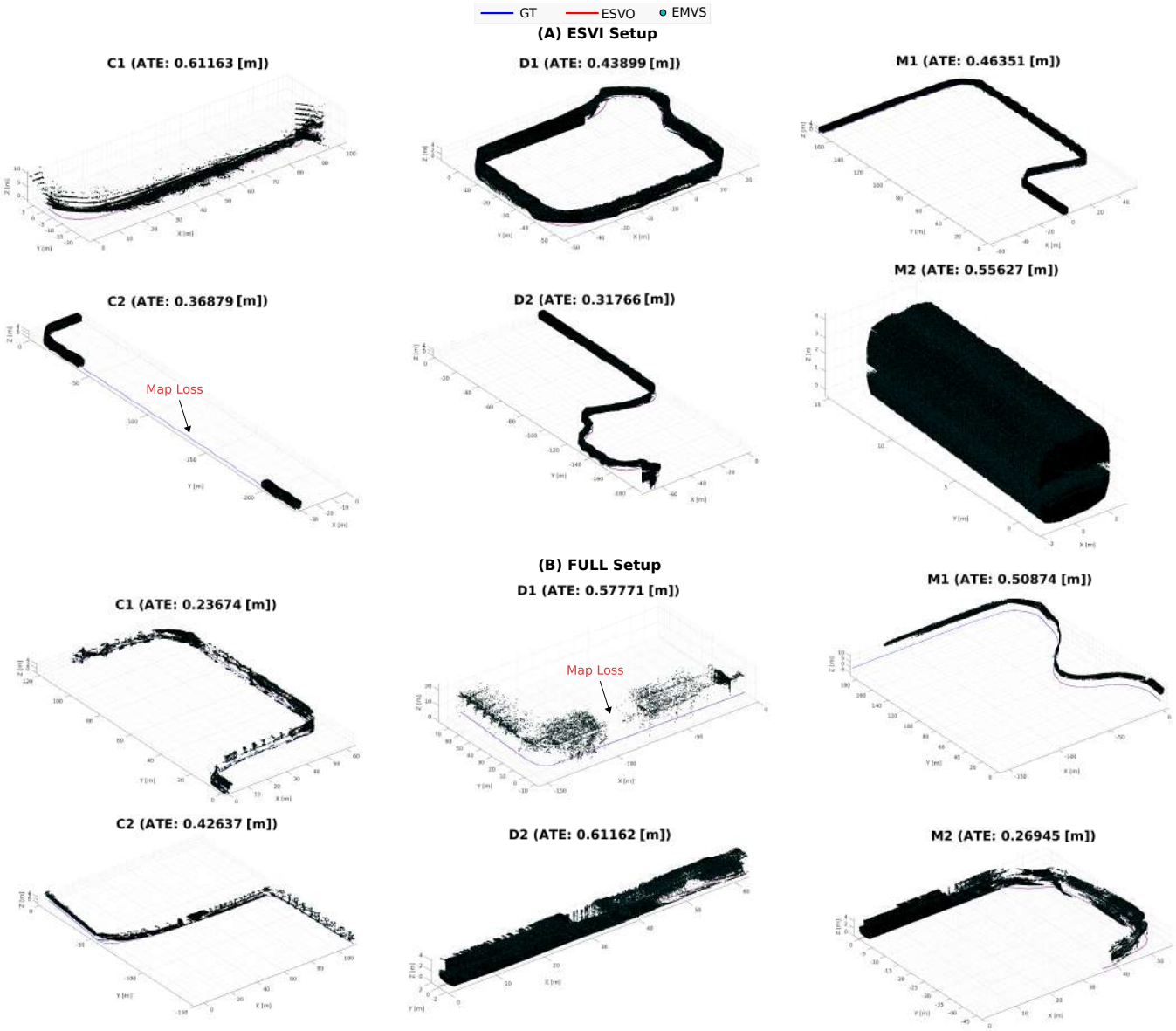


Figure 2.8: Pose estimation by ESVO and point cloud reconstruction by EMVS algorithms on **ESVI** and **FULL sensor setups**.

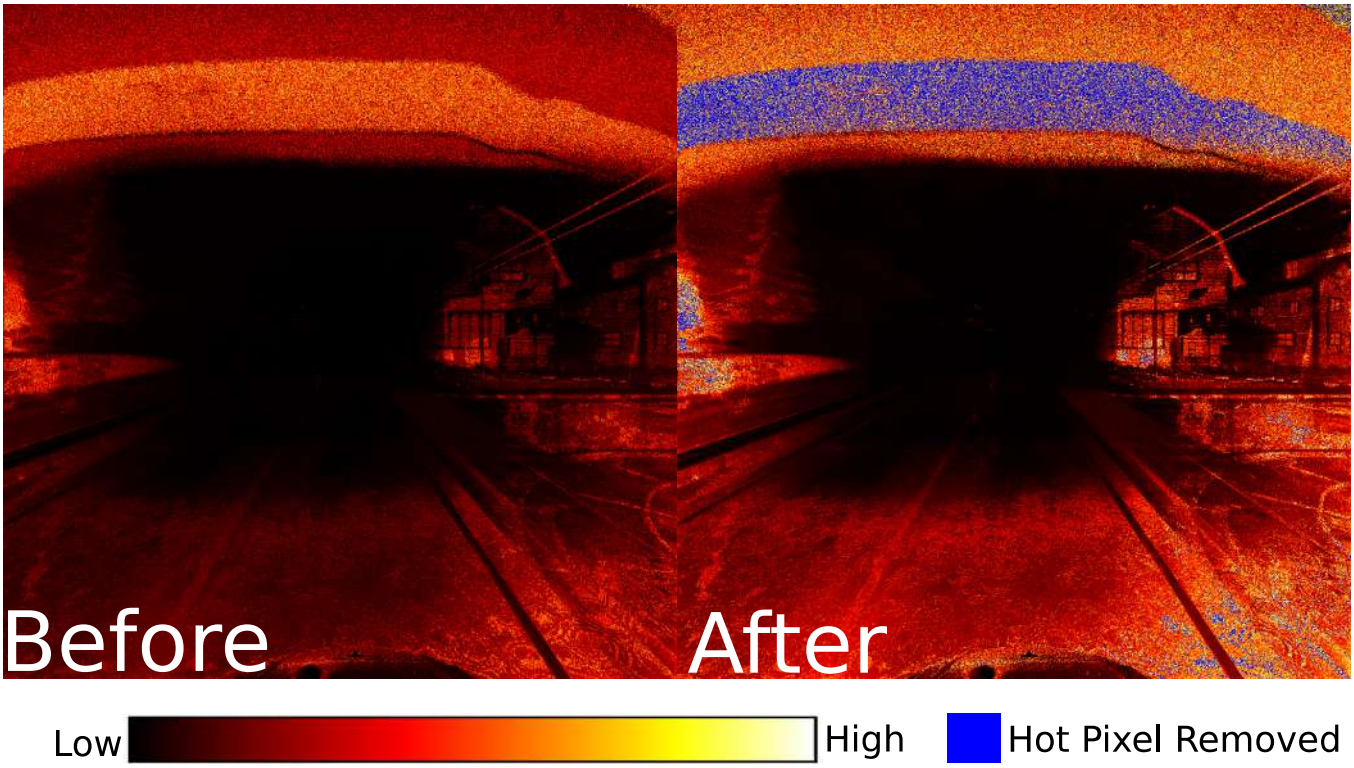


Figure 2.9: Histogram of Events before and after the hot pixels removal with 26.39% of events discarded, caused mainly by fog and rain puddles. Sample from `FULL_Dynamic_1` sequence.

to assess ESVO, the evaluation is run by down-sampling the rosbags playing time by a factor of 0.0005. This leads to high system latency during evaluations; for example, a 23 seconds sequence needs (23/0.0005) seconds to be evaluated, i.e., nearly 12 hours.

Despite this highly high system latency during evaluations, Figure 2.8 reports noticeably low ATE values compared to the other frame-based SLAM system. During EMVS [48] event-based mapping evaluations, we notice a significant map loss due to high fog as seen in sequence `Dynamic 1` of the FULL setup, or rapid motions as seen in sequence `Clear 2` of the ESVI setup.

The second scenario is dynamic weather, including fog and rain droplets that can cause random asynchronous events. Hot pixels in real-world DVS can be hardware defects, but simulated DVS can indicate random rain/fog firings in CARLA. Applying a hot pixel filter can detect and remove these unexpected events. Figure 2.9 shows a sample of the hot pixels removed due to fog and rain. Removing hot pixels in the DVS sensor is based on two criteria: the highest N pixels firing most events or the pixels firing greater than $n_\sigma \times \sigma$ events. n_σ is the event occurrence standard deviation multiplier, and σ is the event occurrence standard deviation.

The second corner case effect is witnessed during evaluating EMVS [48], where black dense blocks of point cloud points are accumulated on the trajectory during navigation in heavy rain and fog.

To evaluate IBISCape stereo-DVS calibration parameters, we construct stereo-RGB frames from the events using the E2VID pre-trained model [50]. We assess the ORB-SLAM3 stereo RGB SLAM system on these reconstructed

frames. Despite filtering the scene from noisy events resulting from fog and rain droplets, Table 2.12 shows a complete failure in trajectory estimation in the case of `Dynamic 1` sequence. Due to the dynamic weather conditions and rapid system dynamics, E2VID frame reconstruction fails with most IBISCape sequences. Consequently, another event-based frame reconstruction method is needed to consider these two corner case scenarios without losing the high dynamic range that DVS sensors can provide.

2.4.2.4 FULL Sensor Setup Evaluation

The most significant contribution of the IBISCape benchmark is its FULL sensor setup sequences, where all sequences contain a combination of all the available sensors simulated in clear/moderate/dynamic weather environments. As a result, a complete comprehensive quantitative evaluation of all the SLAM systems mentioned in the previous subsections can be compared on the same sequence for every specific weather condition, as seen in Figure 2.10. We represent in Table 2.12 an extensive qualitative assessment of the state-of-the-art SLAM systems based on the six FULL setup sequences. Regarding `Clear 1,2`, the trajectory estimation is aligned with the ground truth profile until a rapid motion occurs and the events map is disturbed. Each IBISCape FULL setup sequence is equipped with all the data formats as given with the specialized setup sequences.

Based on all the evaluation observations, we can conclude that the current pre-trained models to reconstruct frames can be unreliable specially in dynamic weather and large-scale environments as represented in Figure 5.8. This gives the most important advantage of IBISCape benchmark providing thousands of event arrays collected in a way to ease the retraining of the current models and motivates the development of new approaches to process events in such scenarios and corner cases.

The most prominent conclusion from evaluations on the FULL setup is that in outdoors dynamic weather where the dynamic range of the scene is considerably high, DVS sensor cannot be reliable to estimate the pose of the AGV with the current event-based SLAM systems. This conclusion is since events are fired asynchronously with high frequency, causing the visual sensor to be susceptible to weather constituents like rain or fog, which can degrade the estimation performance. Accordingly, our multi-modal datasets with the simulated corner cases can be the building block of choose-case scenarios for selecting the most efficient combination of multi-modal VI sensors for AGVs navigating in adverse conditions.

2.4.2.5 LiDAR Setup Evaluation

During the LiDAR based SLAM systems quantitative evaluation, we can observe significantly low RMS ATE values with MULLS systems for all the sequences compared to A-LOAM system as given in Table 2.12, and the lowest RMS ATE corresponds to MULLS with the loop closure option enabled. However, the RPE translation and rotation components slightly show a relatively lower values for A-LOAM compared to MULLS systems.

Figure 2.11 shows a more detailed qualitative evaluations, the efficient LiDAR point cloud registration method

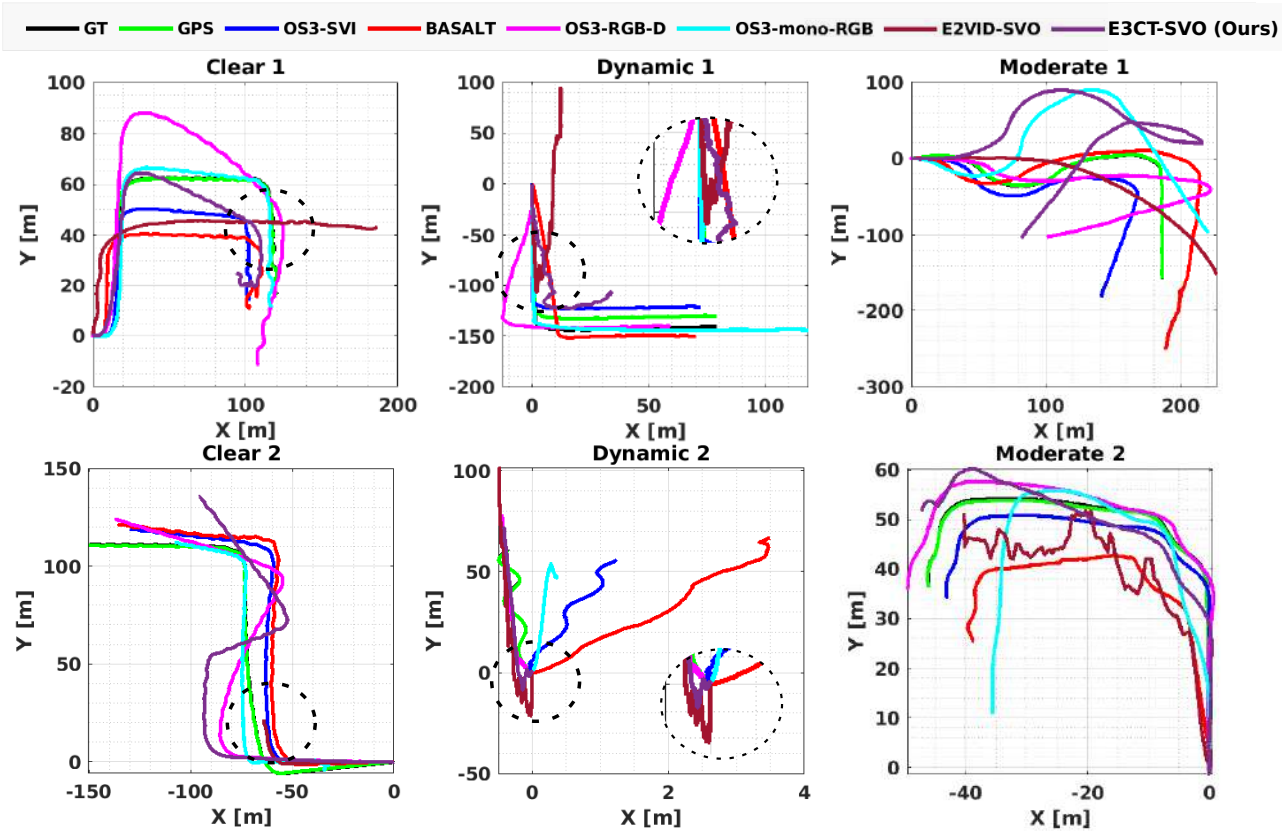


Figure 2.10: Trajectories estimated by ORB-SLAM3 and BASALT SLAM systems using IBISCape sequences with **FULL sensor setup** and comparing to their ground truth and GPS paths. ORB-SLAM3 algorithms involved are: Monocular RGB, Stereo-RGB (S-RGB) with E2VID and E3CT (Ours), Stereo Visual Inertial (SVI), and RGB-D SLAM systems. While for BASALT, the SVI algorithm is assessed.

(TEASER) [58] results are shown in red rectangles presenting the feature matching between two consecutive scans on the left and the global registration results on the right using the Neighborhood Category Context (NCC) encoding. In the blue colored rectangles, we shows NCC encoding results that provide an independent description of every feature extracted from the source and target scans without any additional computational operations that can increase the system latency.

2.4.2.6 Comparative Evaluation

To sum up all the latest state-of-the-art evaluations of nearly 80 experiments using the IBISCape benchmark, we provide a quantitative analysis of the mean value of all errors in Table 2.13. The average of experiments with the `E3CT-SVO` show error values that are considerably less than that of the `E2VID-SVO`. This gives an indication that future developments of event-based SLAM systems using the E3CT event representation method that can benefit from all the 3-channels information will result in low latency and high accuracy system. Then, in order to have a thorough quantitative comparison of all the methods, a weighted normalized accuracy parameter of all the SLAM systems evaluation parameters is proposed:

$$Accuracy = 0.5 \times \frac{ATE_{min}}{ATE_{method}} + 0.25 \times \frac{RPE^{Trans}_{min}}{RPE^{Trans}_{method}} + 0.25 \times \frac{RPE^{Rot}_{min}}{RPE^{Rot}_{method}} \; [ul]. \tag{2.5}$$

Weights are distributed with 50% for the RMS ATE values and 50% for the RPE values divided equally between translation and rotation error values. The SLAM system that provides the lowest ATE and RPE values will give an Accuracy = 1 which is the highest Accuracy value. This qualitative analysis is represented in Figure 2.12, where the SLAM system accuracy is compared to its system latency.

Since IBISCape benchmark targets a realistic simulation for the state-of-the-art SLAM systems evaluation, we compare the evaluation results on all the 31 IBIScape SLAM sequences with the real world publicly available datasets based on the RMS ATE values as given in their original papers in Figure 2.13. To ensure a fair comparison, E2VID results reported in our work can't be compared to that in [50], because the back-end VIO estimation method [3] using E2VID as a pre-processing layer is different than our evaluation back-end method (ORB-SLAM3/stereo-RGB).

The primary outcome of the IBISCape benchmark versus real-world benchmarks comparison is that the IBISCape dataset and the data acquisition APIs are designed to simulate outdoor environments that researchers can confidently use in their novel AGVs SLAM systems reliability evaluation in adverse weather dynamic environments. Furthermore, for reliable semantic SLAM systems, transfer-learning models from simulators to the real world are indispensable, especially in scenarios where real-world data is difficult to collect or in dangerous situations. IBISCape APIs can also provide high-end training and testing data for transfer-learning applications.

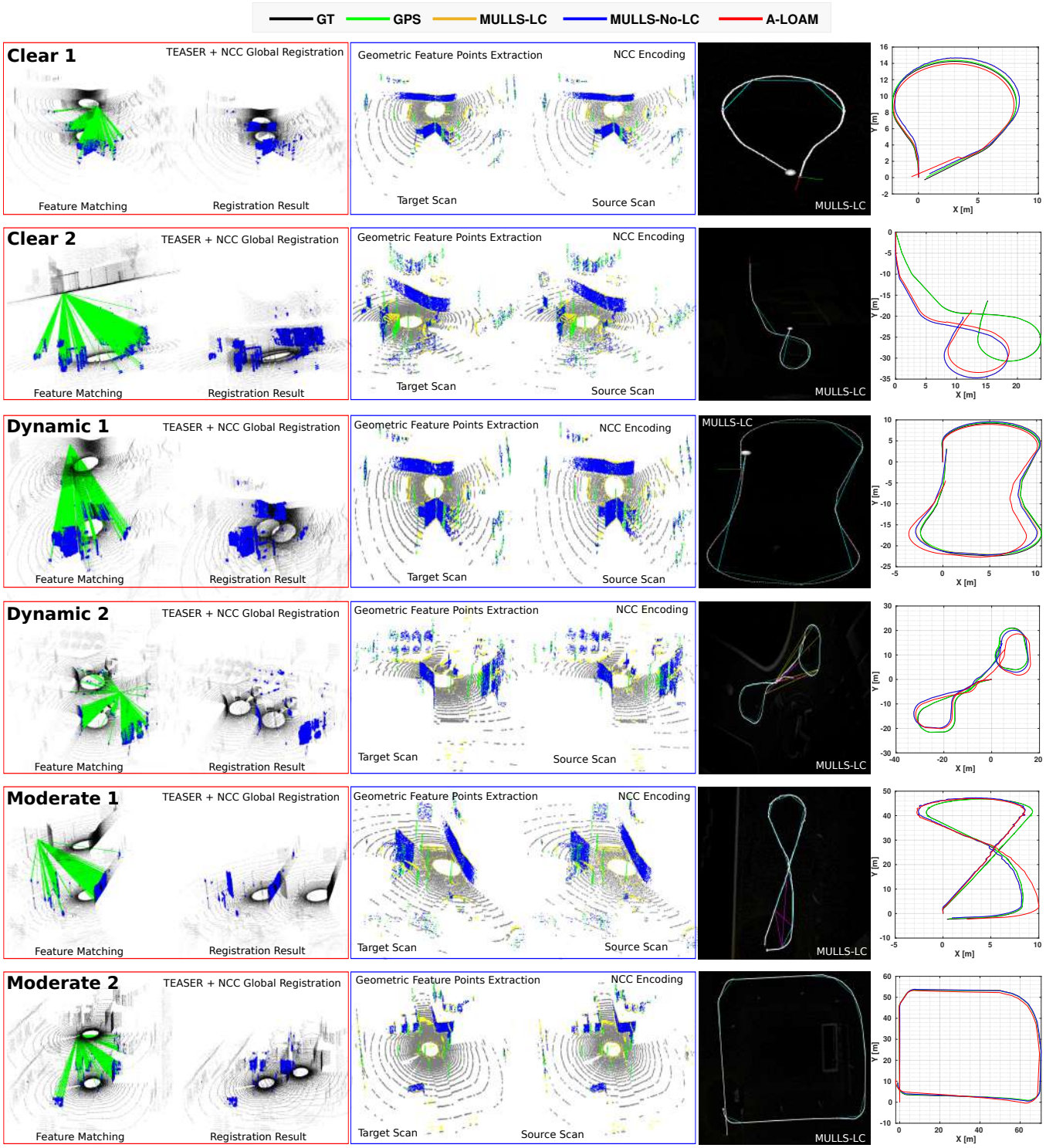


Figure 2.11: **LiDAR sensor setup** sequences qualitative detailed evaluation. From right to left: in red rectangle the point cloud features matching and global registration of two consecutive scans, in blue rectangle geometric feature points extraction, MULLS loop closure detection by Pose Graph Optimization (PGO), and trajectories estimated by MULLS and A-LOAM LiDAR Odometry/SLAM systems and comparison to their ground truth and GPS paths.

Table 2.12: ORB-SLAM3 (SVI, RGB-D, mono-RGB, stereo-RGB(E2VID, E3CT)), BASALT, MULLS and A-LOAM performance analysis based on both ATE and RPE evaluation metrics using IBISCape sequences in all simulated dynamic environments. Relative Pose Error (RPE) is formulated in terms of the mean ± standard deviation.

Sequence	Method 1			Method 2		
	ATE [m]	RPE [m]	RPE [deg]	ATE [m]	RPE [m]	RPE [deg]
FULL Setup - I	ORB-SLAM3 - SVI			BASALT		
Clear-1	**13.7184**	**0.2852±0.1765**	1.1677±1.2825	18.7082	0.3231±0.2047	**0.2675±0.5794**
Clear-2	12.3043	0.1234±0.1437	0.7866±0.8635	**12.0170**	**0.0655±0.0746**	**0.1699±0.3937**
Moderate-1	**32.8159**	**0.4248±0.0748**	**0.3838±0.3351**	49.9634	0.6076±0.2327	0.1420±0.7516
Moderate-2	**3.7829**	**0.1596±0.1356**	0.6265±0.9143	11.8746	0.2190±0.1318	**0.1645±0.4232**
Dynamic-1	17.2807	0.2584±0.1782	0.3845±0.6576	**16.6205**	**0.2433±0.1908**	**0.1222±0.3261**
Dynamic-2	**4.9187**	**0.1801±0.1520**	**0.0730±0.0607**	9.3084	0.2031±0.1757	0.1231±0.5529
FULL Setup - II	ORB-SLAM3 - RGB-D			ORB-SLAM3 - mono-RGB		
Clear-1	20.2653	0.3418±0.2003	**1.1702±1.2942**	**3.5142**	**0.3103±0.1804**	1.1723±1.2939
Clear-2	**14.8820**	**0.1659±0.1276**	**0.7914±0.8535**	18.4484	0.1759±0.1825	0.8240±0.8695
Moderate-1	**40.1021**	**0.4612±0.1130**	**0.4307±0.3410**	67.1074	0.4995±0.0667	0.4212±0.3720
Moderate-2	**3.5969**	**0.2595±0.1471**	**0.6520±0.9115**	15.0772	0.3040±0.2212	0.7999±0.9769
Dynamic-1	**11.5730**	**0.2048±0.1478**	0.3504±0.5771	22.2793	0.3090±0.2640	**0.3154±0.6171**
Dynamic-2	**15.5917**	**0.2824±0.2806**	0.1101±0.0955	17.2632	0.3210±0.4717	**0.0626±0.0679**
FULL Setup - III	E2VID-SVO			E3CT-SVO (Ours)		
Clear-1	84.7657	0.7384±0.8979	**0.6746±0.5510**	**70.9616**	**0.3383±0.8759**	1.1609±2.5091
Clear-2	156.8587	**0.2047±0.0446**	**0.3806±0.4775**	**103.4464**	0.3612±0.0052	0.8020±2.3298
Moderate-1	**157.9537**	1.3439±0.4314	**0.3565±0.1739**	203.4386	**0.6578±0.2343**	0.5997±1.6192
Moderate-2	**29.1791**	1.2812±1.0903	**0.1766±0.1263**	37.1249	**0.2946±0.5253**	0.6242±0.6628
Dynamic-1	235.7885	0.4666±0.3055	**0.0274±0.0733**	**91.2599**	**0.4045±0.0208**	0.3515±1.9189
Dynamic-2	52.1609	5.4907±6.1910	**0.1587±0.1005**	**36.0555**	**0.3870±0.9828**	0.2542±0.3042
SVI Setup	ORB-SLAM3 - SVI			BASALT		
Clear-1	**3.1262**	**0.1145±0.0728**	0.2699±0.4272	12.2769	0.1859±0.0234	**0.0839±0.4300**
Clear-2	**1.6666**	0.1076±0.0577	0.3646±0.4795	4.0626	**0.0514±0.0845**	**0.0919±0.2049**
Moderate-1	**11.5160**	**0.0496±0.0913**	0.1278±0.3385	70.1406	0.1290±0.0712	**0.1087±0.5707**
Moderate-2	**8.8561**	**0.3207±0.1681**	0.5831±0.7833	27.4657	0.3587±0.1527	**0.1519±0.3228**
Dynamic-1	50.7355	0.2580±0.1351	1.1411±1.0240	**12.4161**	**0.1853±0.1512**	**0.2324±0.4938**
Dynamic-2	**9.5503**	**0.1188±0.1445**	0.1338±0.4248	41.4773	0.2414±0.1277	**0.0921±0.5366**
Accident	16.2158	**0.2916±0.1594**	0.8453±1.9645	**2.6652**	0.4169±0.1158	**0.4808±0.9041**
RGB-D Setup	ORB-SLAM3 - RGB-D			ORB-SLAM3 - mono-RGB		
Clear-1	20.9667	**0.2370±0.1846**	**0.3830±0.5798**	**4.9536**	0.2522±0.1970	0.3842±0.5761
Clear-2	5.9339	**0.3647±0.2238**	0.3788±0.6832	**0.8387**	0.3706±0.2227	**0.3297±0.6424**
Moderate-1	**2.8882**	**0.2872±0.2419**	0.0718±0.0740	14.6609	0.2873±0.2562	**0.0290±0.0276**
Moderate-2	**13.5358**	0.2353±0.1597	0.2610±0.6043	29.3680	**0.2207±0.1559**	**0.2473±0.6575**
Dynamic-1	**8.7264**	0.2732±0.2626	**0.5988±0.8542**	15.1911	**0.2628±0.2426**	0.6079±0.8009
Dynamic-2	**12.0050**	**0.4743±0.1710**	0.5558±0.5380	121.4955	0.6201±0.3161	**0.5496±0.4548**
ESVI Setup	E2VID-SVO			E3CT-SVO (Ours)		
Clear-1	62.2875	0.6261±5.3078	**0.6011±0.4366**	**60.6766**	**0.4882±0.8758**	1.3745±1.2502
Clear-2	**121.0946**	**0.7864±10.2467**	**0.2974±2.5671**	169.6332	0.9917±0.1654	0.9103±2.2276
Moderate-1	**79.8216**	25.2978±43.9429	4.8828±15.0766	164.9004	**2.0066±0.2899**	**3.9804±4.2025**
Moderate-2	9.4286	9.4286±1.8231	**0.0323±0.1110**	**4.0446**	**4.6330±0.5663**	0.2109±0.1454
Dynamic-1	65.8318	0.4648±1.8205	**0.6281±0.4794**	**64.6726**	**0.2043±0.8541**	0.8514±0.6860
Dynamic-2	**106.0616**	3.8762±18.3748	1.9862±3.5864	109.9587	**0.9751±0.5803**	**0.9036±1.9564**
LiDAR Setup	MULLS (with/without Loop Closure)			A-LOAM		
Clear-1	**0.5593** 0.5881	0.1851±0.0572 0.1851±0.0572	1.4561±0.8493 1.4583±0.8472	1.0682	**0.1393±0.0292**	**1.0532±0.6091**
Clear-2	**5.5411** 5.5662	0.2121±0.0763 0.2122±0.0761	1.1231±0.9393 1.1221±0.9403	10.4762	**0.1431±0.0161**	**0.8152±0.6753**
Moderate-1	**0.9192** 0.9431	0.2061±0.0791 0.2061±0.0782	0.9073±0.9341 0.9063±0.9341	3.6931	**0.1501±0.0201**	**0.6811±0.6512**
Moderate-2	0.5391 **0.5371**	0.1861±0.1122 0.1871±0.1121	0.3613±0.7033 0.3611±0.7041	5.9511	**0.1531±0.0141**	**0.2761±0.5062**
Dynamic-1	0.5711 **0.5492**	0.1882±0.0693 0.1882±0.0693	1.1321±0.9351 1.1321±0.9351	2.3042	**0.1391±0.0171**	**0.8871±0.6581**
Dynamic-2	**1.6193** 1.6483	0.1821±0.0941 0.1821±0.0941	1.2051±0.9251 1.2051±0.9244	3.0251	**0.1401±0.0151**	**0.8712±0.6642**

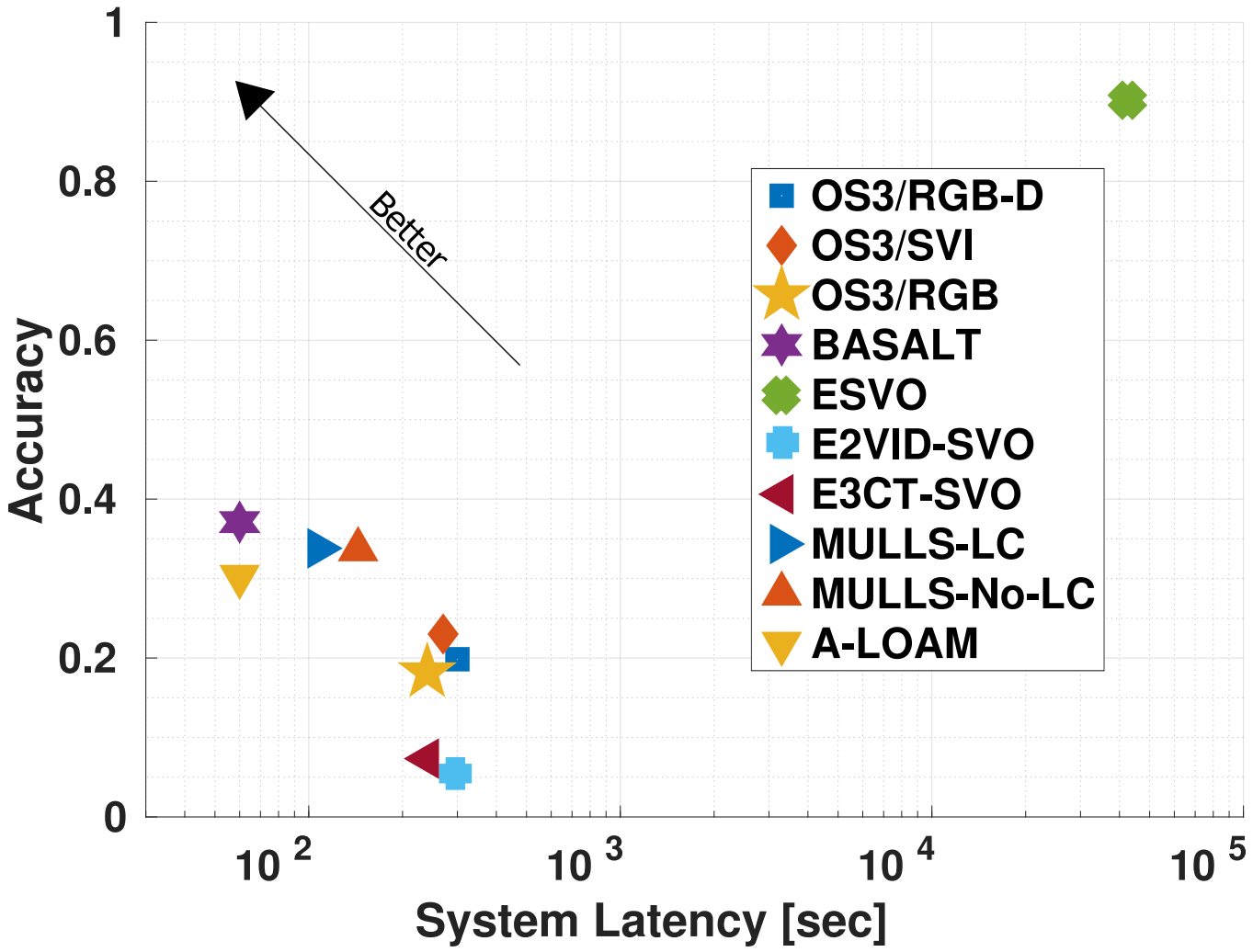


Figure 2.12: Semi-log Accuracy-Latency qualitative analysis of all SLAM systems undergoing evaluation on IBISCape sequences.

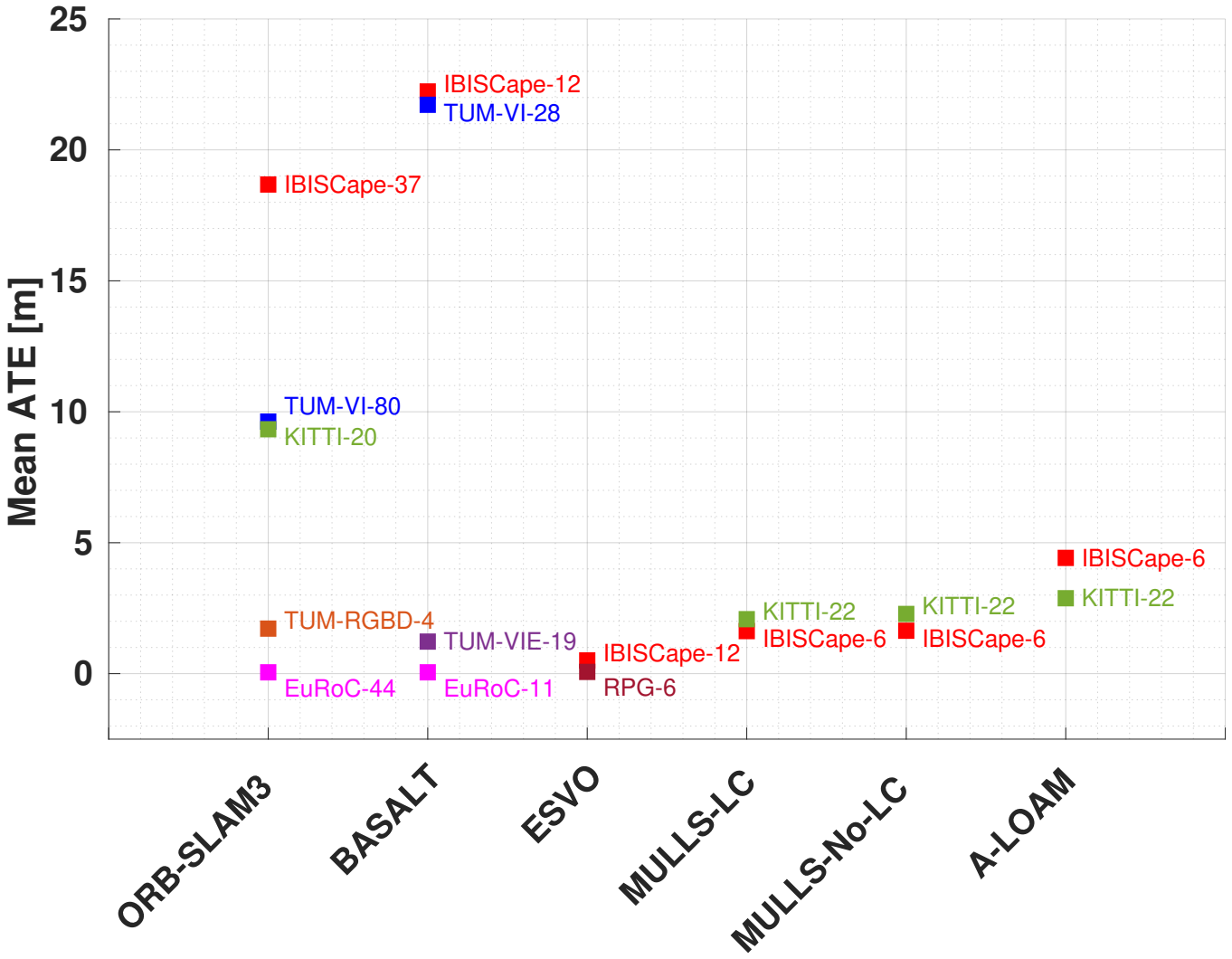


Figure 2.13: Mean of ATE values of evaluations using IBISCape benchmark compared to the real world benchmarks in literature. Marker: Benchmark-number of experiments. RPG: Clear weather & Static indoors scenes dataset [59]

Table 2.13: The average of all evaluation metrics for all experiments on the IBISCape benchmark.

Method	ATE [m]	RPE [m]	RPE [deg]	System
OS3/RGB-D [43]	14.1723	0.2989	0.4795	V-SLAM
OS3/SVI [43]	14.3452	0.2071	0.5298	VI-SLAM
OS3/RGB [43]	27.5165	0.3278	0.4786	V-SLAM
BASALT [44]	22.2305	0.2485	0.1716	VIO
ESVO [46]	0.5051	0.1090	0.3008	VO
MULLS(LC) [52]	1.6249	0.1933	1.0308	SLAM
MULLS(No-LC) [52]	1.6387	0.1935	1.0308	Odometry
A-LOAM [51]	4.4197	0.1441	0.7640	Odometry
E2VID-SVO [50]	96.7694	4.1671	0.8502	V-SLAM
E3CT-SVO (Ours)	93.0144	0.9785	1.0020	V-SLAM

2.5 Conclusion

This chapter proposes the IBISCape simulated heterogeneous sensors benchmark in large-scale dynamic environments along with 43 sequences suitable for multi-modal calibration & LiDAR/VI-SLAM evaluation. We also demonstrated new efficient algorithms for data synchronization during the acquisition process and a new iterative solution to estimate the unknown distortion coefficients of CARLA-simulated cameras. Using multiple adverse weather conditions, we have shown their impact on the latest state-of-the-art SLAM systems trajectory estimations.

A novel event-based pre-processing layer is presented based on the Event Spike Tensor representation called the Event 3-Channel Tensor (E3CT). This efficient model-based layer produces high dynamic range 3-channel event frames and is validated on multiple adverse conditions where it is witnessed to outperform other learning-based pre-trained models. Accordingly, E3CTs will open new paths for working on model-based multiple-channel event-based representations for more robust event-based SLAM systems.

The performance analysis includes a description of the sequence upon which the evaluation is done and the special conditions and corner cases simulated within every sequence to push the limits of the SLAM systems under assessment. The analytical study includes a comprehensive evaluation of the SLAM system performance and a quantitative comparison of ATE and RPE values. We hope this new dataset will help advance the research in the multi-modal heterogeneous sensors fusion applied to Autonomous Ground Vehicles (AGV) navigation in large-scale and dynamic environments.

As a future research trend, it will be indispensable to develop new efficient multi-modal: calibration and SLAM

algorithms based on the fusion of heterogeneous sensors with different caption and spectral technologies. That allows the SLAM system to estimate the trajectory better based on reliable continuous-time 3D scene mapping. Finally, an in-depth investigation is needed concerning the effect of map loss on SLAM systems estimations during long-term navigation in large-scale and dynamic weather environments.

3

Hybrid Online Calibration

Abstract

In this chapter, a new optimization-based method is presented for intrinsic and extrinsic calibration of an RGB-D-IMU visual-inertial setup, accompanied by a GPS-aided optimizer bootstrapping algorithm. The proposed method offers a reliable initialization of the RGB camera intrinsics and trajectory by utilizing an optical flow Visual Odometry (VO) technique. Additionally, it optimizes spatio-temporal parameters, including the target's pose, 3D point cloud, and IMU biases, in the back-end of the calibration process. The effectiveness of the method is demonstrated through extensive experimental evaluations conducted on both real-world and simulated sequences. These evaluations serve to validate the performance and accuracy of the calibration method in various scenarios.

"The best vision is insight."

Malcolm Forbes

3.1 Introduction

A reliable autonomous vehicle odometry solution relies on the continuous availability of the scene and vehicle information, such as scene structure and the vehicle's physical properties (position, velocity, or acceleration). These properties are measured by exteroceptive (Cameras/LiDAR/Radar/GPS) and proprioceptive (IMU/Wheel odometry) sensor modalities. Hence, multi-modal odometry algorithms have attracted the attention of many researchers in the last few years [60, 61, 62, 63], especially in challenging low structured environments.

Solutions incorporating a multi-camera system with no IMUs can be much easier to bootstrap using the 5-point [64] or the 8-point [65] SfM algorithms with a robust outlier filtration method [66, 67] without the need to estimate a global metric scale for the trajectory.

Adding an IMU (or multiple IMUs as in [53]) to a multi-camera calibration framework increases the complexity in the alignment process of the target's initial arbitrarily scaled poses with the initial real-world metric scaled ones [5]. In the recent work of [68], they studied a graph-based optimization approach that fuses GPS and IMU readings with stereo-RGB cameras. They show a superior estimation accuracy, especially in an offline operation, which is ideal for multi-modal calibration applications.

A well-known IMU-based bootstrapping method in the literature is described in [5], where the global metric scale and the IMU gravity direction are estimated using 4-DoF Pose Graph Optimization (PGO) augmented with the

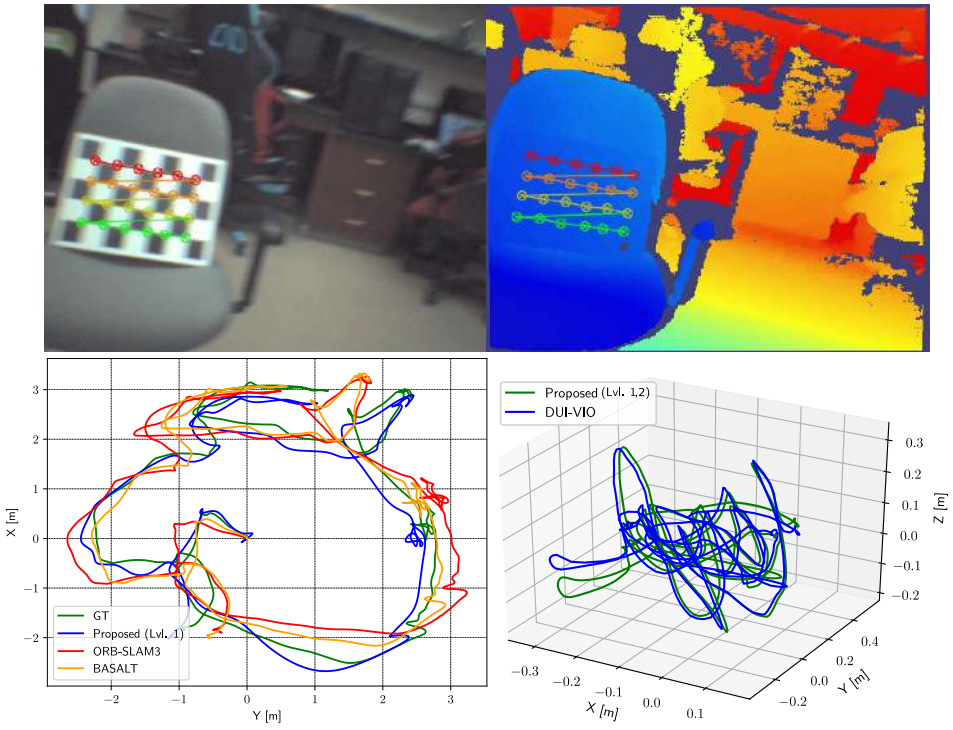


Figure 3.1: Our RGB-D-IMU setup calibration and pose estimation pipeline applied to the VCU-RVI hand-eye calibration sequence (top/bottom-right) and the EuRoC V2-01 sequence (bottom-left).

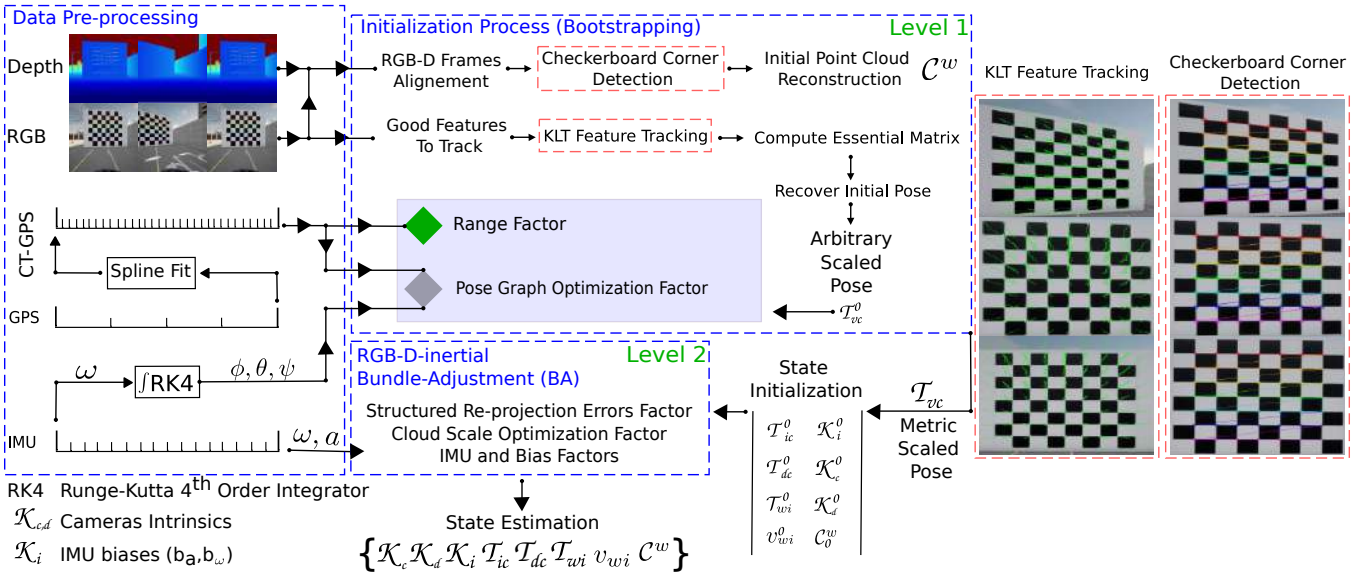


Figure 3.2: The pipeline of our method's front-end and back-end. The front-end is an initial data processing layer after acquiring RGB-D aligned frames. The back-end is the central processing layer of two optimization levels.

IMU preintegration factors. We tackle this scaling problem with a novel method that can be applied online, where low-rate noisy GPS signals can be detected with a 6-DoF PGO and a 3-DoF range factor. These instant initialization factors solve the prominent initialization failure problem due to insufficient IMU excitation resulting in a reliable pose estimation algorithm (see Figure 3.1).

The visual-inertial bundle adjustment (BA) [43, 69] is a highly non-linear process, primarily when there exists an unconventional visual sensor (depth camera, for instance) with a different spectral technology than that of the RGB camera within the multi-modal calibration framework. The accuracy and robustness of the calibration process are thoroughly dependent on the estimator initialization, which we perform using front-end, and back-end (level 1) steps represented in the pipeline in Figure 3.2. Towards a reliable RGB-D-IMU calibration and GPS-aided poses estimation solution, we sum up our main contributions as threefold:

- A novel method for bootstrapping the global metric scale for a visual-inertial BA optimization problem with a prior level of pose graph optimization that relies on noisy low-rate GPS readings combined with gyroscope measurements.

- A novel point cloud scale optimization factor that integrates the untextured depth maps having no distinctive features in a visual-inertial BA as any conventional camera in a stereo-vision setup by a double re-projection with distortion function.

- A robust multi-modal calibration algorithm for RGB-D-IMU sensors setup with a reliable metric scaled 3D pose estimation methodology easily extended to a multi-modal RGB-D-IMU-GPS odometry algorithm.

3.2 Related Work

Multi-modality has become the mainstream of most recent calibration works [70, 71, 72, 73] because an efficient multi-modal odometry solution depends on an optimally calibrated system. In this work, we propose a baseline robust method to calibrate RGB-D-IMU full system parameters considering efficient performance regarding latency, accuracy, and configuration robustness.

3.2.1 RGB-D-IMU Calibration

Over the recent years, RGB-D calibration algorithms [74, 75, 76, 77, 78] have evolved to incorporate various depth correction strategies based on an extra stage of an on-manifold optimization. The works [74, 75, 76] correct depth with an exponential undistortion parametric curve fitting, while others [77, 78] fit the point cloud on a sphere. Adding an IMU sensor to an RGB-D calibration setup is a configuration tackled in the works of [79] and [80] using Extended Kalman Filters (EKFs). However, these RGB-D-IMU calibration works mainly aim to estimate the pose and perform IMU/CAM extrinsic calibration during the odometry task.

3.2.2 RGB-D-IMU Odometry

Inspired by the pipeline of VINS-Mono [5], we tackle the lack of insufficient IMU excitation in the bootstrapping process by incorporating the low-rate noisy GPS readings in a novel approach. The RGB-D Visual-Inertial Odometry (VIO) works [79, 81, 82, 83, 80], report two ways to state estimation for an RGB-D camera-based VIO. The first is to compute the pose change using VO and fuse the estimated pose change with the IMU's preintegration [84, 85]. Another way is to compute the visual features' 3D locations using depth measurements and an iterative approach to reduce the features' re-projection and the IMU's preintegration factors [86, 87].

In the iterative optimization process, existing approaches utilizing either scheme assume a precise depth measurement and consider the depth value of a visual feature as a constant [86, 87]. However, an RGB-D camera's depth measurement may have a high uncertainty level [88], resulting in considerable error values in the odometry state estimation if ignored. The work in [89] incorporates a learning-based dense depth mapping method and performs a filter-based approach for navigation state estimation.

Our work can be considered the first optimization-based RGB-D-IMU complete system calibration with a novel depth correction model that does not require a separate optimization stage to fit the depth map on a high-order parametric curve or surface. The robustness of our method conforms to the works [90, 42], which can be summed up in three main points: minimum information is needed to efficiently bootstrap the system, overcome inertial and celestial sensors limitations during the initialization process, and efficient measurements outlier rejection [66].

3.3 Methodology

This section presents a sequential overview of the proposed hybrid visual odometry with online calibration method. Section 3.3.1 gives a brief overview of the on-manifold rigid body kinematics. In Section 3.3.2, we start by collecting the target's poses (up-to-scale) as well as the checkerboard corners and construct an initial point cloud of the collected corners (see Figure 3.2 (top)). Then in Section 3.3.3, we bootstrap the optimizer with GPS and gyroscope readings for instant metric scale estimation of the estimated target's poses. Finally, Section 3.3.4 presents the tightly-coupled hybridization factors to calibrate the full RGB-D-IMU sensor setup in a non-linear BA optimization process.

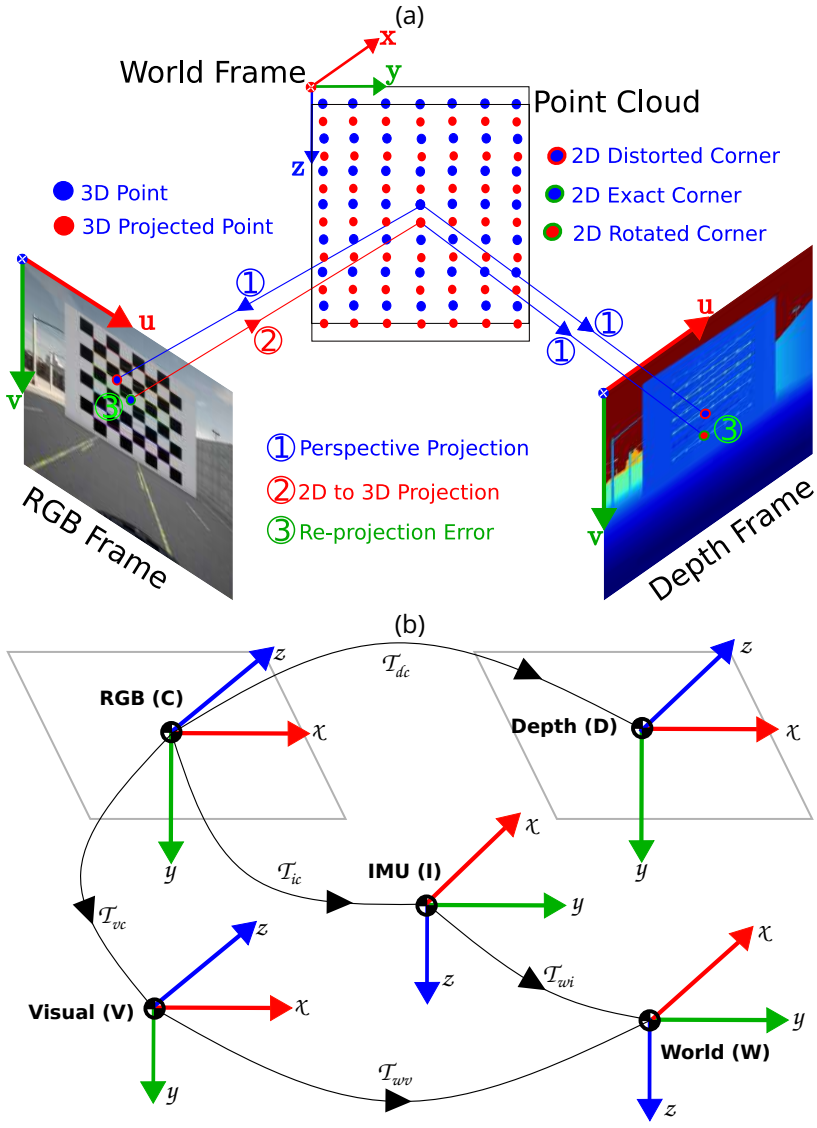


Figure 3.3: Illustration for the re-projection error factors on both RGB and Depth frames, as well as the coordinate frames for all sensors undergoing optimization: (a) 3D to 2D and 2D to 3D to 2D re-projection error for triangulating the same target's 3D corner on both the RGB-D current aligned frames; (b) Coordinate frame of reference for all sensors undergoing the calibration with respect to the world frame. For consistency: all frames follow the right-handed rule as OpenCV library.

3.3.1 Trajectory Rigid Body Kinematics

Visual-inertial odometry task for autonomous vehicles in challenging environments needs efficient real-time processing algorithms. In this context, we need to optimize the estimated vehicle pose on a continuous-time manifold. This manifold should allow the estimation of a continuous-time trajectory of the pose $T(t)$, the velocity $\dot{T}(t)$, and the acceleration $\ddot{T}(t)$ in the $\mathrm{SE}(3)$ Lie group representation for fast and accurate calculations. Choosing a continuous-time manifold allows us to fuse sensors of different types of data being processed with variable frequencies (for example: cameras, LiDARs: 15-30 Hz, IMUs: 100-250 Hz, and GPS: 1 Hz).
B-splines are widely used in manifold modeling due to their ability to handle complex geometries and their computational efficiency. In recent years, several works have been published that focus on developing B-spline manifold models and their derivatives. [91] proposed a spline-based trajectory representation in $\mathrm{SE}(3)$ that can be used to fuse information from different sensors, generate inertial and visual predictions, and even demonstrate self-calibration of a visual-inertial system. In 2017, Ethan Eade's research notes presented complete modeling for the operators, operations, and Jacobians for 2D and 3D transformations. The following year, Patrick Geneva's research notes introduced complete modeling for time derivatives of the B-spline.

[92] published a paper on a micro Lie theory for state estimation in robotics, which is accompanied by a new open-source C++ header-only library called manif. This library implements the widely used groups $\mathrm{SO}(2)$, $\mathrm{SO}(3)$, $\mathrm{SE}(2)$, and $\mathrm{SE}(3)$, with support for the creation of analytic Jacobians, designed for ease of use, flexibility, and performance. Finally, in 2020, Sommer proposed a simple formulation for the time derivatives of Lie group cumulative B-splines that require several matrix operations, which scale linearly with the order k of the spline [93]. These works collectively provide a comprehensive understanding of B-spline manifold modeling along with its derivatives, which are crucial for various applications, including robotics, trajectory planning, and sensor fusion.

For this approach to be practical in a visual-inertial odometry system as well as serve as a data fusion framework for other sensors, it should include specific characteristics:

1. Local control, i.e. change in one segment doesn't affect whole the trajectory allowing the system to function online as well as in batch.

2. C^2 continuity, the temporal derivatives enable inertial predictions.

3. Application of rigid-body motion kinematics free of singularities.

The representation for continuous trajectories in $\mathrm{R}(3)$ is well-known using B-Splines. As B-Splines provide local control, and cubic B-Splines are C^2 continuous in $\mathrm{R}(3)$. However, dealing with 3D rotations using B-Splines isn't an easy task, such as interpolation in $\mathrm{SO}(3)$. The different modeling domains are represented in Table 3.1.

Some methods for interpolating rotations, such as piecewise Spherical Linear Interpolation, SLERP [94], affected by discontinuities, while Spherical Quadratic Interpolation, SQUAD [95], does not preserve C^2 continuity [96]. More interpolation methods for rotations are provided in [97].

In this thesis, we choose to parameterize our continuous-time trajectory using cumulative basis functions formed using the Lie Algebra $\mathfrak{se}(3)$ of the matrix group $\mathrm{SE}(3)$ modeled in 3.3.1.1, equivalent to that proposed by [98].

This choice is based on two primary factors:

1. Using cumulative B-spline basis functions is not only C^2 continuous, but it also provides a very simple second derivative formulation useful for generating inertial predictions.
2. The Lie Algebra parameterization, when applied locally, is free from any singularities and offers a very good analytical approximation to minimum torque trajectories.

3.3.1.1 Trajectory Modeling

Table 3.1: Different Modeling Domains

Space	Definition	Model	Increments Δ
$\mathrm{R}(3)$	3D Euclidean space (Translations or Euler angles)	$t_{3\times3}, R_{3\times3}$	$\mathrm{R}(3) \rightarrow \nu, \omega$
$\mathrm{SO}(3)$	"Special Orthogonal Matrix" is used to describe the possible rotational symmetries of an object, as well as the possible orientations of an object in space.	$R_{3\times3}$	$\mathfrak{so}(3) \rightarrow \lfloor\omega\rfloor_\times$
$\mathrm{SE}(3)$	A “Special Euclidean Transformation” that is a differentiable manifold is called a Lie group	$T_{4x4} = \begin{bmatrix} R_{3\times3} & t_{3\times3} \\ 0_{3\times3} & 1 \end{bmatrix}$	$\mathfrak{se}(3) \rightarrow \begin{bmatrix} \lfloor\omega\rfloor_\times & \nu \\ 0_{3\times3} & 1 \end{bmatrix}$

3.3.1.2 Cumulative B-spline modeling in R(3)

Matrix form of the B-spline segment can be generated depending on the degree of the spline needed and the number of the control points (poses) defining it.

We need to define some key points -symbols- first, represented in Figure 3.4:

- (k) is the order of the spline ex. for cubic $k = 4$ and quadratic $k = 3$, i.e. order = degree + 1
- (n) is equal to the number of control points - 1
- $p(t)$ is the spline segment for interval of $t = [t_i, t_{i+1}]$, where i is the pose number

- Each segment in $p(t)$ is defined by 2 knots; one at each end

- The number of knots defining all the spline segments incremented for all given control points (poses) is equal to $m+1$, where $m = k + n + 1$

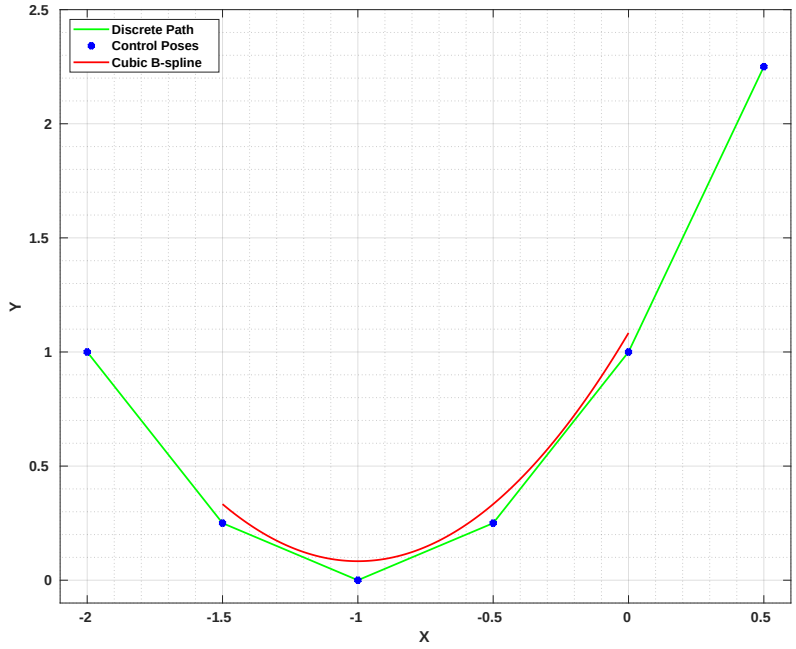


Figure 3.4: Continuous-time B-splines compared to Discrete-time Trajectory

The non-cumulative B-spline basis function is $B^{(k)}$, is a blending matrix with entries calculated using [99] recursive formula.

$$b_{s,n}^{(k)} = \frac{C_{k-1}^{n}}{(k-1)!} \sum_{l=s}^{k-1} (-1)^{l-s} C_{k}^{l-s} (k-1-l)^{k-1-n} \tag{3.1}$$

where $s, n \in \{0, ..., k-1\}$are the numbers of $B^{(k)}$ rows and columns respectively.

At time $t \in [t_i, t_{i+1}]$ the value of $p(t)$ only depends on the control points $t_i, t_{i+1}, ..., t_{i+k-1}$. To simplify calculations, we transform time to a uniform representation $s(t) := (t - t_0)/\Delta t$, such that the control points transform into $0, .., k-1$.

We define $u(t) := s(t) - i$ as normalized time elapsed since the start of the segment $[t_i, t_{i+1}]$ and from now on use u as temporal variable. i.e. $u = 0 : 1$, with certain precision.

Generalizing

$$\bar{u}^{(k)} = \begin{bmatrix} u^0 \\ \vdots \\ u^{k-1} \\ u^k \end{bmatrix} \tag{3.2}$$

The value of $p(u)$ can then be evaluated using a matrix representation as follows:

$$p(u) = \begin{bmatrix} p_i & p_{i+1} & \cdots & p_{i+k-1} \end{bmatrix} B^{(k)} \bar{u}^{(k)} \tag{3.3}$$

The cumulative B-spline matrix form in the $\mathrm{R}(3)$ can be modeled as:

$$p(u) = \begin{bmatrix} p_i & d_1^i & \cdots & d_{k-1}^i \end{bmatrix} \widetilde{B}^{(k)} \bar{u}^{(k)} \tag{3.4}$$

with the cumulative basis function matrix $\widetilde{B}^{(k)}$, is a blending matrix with entries

$$\widetilde{b}_{j,n}^{(k)} = \sum_{s=j}^{k-1} b_{s,n}^{(k)} \tag{3.5}$$

with j, is the number of row the accumulation of $b_{s,n}^{(k)}$ elements start from.

and difference vectors $d_j^i = p_{i+j} - p_{i+j-1}$, for poses (translations and rotations).

Definition 3.1

The B-spline of order k at position u can be written as

$$p(u) = p_i + \sum_{j=1}^{k-1} \widetilde{B_j}^{(k)} \bar{u}_j^{(k)} d_j^i \tag{3.6}$$

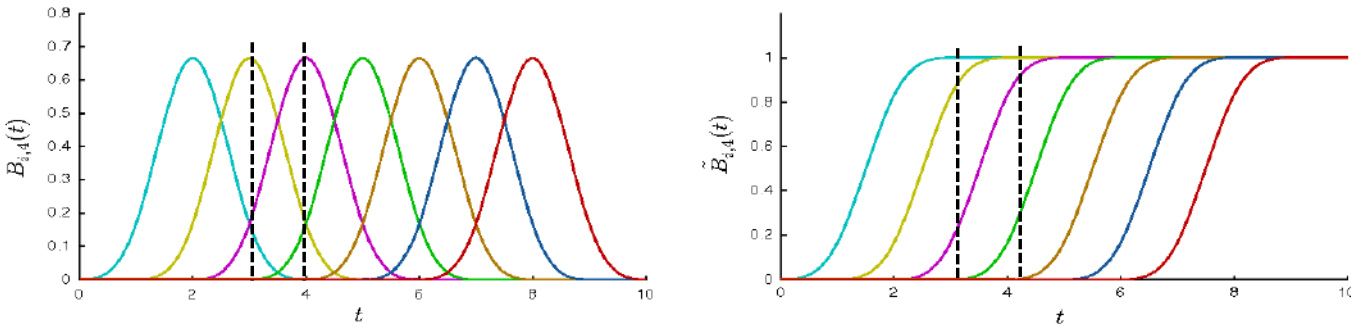
3.3.1.3 Cumulative B-spline modeling in SO(3)

$$\mathrm{SO}(3) = \left\{ R | R \in \mathrm{SO}(3), R^\top R = R R^\top = I_{3\times 3}, |R| = +1 \right\}$$

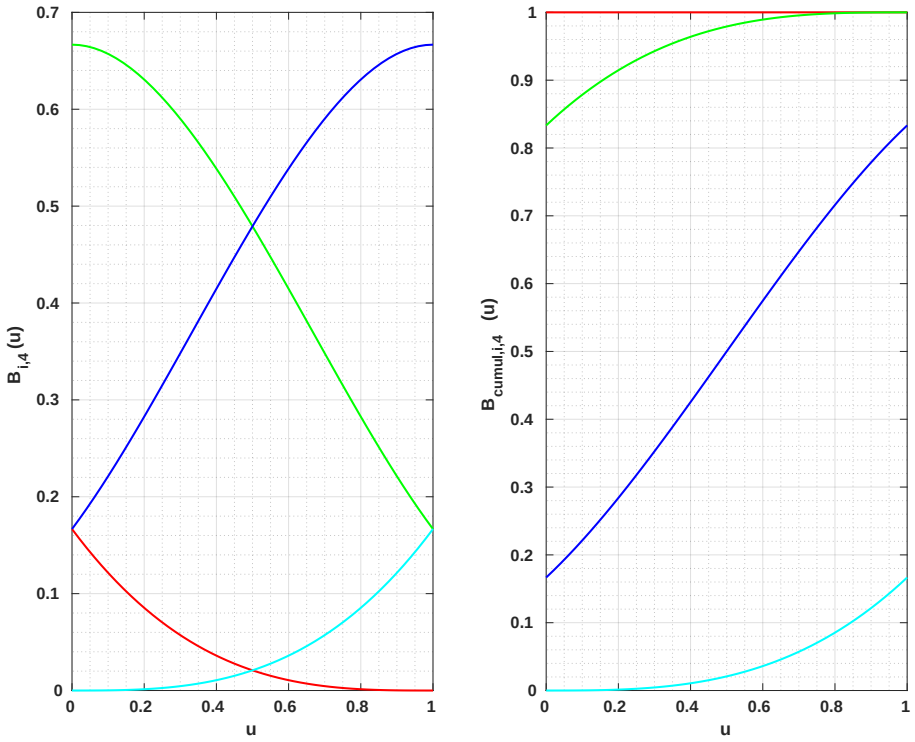
Definition 3.2

The cumulative B-spline of order k in a Lie group $\mathrm{SO}(3)$ with control points $R_0, \cdots, R_N \in \mathrm{SO}(3)$ has the form

$$R(u) = R_i \prod_{j=1}^{k-1} exp(\widetilde{B_j}^{(k)} \bar{u}_j^{(k)} d_j^i) \in \mathrm{SO}(3) \tag{3.7}$$



(a) Standard Basis Function Illustration [91]. On the left the normal basis function on the right the cumulative one.



(b) Our simulation for the validity region for every spline segment in both cases.

Figure 3.5: Non-cumulative and cumulative Basis Functions

with the generalized difference vector d_j^i

$$d_j^i = log(R_{i+j-1}^{-1} R_{i+j}) = \lfloor \omega \rfloor_\times \in \mathfrak{so}(3) \tag{3.8}$$

With the definition of the exponential map $exp(\mathfrak{so}(3)) \to \mathrm{SO}(3)$

$$exp(\lfloor \omega \rfloor_\times) = I_{3\times 3} + \frac{sin(\|\omega\|)}{\|\omega\|} \lfloor \omega \rfloor_\times + \frac{1 - cos(\|\omega\|)}{\|\omega\|^2} \lfloor \omega \rfloor_\times^2 \in \mathrm{SO}(3) \tag{3.9}$$

where $\lfloor \omega \rfloor_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$, is the skew-symmetric matrix of the 3 rotations increments.
Along with the definition of the logarithmic map $log(\mathrm{SO}(3)) \to \mathfrak{so}(3)$

$$log(R_d) = \frac{\theta}{2sin\theta}(R_d - R_d^\top), \quad with \quad \theta = cos^{-1}\left(\frac{trace(R_d) - 1}{2}\right) \in \mathfrak{so}(3) \tag{3.10}$$

where $R_d = R_{i+j-1}^{-1} R_{i+j}$, $R^{-1} = R^\top$.

3.3.1.4 Cumulative B-spline modeling in SE(3)

$$\mathrm{SE}(3) = \left\{ T | T = \begin{bmatrix} R & t \\ 0_{3\times 3} & 1 \end{bmatrix}, R \in \mathrm{SO}(3), t \in \mathrm{R}(3), R^\top R = R R^\top = I_{3\times 3}, |R| = +1 \right\}$$

Definition 3.3

The cumulative B-spline of order k in a Lie group [91] $\mathrm{SE}(3)$ with control points $T_0, \cdots, T_N \in \mathrm{SE}(3)$ has the form

$$T(u) = exp(\widetilde{B}_0^{(k)} \bar{u}_0^{(k)} log(T_0)) \prod_{j=1}^{k-1} exp(\widetilde{B}_j^{(k)} \bar{u}_j^{(k)} d_j^i) \in \mathrm{SE}(3) \tag{3.11}$$

with the generalized twist vector d_j^i

$$d_j^i = log(T_{w,i+j-1}^{-1} T_{w,i+j}) = \begin{bmatrix} \omega \\ \nu \end{bmatrix} \in \mathfrak{se}(3) \tag{3.12}$$

With the definition of the exponential map $exp(\mathfrak{se}(3)) \to \mathrm{SE}(3)$

$$exp\left(\begin{bmatrix} \omega \\ \nu \end{bmatrix}\right) = \begin{bmatrix} exp(\lfloor \omega \rfloor_\times) & V\nu \\ 0_{3\times 3} & 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0_{3\times 3} & 1 \end{bmatrix} = T \in \mathrm{SE}(3) \tag{3.13}$$

with

$$V = I_{3\times3} + \frac{1 - cos(\|\omega\|)}{\|\omega\|^2} \lfloor\omega\rfloor_\times + \frac{\|\omega\| - sin(\|\omega\|)}{\|\omega\|^3} \lfloor\omega\rfloor_\times^2 \tag{3.14}$$

Along with the definition of the logarithmic map $log(\mathrm{SE}(3)) \rightarrow \mathfrak{se}(3)$

$$log(T_d) = \begin{bmatrix} log(R_d) \\ V^{-1}t_d \end{bmatrix} \in \mathrm{R}^6 \tag{3.15}$$

where $T_d = T_{i+j-1}^{-1} T_{i+j}$, $T^{-1} = \begin{bmatrix} R^\top & -R^\top t \\ 0_{3\times3} & 1 \end{bmatrix}$, $T_1T_2 = \begin{bmatrix} R_1R_2 & R_1t_2 + t_1 \\ 0_{3\times3} & 1 \end{bmatrix}$.

3.3.1.5 Trajectory Temporal Derivatives in SE(3)

Inertial Predictions : Spline as a Generative Model

The ability to calculate the analytical derivative of the B-spline, enables us to calculate the velocity and acceleration in a continuous-time manner. This gives us a huge plus in trivially synthesizing the IMU measurements of the Gyroscope and the Accelerometer readings. Accordingly the IMU biases can be calculated precisely for every IMU step by setting their residuals equals to zero.

The accelerometer and gyroscope residuals can be defined as:

$$r_\omega(u) = \omega(u) - \hat{\omega} + b_\omega \tag{3.16}$$

$$r_a(u) = R_{wi}(u)^{-1}(a_{wi}(u) + g) - \hat{a} + b_a \tag{3.17}$$

where $g = \begin{bmatrix} 0 & 0 & -9.80665 \end{bmatrix}$ m/sec^2 is the gravity vector in world coordinates. $\hat{a}$ & $\hat{\omega}$ are the IMU measurements of the accelerometer and gyroscope respectively.

For $\mathrm{SE}(3)$, $\omega(u)$ is the angular velocity calculated and $\ddot{T}_{wi}(u)$ is the translation vector of the second time derivative of the pose computed in 3.11. The $\mathrm{SE}(3)$ formulation of these residuals is identical to that in both [91, 93].

Baseline Method

[91] worked on a cumulative cubic B-spline model ($k = 4, n = 3$) to represent the trajectory. With the control point separated with Δt on a uniform time intervals.

Starting with the pose T form in (3.11), we differentiate once we get this general model for B-spline of order (n) is proposed by [93]:

$$\dot{T}(u) = T_i \sum_{j=1}^{k-1} \left[\left(\prod_{l=1}^{j-1} A_l(u) \right) \dot{A}_j(u) \left(\prod_{l=j}^{k-1} A_l(u) \right) \right] \tag{3.18}$$

Expanding the first and second derivatives for cubic B-splines with $n = 3$:

$$\dot{T}(u) = T(u)(\dot{A}_0 A_1 A_2 + A_0 \dot{A}_1 A_2 + A_0 A_1 \dot{A}_2) \tag{3.19}$$

$$\ddot{T}(u) = T(u)(\ddot{A}_0 A_1 A_2 + A_0 \ddot{A}_1 A_2 + A_0 A_1 \ddot{A}_2 + 2\dot{A}_0 \dot{A}_1 A_2 + 2\dot{A}_0 A_1 \dot{A}_2 + 2A_0 \dot{A}_1 \dot{A}_2) \tag{3.20}$$

with

$$\dot{A}_j(u) = \widetilde{B}_j \dot{\tilde{u}}_j A_j(u) D_j = \widetilde{B}_j \dot{\tilde{u}}_j D_j A_j(u) \tag{3.21}$$

and

$$\ddot{A}_j(u) = \widetilde{B}_j \ddot{\tilde{u}}_j A_j(u) D_j + \widetilde{B}_j \dot{\tilde{u}}_j D_j \dot{A}_j(u) \tag{3.22}$$

and $D_j = (d_j)_\wedge = \begin{bmatrix} 0 & -\omega_z & \omega_y & \nu_x \\ \omega_z & 0 & -\omega_x & \nu_y \\ -\omega_y & \omega_x & 0 & \nu_z \\ 0 & 0 & 0 & 0 \end{bmatrix}$, $\dot{\tilde{u}}^{(k)} = \frac{1}{\Delta t} \begin{bmatrix} (i-1)u^{i-1} \end{bmatrix}$, $\ddot{\tilde{u}}^{(k)} = \frac{1}{\Delta t^2} \begin{bmatrix} (i-1)^2 u^{i-2} \end{bmatrix}$,

with $i = 1 : k$. Noting in (3.21),(3.22) that $A_j(u)$ and D_j are commuting by definition.

Generalizing the second order derivative for any order (n) B-spline, we contributed with the following :

1. The first **3 terms** in (3.20), can be modeled using (3.18) with replacing $\dot{A}_j(u)$ with $\ddot{A}_j(u)$.

2. The last **3 terms** are always formed in pairs of 2 derived A's in the same expression, for any order spline. A look-up table is constructed to visually represent the selected the pairs of "A" terms to be derived which increased the computational speed significantly:

   For B-spline with n=4 the terms are ($A_0 A_1 A_2 A_3$), the pairs are:

counter i=1:n-1	counter j=1:n-i	First Derivative term	Second Derivative term	Resulting term
1	1	1	2	$\dot{A}_0 \dot{A}_1 A_2 A_3$
	2	1	3	$\dot{A}_0 A_1 \dot{A}_2 A_3$
	3	1	4	$\dot{A}_0 A_1 A_2 \dot{A}_3$
2	1	2	3	$A_0 \dot{A}_1 \dot{A}_2 A_3$
	2	2	4	$A_0 \dot{A}_1 A_2 \dot{A}_3$
3	1	3	4	$A_0 A_1 \dot{A}_2 \dot{A}_3$

(a) for n=4, we have 3 $\rightarrow (n-1)$ colored groups starting from 1 to $n-1$. (The outer *for* loop)

(b) Also, to form this table we need 2 for loops, the first loops on (i) the second loops on (j) and a counter (c) to fill each row in this table.

(c) We can conclude that in our loops, the first column is the (i) iterator value in every loop, while the second column is (i+j).

(d) Completing this table will fill the selector table of n choose 2, C_2^n rows and 2 columns.

The Inertial terms in Equations (3.16), (3.17) can be modeled as:

1. The Angular velocity term $\lfloor \omega(u) \rfloor_\vee$:

$$\begin{bmatrix} \lfloor \omega(u) \rfloor_\wedge & v(u) \\ 0_{3\times 3} & 1 \end{bmatrix} = T(u)^{-1}\dot{T}(u), \quad V(u) = R_{wi}(u)v(u)$$

where $V(u), \omega(u)$ are the linear and angular velocity terms.

This step can be simplified by directly using the term:

$$\begin{bmatrix} \lfloor \omega(u) \rfloor_\wedge & v(u) \\ 0_{3\times 3} & 1 \end{bmatrix} = \sum_{j=1}^{k-1} \left[\left(\prod_{l=1}^{j-1} A_l(u) \right) \dot{A}_j(u) \left(\prod_{l=j}^{k-1} A_l(u) \right) \right]$$

2. The Linear Acceleration term $\ddot{a}(u)$:

$$\begin{bmatrix} \lfloor \dot{\omega}(u) \rfloor_\wedge & \ddot{s}(u) \\ 0_{3\times 3} & 1 \end{bmatrix} = \left[T(u)^{-1}\ddot{T}(u) \right] - \begin{bmatrix} \lfloor \omega(u) \rfloor_\wedge & v(u) \\ 0_{3\times 3} & 1 \end{bmatrix}^2, \ddot{a}(u) = \ddot{T}(1:3,4)$$

- where $\ddot{a}(u), \dot{\omega}(u)$ are the linear and angular acceleration terms, respectively.

Efficient Method

In the work of [93], the proposed formulation improved the performance instead of having a matrix-matrix multiplication complexity of $(k-1)^2+1$ in case of first derivative formula. And $\frac{1}{2}k^2(k-1)$ in case of second derivatives formula.

The first derivative formula is recursively defined by the relations:

$$\begin{bmatrix} v(u) \\ \lfloor \omega(u) \rfloor_\vee \end{bmatrix}^{(j)} = Adj_{A_{j-1}^{-1}} \omega^{(j-1)} + \widetilde{B}_{j-1}\dot{\tilde{u}}_{j-1}d_{j-1}, \tag{3.23}$$

$$\begin{bmatrix} v(u) \\ \lfloor \omega(u) \rfloor_\vee \end{bmatrix}^{(1)} = 0 \tag{3.24}$$

The second derivative formula is also recursively defined by the relations:

$$\begin{bmatrix} s(u) \\ \lfloor \dot{\omega}(u) \rfloor_\vee \end{bmatrix}^{(j)} = \widetilde{B}_{j-1}\dot{\tilde{u}}_{j-1}[\left\lfloor \omega^{(j)} \right\rfloor_\wedge, D_{j-1}]_\vee + Adj_{A^{j-1}}\dot{\omega}^{(j-1)} + \widetilde{B}_{j-1}\ddot{\tilde{u}}_{j-1}d_{j-1}, \tag{3.25}$$

$$\begin{bmatrix} s(u) \\ \lfloor \dot{\omega}(u) \rfloor_\vee \end{bmatrix}^{(1)} = 0 \tag{3.26}$$

with

$$\ddot{T}(u) = T(u)\left(\left\lfloor \omega^{(k)} \right\rfloor_\wedge^2 - \left\lfloor \dot{\omega}^{(k)} \right\rfloor_\wedge\right),\ \ddot{a}(u) = \ddot{T}(1:3,4)$$

using the adjoint transformation matrix definition:

$$Adj_A = \begin{bmatrix} R & \lfloor t \rfloor_\wedge R \\ 0_{3\times 3} & R \end{bmatrix} \in \mathrm{R}^{6\times 6}, with \quad A = \begin{bmatrix} R & t \\ 0_{3\times 3} & 1 \end{bmatrix} \in \mathrm{SE}(3)$$

$$\left[\left\lfloor \omega^{(j)} \right\rfloor_\wedge, D_{j-1}\right] = \left\lfloor \omega^{(j)} \right\rfloor_\wedge D_{j-1} - D_{j-1}\left\lfloor \omega(u) \right\rfloor_\wedge^{(j)}$$

3.3.1.6 Application: IMU Online Calibration

All the estimations using the ground truth readings (Vicon system) are transformed with respect to the IMU frame of reference. We used the Monocular VIO dataset provided with the EuRoC benchmark [26] (V101 Easy - V102 Medium - V103 Difficult), and we compared the performance of our IMU calibration estimations based-on:

1. EuRoC IMU and ground truth (Vicon)

2. EuRoC IMU and a non-linear least squares ground truth estimator (Optimizer).

In our experiments (see Appendix B), the precision of every B-spline segment (mesh-grid) is 20 points for the pose (p,q), velocity, and acceleration estimation in order to have more readings than that of the IMU to calculate its biases. Also, we performed the calibrations on a 4th-order cumulative B-spline in $\mathrm{SE}(3)$. i.e. using $u = linspace(0, 1, 20)$.

3.3.2 Flow-based Visual Odometry

Corners and their corresponding features from the scene are first extracted via [100] with a block size of 17 pixels. To enhance the robustness and the versatility of the VO process, we adopt the optical flow-based feature tracking

method: Kanade–Lucas–Tomasi (KLT) [56], to match corresponding features in a pyramidal resolution approach of 7 levels with a 17×17 pixels window size.

On tracking the most robust and stable features in 10 consecutive frames, we calculate the *Essential Matrix* with feature outlier rejection by MAGSAC++ [67]. While both RANSAC and MAGSAC++ are useful for estimating model parameters from noisy data, MAGSAC++ offers improved accuracy, robustness, and computational efficiency. Then the relative transformation between every two consecutive frames $\mathcal{T}_{vc} \in \mathrm{SE}(3)$ is recovered from the *Essential Matrix*, which we use to initialize our level 1 optimization process with the initial pose graph using the following arbitrarily scaled transformation:

$$\mathcal{T}_{wc} \doteq \mathcal{T}_{wv} \ \mathcal{T}_{vc}, \tag{3.27}$$

where $\mathcal{T}_{wv} \in \mathrm{SE}(3)$ is the rigid-body transformation between the IMU/body (world) and RGB camera (visual) inertial frames of reference w, v, respectively. In initialization, we assume that there is no translation between the IMU-camera reference frames, i.e., $t_{wv} = [0, 0, 0]^\top$, and the rotation R_{wv} between them is given in Figure 3.3 (b), knowing that the camera frame c and its inertial frame of reference (visual frame v) initially coincides on each other. Until this step, the RGB camera's rigid-body motion $\mathcal{T}_{wc}$ is considered the arbitrary scaled rigid-body motion of all the multi-modal sensor setup $\mathcal{T}^0_{wi}$.

In parallel, a checkerboard corner detection is run on all RGB camera frames. When a checkerboard is detected, an RGB frame is considered a calibration keyframe (KF). We integrate the corresponding time-synchronized, and spatially aligned [101] depth frame (d) to construct a 3D point cloud of the currently detected corners.

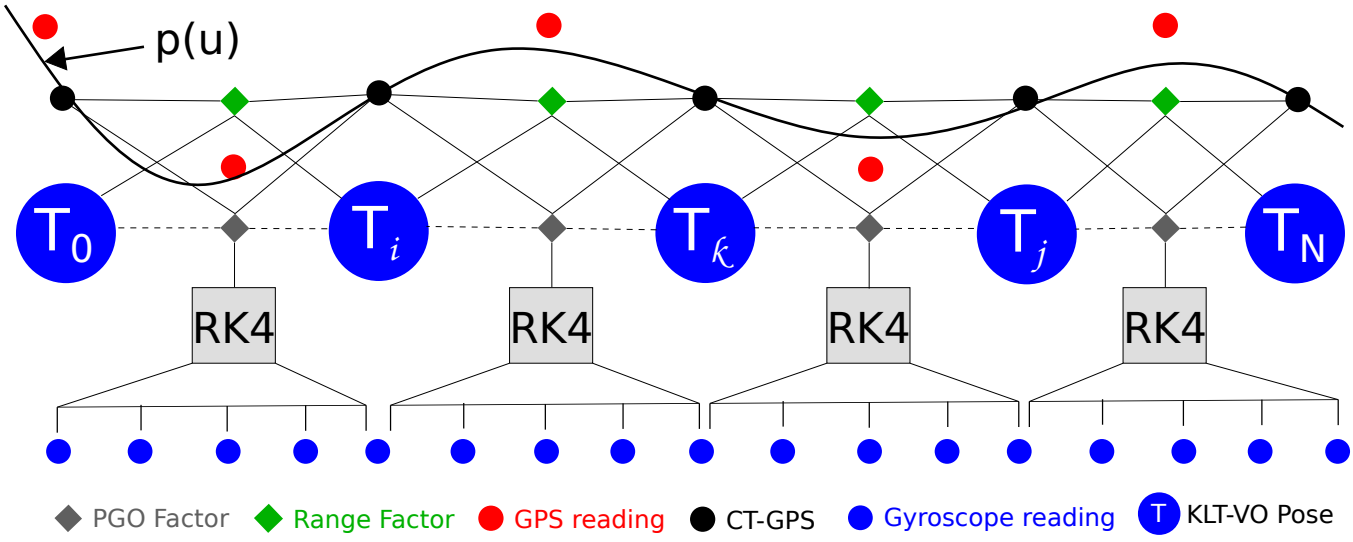


Figure 3.6: Level 1 initialization factor graph. $p(u)$ is the CT-GPS trajectory generated at high frequency. RK4 is the Runge-Kutta 4^{th} order gyroscope integration scheme. Dotted lines denote the error term $(\hat{\boldsymbol{T}}_i^{-1}\hat{\boldsymbol{T}}_j)$ in Equation (3.31) between any two KLT-VO poses.

3.3.3 Optimizer Robust Initialization

After estimating the target's poses and initially constructing point clouds of the checkerboard, bootstrapping the optimizer is essential for a reliable calibration process. This method is efficient in terms of complexity since the bootstrapping relies only on low-rate noisy GPS measurements and gyroscope preintegrated readings. To tackle these GPS problems, we apply an on-manifold cumulative B-spline interpolation [93] to synthesize a very smooth continuous-time (CT) trajectory $\in \mathrm{R}^3$ from the low-rate noisy GPS readings, as illustrated in Figure 3.6.

The matrix form for the cumulative B-spline manifold of order $k = n+1$, where n is the spline degree, is modeled at $t \in [t_i, t_{i+k-1}]$ as:

$$p(u) = p_i + \sum_{j=1}^{k-1} \widetilde{B}_j^{(k)} \bar{u}_j^{(k)} d_j^i \in \mathrm{R}^3, \tag{3.28}$$

where $p(u) \in \mathrm{R}^3$ is the continuous-time B-spline increment that interpolates k GPS measurements on the normalized unit of time $u(t) := (t - t_i)/\Delta t_s - P_n$ with $1/\Delta t_s$ denoting the spline generation frequency and P_n being the pose number that contributes to the current spline segment $P_n \in [0, \cdots, k-1]$. p_i is the initial discrete-time (DT) GPS location measurement at time t_i. The term $d_j^i = p_{i+j} - p_{i+j-1}$ is the difference vector between two consecutive DT-GPS readings. The matrix $\widetilde{B}_j^{(k)}$ is the cumulative basis blending and $\bar{u}_j^{(k)}$ is the normalized time vector, and are defined as:

$$\begin{aligned}
&\widetilde{B}_j^{(k)} = \widetilde{b}_{j,n}^{(k)} = \textstyle\sum_{s=j}^{k-1} b_{s,n}^{(k)}, \\
&b_{s,n}^{(k)} = \tfrac{C_{k-1}^n}{(k-1)!} \textstyle\sum_{l=s}^{k-1} (-1)^{l-s} C_k^{l-s} (k-1-l)^{k-1-n}, \\
&\bar{u}_j^{(k)} = [u^0, \cdots, u^{k-1}, u^k]^\top, \; u \in [0, \cdots, 1].
\end{aligned} \tag{3.29}$$

Our GPS-IMU aided initialization system comprises two optimization factors; the first is a Pose Graph Optimization (PGO) factor r^p that optimizes the 6-DoF of every pose, whereas the second is a Range factor r^s that constraints the translation limits between every two KLT-VO poses. Hence, the metric scale of the visual odometry pose is recovered using the gyroscope and GPS readings, leveraging the high accuracy of the optimization process. An illustrative scheme for the initialization process factor graph is shown in Figure 3.6.

The initialization process objective function $L^{p,s}$ is modeled as:

$$L^{p,s} = \arg\min_{\mathcal{T}_{wi}} \Big[\sum_{(i,j)}^{N} \Big(||r^p(i,j)||^2_{\Sigma^p_{i,j}} + ||r^s(i,j)||^2_{\Sigma^s_{i,j}} \Big) \Big]. \tag{3.30}$$

$\Sigma^p_{i,j}, \Sigma^s_{i,j}$ are the information matrices associated with the GPS readings covariance, reflecting the PGO and Range factors noises on the global metric scale estimation process between two RGB-D aligned frames.

3.3.3.1 Pose Graph Optimization (PGO) factor

The PGO is a 6-DoF factor that controls the relative pose error between two consecutive edges i, j and is formulated as:

$$r^p = \left|\left|\left(\hat{\boldsymbol{T}}_i^{-1}\hat{\boldsymbol{T}}_j\right) \ominus \boldsymbol{\Delta T}_{ij}^{\omega,GPS}\right|\right|_2, \tag{3.31}$$

where $||.||_2$ is the L2-norm, $\hat{\boldsymbol{T}}_{i,j} \in \mathrm{SE}(3)$ is the $\mathcal{T}_{wi}^0$ estimated from the front-end pipeline at frames i, j. The operator $\ominus$ is the $\mathrm{SE}(3)$ logarithmic map as defined in [102]. The error transformation $\Delta T_{ij}^{\omega,GPS}[\delta R_{ij}^{\omega}, \delta p_{ij}^{GPS}] \in \mathfrak{se}(3)$, where $\delta p_{ij}^{GPS} = p_j - p_i$ is the CT-GPS measurement increment and $\delta R_{ij}^{\omega} = [\delta\phi, \delta\theta, \delta\psi]^\top \in \mathfrak{so}(3)$ is the gyroscope integrated increment $\delta R_{ij}^{\omega} = \int_{k=i}^{j} (\omega_k) dk$ using Runge-Kutta 4^{th} order (RK4) integration method [103] between the keyframes i and j.

3.3.3.2 Velocity Graph Optimization (VGO) factor

Velocity Graph Optimization (VGO) is not a standard term in robotics or SLAM. Still, we can proceed with the following model to create an analogous concept to Pose Graph Optimization (PGO) by considering the velocities instead of the poses. Given some constraints or relative velocity measurements, the idea would be to optimize the velocities to minimize the error in the accumulated velocities over time.

We define the Velocity Graph Optimization problem by differentiating the PGO equation with respect to time. We'll assume that the robot poses $\hat{\boldsymbol{T}}_i$ and $\hat{\boldsymbol{T}}_j$ belong to a manifold that is function of time (t), and we have relative velocity measurements v_{ij} instead of $\hat{\boldsymbol{T}}_{i,j} \in \mathrm{SE}(3)$. The residual error function formulating the VGO problem can be defined as:

$$r^v = \left|\left|\left(\hat{\boldsymbol{\zeta}}_i - \hat{\boldsymbol{\zeta}}_j\right) - \boldsymbol{\Delta\zeta}_{ij}\right|\right|_2, \tag{3.32}$$

where $||.||_2$ is the L2-norm, $\hat{\boldsymbol{\zeta}} = [\omega, v]^\top$ is the estimated twist state vector, and a twist vector is a mathematical representation of the combination of linear and angular velocities of a rigid body in motion. It is a compact way to describe the instantaneous motion of an object in 3D space. The twist vector is a 6-dimensional vector comprising two 3-dimensional components: linear and angular. $\boldsymbol{\Delta\zeta}_{ij}$ is the relative velocity vector between the two consecutive poses.

The relative velocity between two poses can be measured using various sensors, including standard cameras, IMUs, event cameras, LiDARs, or RADARs. Each sensor type has its strengths and weaknesses and may require different processing methods to obtain the relative velocity measurement $\boldsymbol{\Delta\zeta}_{ij}$. An overview of how to measure $\boldsymbol{\Delta\zeta}_{ij}$ with each sensor type:

- **Standard Camera (Visual Odometry and Optical Flow)**

Visual odometry algorithms estimate the motion of a camera between consecutive frames by tracking and matching feature points. By processing these tracked features, one can compute the relative pose change and, thus, the relative velocity between frames.

Optical flow is a technique to estimate the apparent motion of objects in consecutive frames of a video sequence. It computes the 2D motion vectors for each pixel, which can be used to approximate the 3D motion between frames. By using techniques like RANSAC and epipolar geometry, one can recover the relative pose and velocity between frames.

- **Inertial Measurement Unit (IMU)**

  An IMU measures the linear accelerations and angular velocities of a device. IMU measurements can be integrated to estimate the relative velocity between two poses. However, integrating the IMU data is prone to drift, and it's usually fused with other sensors like cameras or LiDARs to improve accuracy.

- **Event Camera**

  An event camera is a type of camera that measures changes in pixel intensity asynchronously, capturing events when they happen rather than at fixed intervals. These events can be used to estimate optical flow, which can then be used to compute the relative pose and velocity between poses similar to a standard camera. In our work [104], we apply this VGO factor in an event-based visual-inertial odometry method where the optical flow of events is coupled with the gyroscope's angular velocity readings to measure the $\boldsymbol{\Delta\zeta}_{ij}$ term in a novel and highly efficient approach.

- **LiDAR**

  LiDAR sensors emit laser pulses and measure the time the light bounces back after hitting an object. By processing the point clouds generated by LiDAR, you can estimate the relative pose and velocity between sensor readings. This can be done using algorithms like Iterative Closest Point (ICP) or Generalized Iterative Closest Point (GICP) to align consecutive point clouds.

- **Radar**

  Radar systems emit radio waves and measure the time it takes for the waves to bounce back after hitting an object. Some radar systems can directly measure the velocity of objects using the Doppler effect. One can estimate the relative velocity between poses by processing the radar data and fusing it with other sensor data like cameras or IMUs.

In most practical applications, a combination of multiple sensors is used to obtain more accurate and robust velocity measurements. Sensor fusion techniques like Kalman filters, particle filters, or optimization-based approaches

can be employed to combine the information from different sensors and obtain an optimal estimate of the relative velocity $\boldsymbol{\Delta\zeta}_{ij}$.

This formulation assumes that the relationship between velocities is linear. In practice, this assumption might not always hold, and the residual error function may need to incorporate more complex models, considering the specific motion dynamics of the system and the coordinate transformations between poses.

Note that this formulation is not standard in robotics and SLAM literature. It's our conceptual extension of PGO to velocities. The optimization problem can still be solved using iterative optimization algorithms such as Gauss-Newton or Levenberg-Marquardt.

3.3.3.3 Range constraining factor

The range factor limits the front-end visual drift and keeps the global metric scale under control within a sensible range defined by the GPS signal and is formulated as:

$$r^s = \Big|\Big| ||\hat{\boldsymbol{t}}_j - \hat{\boldsymbol{t}}_i||_2 - ||p_j^{GPS} - p_i^{GPS}||_2 \Big|\Big|_2, \tag{3.33}$$

where inner $||.||_2$ is the Euclidean norm between the translation vectors $\hat{\boldsymbol{t}}_{i,j}, p_{i,j}^{GPS} \in \mathrm{R}^3$ of two consecutive front-end (KLT-VO) poses and CT-GPS signals, respectively.

3.3.4 RGB-D-IMU Local Bundle Adjustment

To estimate the calibration parameters of the RGB-D-IMU, we fuse the tracked checkerboard corners and point clouds with the IMU preintegrated measurements factor proposed in [4]. Figure 3.7 shows our sliding window approach. The local BA is performed on all collected 2D corners $\mathcal{B}$ within their corresponding 3D point cloud $\mathcal{C}$ between two aligned RGB camera c and Depth camera d keyframes i, j, and the IMU readings $\mathcal{I}$ in-between. Our local bundle-adjustment minimization objective function $L^{c,d,\mathcal{I}}$ is defined by:

$$L^{c,d,\mathcal{I}} = \arg\min_{\mathcal{X}} \Big[\sum_{(i,j)}^{N} \rho_{\mathcal{H}}(||r^{\mathcal{I}}(i,j)||^2_{\Sigma^{\mathcal{I}}_{i,j}}) + \sum_{\mathcal{C}_i}^{N} \sum_{\mathcal{B}_i}^{M} \Big(\rho_{\mathcal{H}}(||r^c(\mathcal{B}_i|\mathcal{C}_i)||^2_{\Sigma^c_i}) + \rho_{\mathcal{C}}(||r^d(\mathcal{B}_i|\mathcal{C}_i)||^2_{\Sigma^d_i}) \Big) \Big], \tag{3.34}$$

with $\mathcal{X}$, the full local BA optimization states, which is defined as:

$$\begin{aligned}
&\mathcal{X} = \{\mathcal{K}_c, \mathcal{K}_d, \mathcal{K}_i, \mathcal{T}_{ic}, \mathcal{T}_{dc}, \mathcal{T}_{wi}, v_{wi}, \mathcal{C}^w\}, \\
&\mathcal{K}_c, \mathcal{K}_d = [f_x,\, f_y,\, c_x,\, c_y,\, k_1,\, k_2,\, p_1,\, p_2,\, k_3,\, \lambda] \in \mathrm{R}^{10}, \\
&\mathcal{K}_i{}^k = [\tau_{ic},\, b^{\omega},\, b^{a}] \in \mathrm{R}^7,\ \forall k \in [0, N], \\
&\mathcal{T}_{ic}, \mathcal{T}_{dc}, \mathcal{T}_{wi} = [R_{ic} \,|\, t_{ic},\ R_{dc} \,|\, t_{dc},\ R_{wi} \,|\, t_{wi}] \in \mathrm{SE}(3), \\
&\mathcal{C}^w_k = [X^w,\, Y^w,\, Z^w] \in \mathrm{R}^3,\ \forall k \in [0, N],
\end{aligned} \tag{3.35}$$

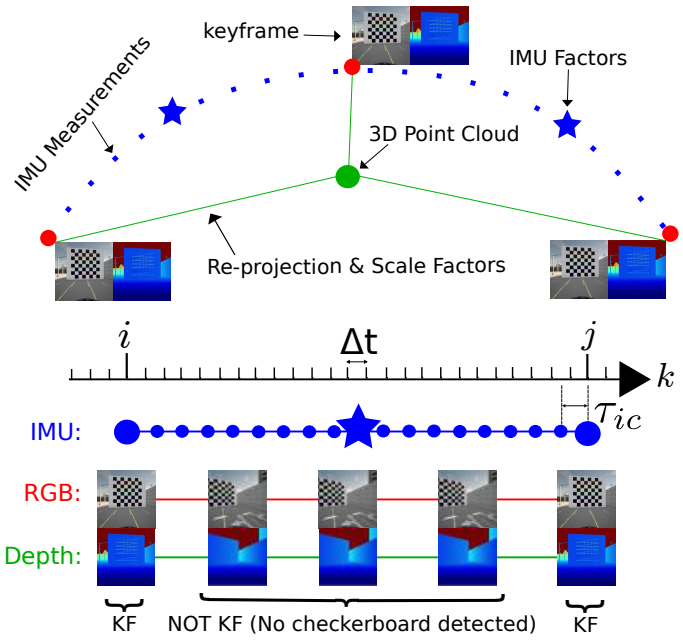


Figure 3.7: Level 2 factor graph between RGB-D aligned keyframes (KF). This factor graph illustrates the non-linear BA process to calibrate the full RGB-D-IMU sensor setup. Δt denotes the IMU time step. τ_{ic} denotes the camera-IMU time offset.

where $\mathcal{K}_c, \mathcal{K}_d$ are intrinsic parameters containing the cameras focal lengths $f_x,\ f_y$, focal centers $c_x,\ c_y$, radial-tangential distortion coefficients $k_{1,2,3},\ p_{1,2}$, and the cloud scale factor λ. $\mathcal{T}_{ic}, \mathcal{T}_{dc}$ are the inter-sensor extrinsic rigid-body transformations. While the spatio-temporal parameters include the scene structure $\mathcal{C}^w$, the body metric scaled pose $\mathcal{T}_{wi}$, velocity v_{wi} with respect to the world coordinates, $\tau_{ic}\ [sec]$ is the IMU-camera time-offset [105], and $b^\omega \in \mathrm{R}^3,\ b^a \in \mathrm{R}^3$ are the gyroscope and accelerometer biases, respectively. N, M are the number of calibration keyframes and corner observations, respectively. $r^{\mathcal{I}}, r^c, r^d$ are the IMU, corner re-projection, and cloud-scale factors, respectively. $\Sigma^{\mathcal{I}}_{i,j}, \Sigma^c_i, \Sigma^d_i$ are the information matrices associated with the IMU readings $\mathcal{I}$, detected corners $\mathcal{B}$, and reconstructed cloud $\mathcal{C}$ scale noise covariance. ρ is the loss function defined by Huber norm [106] $\rho_{\mathcal{H}}$ for $r^{\mathcal{I}}, r^c$ and Cauchy norm [107] $\rho_{\mathcal{C}}$ for r^d.

3.3.4.1 Structured Re-projection Errors factor

We apply the RGB camera pinhole model with radial-tangential distortion coefficients with intrinsic parameters matrix $\mathcal{K}_c$. As illustrated in Figure 3.3 (a), we consider a constructed 3D point cloud $\mathcal{C}^w_k$ using the depth camera aligned k^{th} frame with the current RGB keyframe k. For every checkerboard, we have $H \times W$ feature observations, representing the keyframe's detected corners $\mathcal{B}^c_k[u, v]$.

There is a factor for every detected corner on the current keyframe k that minimizes the error between this corner's location $\mathcal{B}^c_k[u, v]$ and the re-projection of the cloud's $\mathcal{C}^w_k(u, v)$ corresponding 3D point on k^{th} keyframe after distortion $\hat{\mathcal{B}}^c_k[\hat{u}, \hat{v}]$. This factor is defined by:

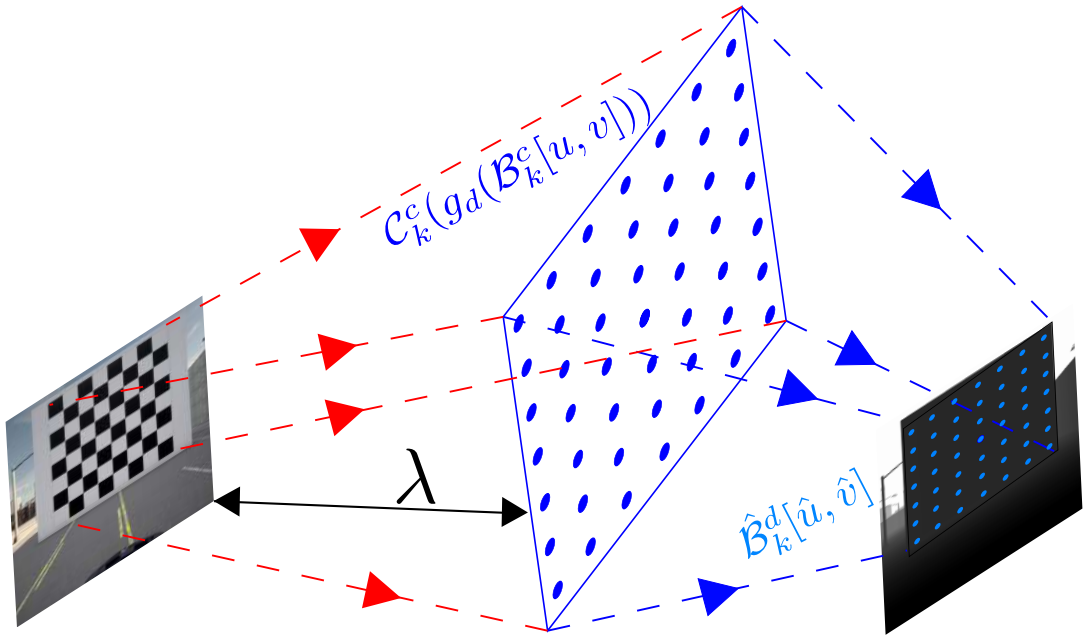


Figure 3.8: Illustration for the 2D-3D-2D projection of the $H \times W = 7 \times 7$ checkerboard feature points from the RGB frame to the point cloud and then to the depth frame. λ is the correction factor for RGB camera intrinsics to estimate the cloud scale factor optimally.

$$r^c = ||\mathcal{B}_k^c[u,v] - \hat{\mathcal{B}}_k^c[\hat{u},\hat{v}]||_2. \tag{3.36}$$

Applying the pinhole camera radial-tangential distortion model [108] to calculate the distorted pixel location of the re-projected 3D point on the current frame $\hat{\mathcal{B}}_k^c[\hat{u},\hat{v}]$, we get:

$$\begin{aligned}
\mathcal{C}_k^c(u,v) &= {\mathcal{T}_{ic}}^{-1}{\mathcal{T}_{wi}}^{-1}\mathcal{C}_k^w(u,v) = [X_k^c, Y_k^c, Z_k^c], \\
\bar{u} &= X_k^c/Z_k^c + c_x/f_x,\ \bar{v} = Y_k^c/Z_k^c + c_y/f_y, \\
r^2 &= \bar{u}^2 + \bar{v}^2, \\
\hat{u} &= f_x(\bar{u}(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 \\
&\quad +2p_1\bar{v}) + p_2(r^2 + 2\bar{u}^2)), \\
\hat{v} &= f_y(\bar{v}(1 + k_1 r^2 + k_2 r^4 + k_3 r^6 \\
&\quad +2.p_2\bar{u}) + p_1(r^2 + 2\bar{v}^2)).
\end{aligned} \tag{3.37}$$

3.3.4.2 Cloud Scale Optimization factor

This factor is modeled to fuse the corner features from RGB frames with the untextured depth maps to benefit from the advantages of both sensors by minimizing the error between the distorted re-projection of the 3D cloud point $\mathcal{C}_k^w(u,v)$ on the k^{th} depth frame $\hat{\mathcal{B}}_k^d[\hat{u},\hat{v}]$ and the current corner feature observation $g_d(\mathcal{B}_k^c[u,v])$ with respect to it.

The effectiveness of this factor comes from the hypothesis that undistorting the depth frame will, in return,

undistort the planar coordinates of the point cloud $\mathcal{C}_k^d[X_k^d, Y_k^d]$. In Figure 3.8, we apply the scale of the cloud λ (known as inverse depth) to optimize the RGB camera focal lengths with the cloud's 3^{rd} coordinate $\mathcal{C}_k^d[Z_k^d]$ which is optimized within the joint calibration model, knowing the metric scale of the pose. This factor is defined by:

$$r^d = ||g_d(\mathcal{B}_k^c[u, v]) - \hat{\mathcal{B}}_k^d[\hat{u}, \hat{v}]||_2, \tag{3.38}$$

where $\hat{\mathcal{B}}_k^d[\hat{u}, \hat{v}]$ follows the same model in Equation (3.37) by replacing $\mathcal{C}_k^c(u, v)$ with $\mathcal{C}_k^d(u, v) = \mathcal{T}_{dc}\mathcal{C}_k^c(u, v)$. $g_d(.)$ is a double re-projection with distortion function, that ***firstly*** projects the observation $\mathcal{B}_k^c[u, v]$ to the 3D point cloud $\mathcal{C}_k^c(\mathcal{B}_k^c[u, v])$ as illustrated by red arrow numbered (2) in Figure 3.3 (a) using the rigid-body transformation $\mathcal{T}_{wc} = \mathcal{T}_{wi}\mathcal{T}_{ic}$ from c to w coordinates with the following formula:

$$\mathcal{C}_k^c(\mathcal{B}_k^c[u, v]) = R_{wc}(\lambda\mathcal{K}_c^{-1}\mathcal{B}_k^c[u, v]) + t_{wc}. \tag{3.39}$$

Then ***secondly***, rotates $\mathcal{C}_k^c(\mathcal{B}_k^c[u, v])$ to $\mathcal{C}_k^d(\mathcal{B}_k^c[u, v])$ using $\mathcal{T}_{dc}$, and ***finally***, re-projects and undistorts this double rotated point on the depth frame $\mathcal{C}_k^d(\mathcal{B}_k^c[u, v])$ using the same model in Equation (3.37).

3.3.4.3 IMU Pre-integration factors

The IMU preintegration factors between two consecutive keyframes i, j is defined in [4] by:

$$\begin{aligned}
r^{\mathcal{I}} &= [\Delta R_{i,j}, \Delta v_{i,j}, \Delta p_{i,j}, \Delta b_{i,j}^{\omega,a}] \in \mathrm{R}^{15}, \\
r_{\Delta R_{i,j}}^{\mathcal{I}} &= log((\Delta\tilde{R}_{i,j})^\top R_i^\top R_j), \\
r_{\Delta v_{i,j}}^{\mathcal{I}} &= R_i^\top(v_j - v_i - g\Delta t_{i,j}) - \Delta\tilde{v}_{i,j}, \\
r_{\Delta p_{i,j}}^{\mathcal{I}} &= R_i^\top(t_j - t_i - v_i\Delta t_{i,j} - \tfrac{1}{2}g\Delta t_{i,j}^2) - \Delta\tilde{p}_{i,j}, \\
r_{\Delta b_{i,j}}^{\mathcal{I}} &= ||b_j^\omega - b_i^\omega||_2 + ||b_j^a - b_i^a||_2,
\end{aligned} \tag{3.40}$$

where $\Delta\tilde{R}_{i,j}, \Delta\tilde{v}_{i,j}, \Delta\tilde{p}_{i,j}$ are the preintegrated rotation, velocity and translation increments. All these on-manifold preintegration increments derivations, as well as the covariance $\Sigma_{i,j}^{\mathcal{I}}$ propagation, are given in the supplementary material of [4], and for better readability, we write $R_{i,j}, t_{i,j}, v_{i,j}$ instead of $[R_{wi}, t_{wi}, v_{wi}]$.

3.4 Experiments

We evaluate the performance of our method (see Algorithm 1) on two applications: RGB-D-IMU Calibration and GPS-aided pose estimation. Using the IBISCape [1] benchmark's CARLA-based data acquisition APIs, we collect three simulated calibration sequences with a vast range of sizes. Moreover, algorithm validation on simulated sequences eases the change of settings to various sensor configurations for robust validation of all corner cases and provides a baseline for most system parameters. Furthermore, for real-world assessment, we evaluate our calibration method

Algorithm 1 End-to-End Optimization Scheme

Input: RGB frames (c), RGB-aligned depth maps (d), GPS readings (DT-GPS), IMU readings ($\mathcal{I}$)
Output: $\mathcal{X} = \{\mathcal{K}_c, \mathcal{K}_d, \mathcal{K}_i, \mathcal{T}_{ic}, \mathcal{T}_{dc}, \mathcal{T}_{wi}, v_{wi}, \mathcal{C}^w\}$

$\mathcal{T}_{vc} \Leftarrow$ KLT-VO (c, $\mathcal{K}_c^0$) ▷ Arbitrary scaled
$\mathcal{T}_{wi}^0 \Leftarrow$ rotate ($\mathcal{T}_{wc} * [\mathcal{T}_{ic}^0]^{-1}$) ▷ Eq. (3.27)
$\mathcal{B}_k^c[u,v] \Leftarrow$ collect_corners (c, H, W) ▷ pix-2D
$\mathcal{C}_0^w \Leftarrow$ construct (d, $\mathcal{B}_k^c[u,v]$, $\mathcal{K}_d^0$) ▷ Initial pcl-3D
$p(u) \Leftarrow$ spline_fit (DT-GPS) ▷ Eq. (3.28)
$[\phi, \theta, \psi] \Leftarrow$ RK4 ($\mathcal{I}_{gyro}(\omega)$) ▷ Initial orientations
while *not converged* **do** ▷ Start Level 1
    $\mathcal{T}_{wi} \Leftarrow$ optimize ($\mathcal{T}_{wi}^0$, $p(u)$, $[\phi, \theta, \psi]$) ▷ Eq. (3.30)
end while
while *not converged* **do** ▷ Start Level 2
    $\mathcal{X} \Leftarrow$ optimize ($\mathcal{I}$, $\mathcal{X}_0$ ($\mathcal{T}_{wi}$, $\mathcal{C}_0^w$)) ▷ Eq. (3.34)
end while

on the RGB-D-IMU checkerboard hand-eye calibration sequence from the VCU-RVI benchmark [36]. Finally, we conduct ablation studies on both IBISCape (Vehicle) and EuRoC [26] (MAV) sequences to assess the contribution of each sensor in an RGB-D-IMU-GPS setup to the accuracy of the pose estimation for a reliable long-term navigation.

Factor graph optimization problems in Equations (3.30) and (3.34) are modeled and solved using a sparse direct method by the Ceres solver [109] with the automatic differentiation tool for Jacobian calculations. The sparse Schur linear method is applied to use the Schur complement for a more robust and fast optimization process. Maximum calibration time for the largest sequence S3 is $\approx 50[min]$ on a 16 cores 2.9 GHz processor and a Radeon NV166 RTX graphics card. The front-end pipeline is developed in Python for better visualization, and the back-end cost functions are developed in C++ to decrease the system latency during the optimization process.

A more in-depth quantitative analysis of the optimization process computational cost is given in Table 3.2, where all experiments converged successfully. The prominent conclusion from this complexity analysis is that the level 2 BA optimization process is computationally highly expensive compared to the target's pose estimation optimization process of level 1. However, this level 2's high computational load can still compete with other calibration tools' BA optimization time, such as Kalibr [53].

3.4.1 Application I: RGB-D-IMU Online Calibration

For both VCU-RVI and CARLA sequences, initial values for the cameras' intrinsic matrices are set to $W/2$ for c_x, f_x, $H/2$ for c_y, f_y, and zeros for the radial-tangential distortions. Initial λ is set with 0.1643, which is the pixel density of CARLA cameras. For extrinsic parameters $\mathcal{T}_{ic}^0$ and $\mathcal{T}_{dc}^0$ initialization, we set the translation part with zeros, and the rotation matrix is set as given in Figure 3.3 (b). Since the VCU-RVI handheld sequence can provide sufficient IMU excitation but with no GPS data available, bootstrapping the calibration system is performed by the traditional IMU-based method [5].

We validate our new cloud global optimization factor based on two criteria: the estimated point cloud after

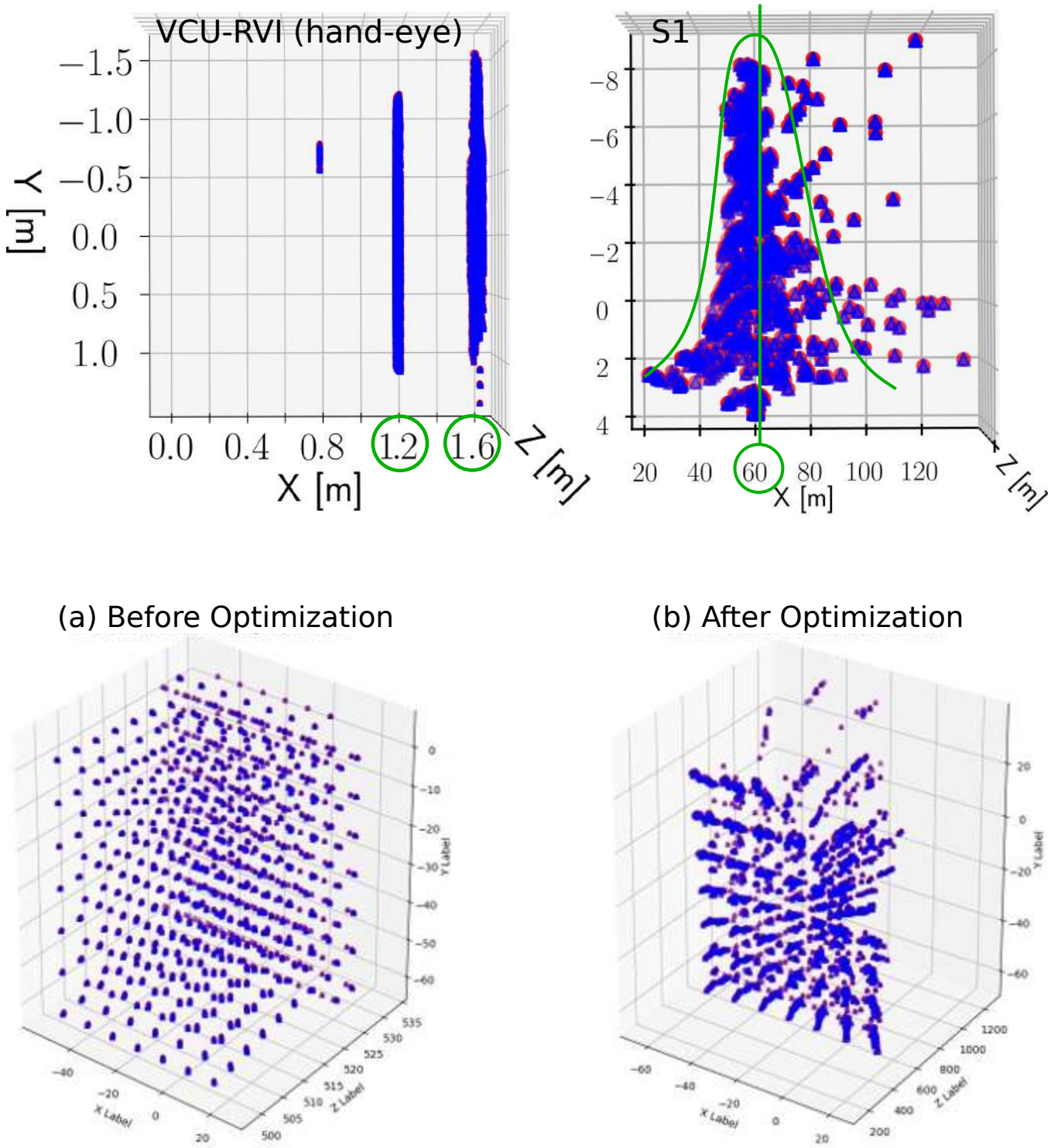


Figure 3.9: **Top**: the target's top-view 3D point cloud reconstruction; (left) VCU-RVI initially constructed point clouds, (right) CARLA optimized point cloud. Blue dots with a red outline denote the checkerboard corners' 3D location. The green colored curve represents the point cloud's normal distribution convergence after optimization. Green circles denote a point cloud depth mean value. **Bottom**: the calibration target's front-view 3D point clouds in the qualitative analysis of the level 2 optimizer performance (a) before and (b) after optimization. Snippets from the S1 calibration sequence of the IBISCape benchmark.

Table 3.2: Optimization process complexity analysis on IBISCape benchmark S1,S2,S3 sequences.

Level		Initial Cost	Final Cost	Residuals	Iterations	Time
1.PGO	S1	1.69e+4	3.19e-9	2464	22	2.81"
	S2	9.62e+4	3.12e-8	6951	26	8.79"
	S3	1.47e+5	2.31e-8	14875	22	16.08"
Average				8097	23	9.23"
2.BA	S1	6.02e+7	1.57e+5	74820	274	2’40.56"
	S2	3.67e+8	7.09e+5	210500	758	22’10.23"
	S3	7.53e+8	1.22e+6	450696	779	49’22.04"
Average				245339	604	24’44.28"

optimization and the depth frame distortion estimation as an indicator for depth correction. Figure 3.9 shows that the optimized cloud is converging to a normal distribution whose mean is the exact location in the simulation world at 60 m, which is at the checkerboard's location as marked on Figure 3.10. Table 3.3 shows the considerably high values for depth frame distortion coefficients, indicating our factor's effect on the cloud's planar undistortion.

Using Kalibr [53] as a baseline for the RGB camera intrinsics for both CARLA and VCU-RVI sequences, we evaluate our optimizer estimation quality in Table 3.3. Since the map scale λ is an RGB camera optimization parameter based on the RGB-D geometric linking constraint introduced in Equation (3.39), the estimates of the focal length need scale correction using: $f_{x,y}^{corr} = f_{x,y}^{est} * \lambda$. For the VCU-RVI hand-eye sequence, we notice that the cloud scale factor is approaching the value 1, which indicates that the initial point cloud is constructed with a high-quality depth sensor.

In Table 3.4, we show the optimal performance of our optimizer to estimate the inter-sensor extrinsic parameters compared to the GT values. Compared to the baseline, our optimizer efficiently estimates the inter-sensor rotation and translation in the case of RGB-D sensors. For the IMU-camera extrinsic parameters and in contrast to rotations, the IMU-camera rigid-body translation mainly depends on the initial values set in the optimizer. In order to estimate the optimal values for the translation part, multiple experiments should be executed with zeros as initial conditions with large data sets. Based on the quality of the IMU still calibration values, all the experiments will converge to relative values, as shown in Table 3.4.

3.4.2 Application II: GPS-aided Visual-Inertial Odometry

Two ablation studies are carried out to assess the contribution of the GPS sensor to the accuracy of the pose estimation when the depth information is available or not available. Standard VIO evaluation metrics [57] are used for assessment: Root Mean Square Absolute Trajectory Error (RMS ATE$_p$ [m]) and Relative Pose Error (RPE$_p$ [m]).

Table 3.3: RGB-D-IMU Sensors Setup Intrinsic Parameters Estimation. Since the CARLA simulator does not provide exact intrinsics values, GT for RGB camera intrinsics are obtained with Kalibr [53]. KF: keyframes count. TL: Sequence Trajectory Length. D: Sequence Duration. * denotes a value calculated from the Structure Core (SC) RGB-D camera specifications with depth FOV=70°. ** denotes a value from the Bosch BMI085 IMU technical data sheet.

Parameter		CARLA Simulator (IBISCape [1])				VCU-RVI [36]	
		S1	S2	S3	GT	hand-eye	GT
Specifications	RGB	20 Hz - 1024×1024 px				30 Hz - 640×480 px	
	Depth	20 Hz - 1024×1024 px				30 Hz - 640×480 px	
	IMU	6-axis acc/gyro @200Hz				6-axis acc/gyro @100Hz	
	#KF	353	994	2126	-	1118	-
	TL[m]	122.06	345.42	737.88	-	11.16	-
	D[sec]	17.640	49.730	106.29	-	46.59	-
RGB Camera	$\lambda.f_x$	164.01	122.71	148.42	151.51	375.67	459.36
	$\lambda.f_y$	163.30	122.22	149.39	151.89	398.44	459.76
	c_x	498.89	506.21	507.59	510.01	315.48	332.69
	c_y	514.01	515.49	518.61	510.71	289.64	258.99
	k_1	-5.10e-3	-6.20e-3	-6.15e-3	2.42e-5	-1.62e-2	-2.98e-1
	k_2	-1.95e-3	-1.96e-3	-2.07e-3	2.89e-6	-3.62e-3	9.22e-2
	p_1	-1.25e-3	-1.96e-3	-8.31e-4	1.71e-4	-2.31e-3	-1.19e-4
	p_2	-3.20e-3	-2.27e-3	-3.53e-3	-3.22e-5	-1.09e-2	-7.46e-5
	k_3	-8.16e-4	-8.70e-4	-8.64e-4	0.0	-7.84e-4	-
	λ	0.3581	0.2819	0.3432	-	0.9831	-
Depth Camera	f_x	511.42	511.51	511.51	512.0	456.82	457.01*
	f_y	511.91	511.83	511.82	512.0	456.06	457.01*
	c_x	512.20	512.22	512.30	512.0	333.29	320.0*
	c_y	511.81	512.01	512.02	512.0	259.17	240.0*
	k_1	-3.53e-2	-3.37e-2	-3.54e-2	-	-5.74e-2	-
	k_2	-5.60e-3	-6.20e-3	-6.25e-3	-	-9.07e-3	-
	p_1	-3.41e-2	-3.22e-2	-3.29e-2	-	-4.13e-2	-
	p_2	-3.93e-2	-3.50e-2	-3.82e-2	-	-6.09e-2	-
	k_3	-1.10e-3	-1.45e-3	-1.38e-3	-	-2.98e-4	-
IMU Sensor	τ_{ic}	4.986e-3	4.989e-3	4.998e-3	5e-3	4.473e-3	-
	b_x^ω	-7.549e-3	-2.242e-2	-4.907e-3	-2.383e-3	1.512e-4	9.69e-5**
	b_y^ω	-3.283e-2	3.813e-2	-2.054e-2	-3.364e-3	9.337e-5	9.69e-5**
	b_z^ω	8.151e-2	2.659e-2	-2.540e-2	1.555e-3	-2.967e-4	9.69e-5**
	b_x^a	0.109	-0.062	0.147	-0.951	-5.704e-4	-
	b_y^a	-0.707	-1.069	-0.091	-0.691	6.757e-4	-
	b_z^a	-1.926	-2.295	-2.364	0.183	-9.304e-4	-

Table 3.4: Extrinsic parameters estimation for both IBISCape (S1,S2,S3) and VCU-RVI (hand-eye) calibration sequences.

Parameter		$t_x[m]$	$t_y[m]$	$t_z[m]$	q_x	q_y	q_z	q_w
RGB-D ($\mathcal{T}_{dc}$)	S1	4.95e-3	0.017	0.037	-0.037	-0.022	0.030	0.997
	S2	5.47e-3	0.020	0.065	-0.041	0.005	0.019	0.996
	S3	9.10e-3	0.018	0.065	-0.036	-0.010	0.025	0.997
	GT	0.0	0.020	0.060	0.0	0.0	0.0	1.0
	hand-eye	-0.103	0.003	0.018	0.041	0.081	0.009	0.969
	GT	-0.100	0.0	0.0	0.0	0.0	0.0	1.0
RGB-IMU ($\mathcal{T}_{ic}$)	S1	-0.806	0.154	-0.308	0.493	0.507	0.499	0.500
	S2	-0.854	-0.057	0.006	0.503	0.495	0.501	0.498
	S3	-0.808	-0.028	-0.102	0.503	0.501	0.499	0.496
	GT	-0.800	0.0	0.0	0.500	0.500	0.500	0.500
	hand-eye	0.077	0.020	-0.041	0.699	-0.713	-0.009	-9e-4
	GT	-0.008	0.015	-0.011	0.708	-0.706	0.001	-4e-4

Table 3.5: Ablation study on the contribution of the GPS sensor on the system accuracy when depth information is available.

Method	IBISCape [1] (RPE$_p$ ($\mu \pm \sigma$) [m]) S1	S2	S3	Average
DUI-VIO [88]	0.115±0.113	0.115±0.114	0.120±0.119	0.117±0.115
BASALT [69]	0.084±0.084	0.052±0.051	0.026±0.026	0.054±0.054
ORB-SLAM3 [43]	0.028±0.013	0.073±0.034	0.031±0.028	0.044±0.025
Proposed (Lvl.1+2)	**0.016±0.019**	**0.025±0.030**	**0.018±0.025**	**0.020±0.025**

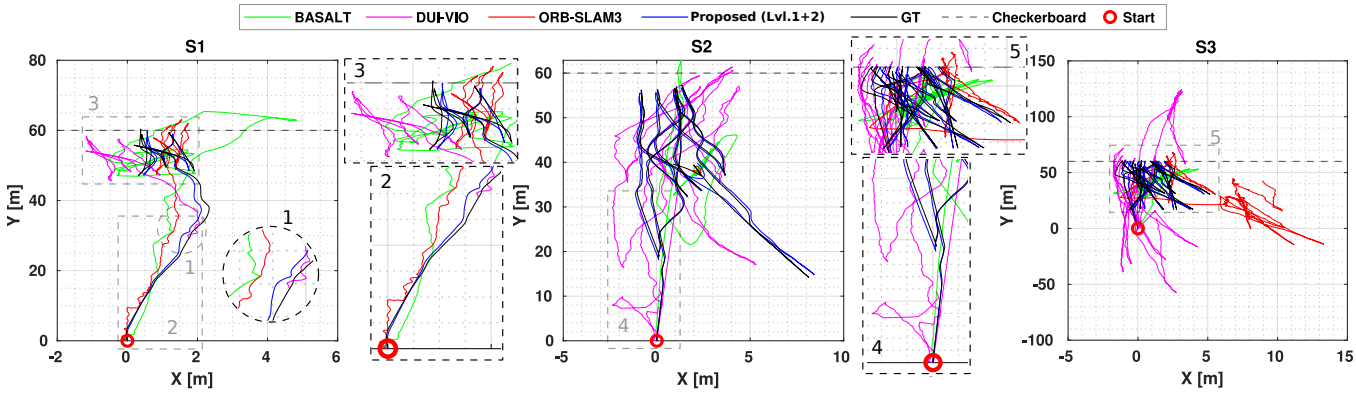
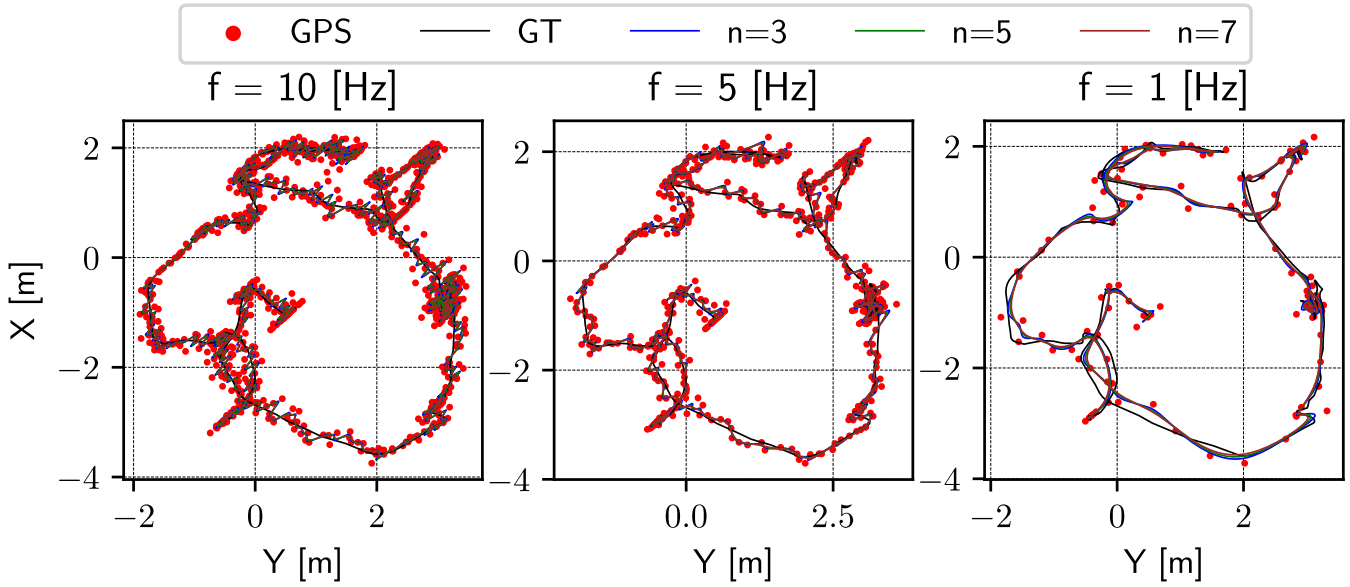


Figure 3.10: Pose estimation evaluation of our method compared to ORB-SLAM3, BASALT, and DUI-VIO on S1,S2,S3 sequences. Different axes scale for showing fine details.



RMSE [m]	f	10 [Hz]			5 [Hz]			1 [Hz]		
	n	3	5	7	3	5	7	3	5	7
		0.12372	0.11907	0.12205	0.11709	0.12786	0.15240	0.23013	0.38962	0.55157

Figure 3.11: Synthesizing low-rate noisy DT-GPS readings with three frequencies [10,5,1] Hz on EuRoC V2-01 sequence and performing the B-spline interpolation (CT-GPS) with manifolds of degree (n=3,5,7). Blue denotes the most accurate, red denotes the least accurate, and green denotes the parameters used in our experiments (n=3, f=10Hz). RMSE is the accuracy evaluation metric.

Table 3.6: Ablation study on the contribution of the GPS sensor on the system accuracy when depth information is unavailable. * denotes tracking features in 5 consecutive frames instead of 10 due to the rapid motion of the MAV. + denotes the only learning-based baseline in the table and the only method incorporating LiDAR point clouds. V,I,G: Vision, IMU, and GPS.

	Method	EuRoC [26] (RMS ATE$_p$ [m])						Avg.
		V1-01	V1-02	V1-03	V2-01	V2-02	V2-03	
Mono-VI	OKVIS [110]	0.090	0.200	0.240	0.130	0.160	0.290	0.185
	ROVIO [42]	0.100	0.100	0.140	0.120	0.140	0.140	0.123
	VINS-Mono [5]	0.047	0.066	0.180	0.056	0.090	0.244	0.114
	OpenVINS [111]	0.056	0.072	0.069	0.098	0.061	0.286	0.107
	CodeVIO$^+$ [89]	0.054	0.071	0.068	0.097	0.061	0.275	0.104
Stereo-VI	VINS-Fusion [112]	0.076	0.069	0.114	0.066	0.091	0.096	0.085
	BASALT [69]	0.040	0.020	0.030	0.030	0.020	0.050	0.032
	Kimera [113]	0.050	0.110	0.120	0.070	0.100	0.190	0.107
	ORB-SLAM3 [43]	0.038	0.014	0.024	0.032	0.014	0.024	0.024
Mono-V/I/G	CT (V+I+G) [114]	0.024	0.014	**0.011**	0.012	0.010	**0.010**	0.014
	CT (V+G) [114]	0.011	**0.013**	0.012	0.009	**0.008**	0.012	**0.011**
	CT (I+G) [114]	0.062	0.102	0.117	0.112	0.164	0.363	0.153
	DT (V+I+G) [114]	0.016	0.024	0.018	0.009	0.018	0.033	0.020
	DT (V+G) [114]	0.010	0.025	0.024	0.010	0.012	0.029	0.018
	DT (I+G) [114]	0.139	0.137	0.138	0.138	0.138	0.139	0.138
Proposed (Lvl.1)		**0.008**	0.017*	0.023*	**0.008**	0.022	0.025*	0.017

3.4.2.1 Ablation Study on a Simulated Ground Vehicle

In the first ablation study, we assess the performance of our depth-incorporated pose estimation with GPS-aided bootstrapping compared to the latest state-of-the-art VIO systems that do not utilize GPS readings in their estimations. We compare our GPS-aided RGB-D-IMU pose estimation accuracy with that of ORB-SLAM3 (RGB-D) [43], BASALT ($2\times$RGB-IMU) [69], and DUI-VIO (RGB-D-IMU) [88] systems using both VCU-RVI and CARLA sequences.

During the evaluation of the DUI-VIO [88] system, we noticed an initialization failure with the S1 sequence till the system initialized successfully at the end of the speed bump at nearly 30 m as magnified in Figure 3.10 (#1). This initialization problem is not witnessed with the VCU-RVI hand-eye calibration sequence due to its complex combined motions (see Figure 3.1 (right)). Sequences (S2, S3) are simulated with a high combined motion to ensure the optimal checkerboard coverage for all the RGB-D camera frames. The complex motion generated sufficient IMU excitation to initialize BASALT and DUI-VIO.

In our analysis in Table 3.5, the quantitative results show superior performance for our method compared to other approaches. Indeed, the pose estimation error is reduced by 54.55%, 62.96%, and 82.91% compared to ORB-SLAM3, BASALT, and DUI-VIO, respectively. This happens thanks to our fast bootstrapping GPS-aided method that decreases the relative pose error accumulation with time.

3.4.2.2 Ablation Study on a Real-world Aerial Vehicle

To further validate the performance of our pose estimation method in a real-world application, we perform another ablation study. The experiments of this study were performed on the EuRoC MAV dataset [26] incorporating RGB-IMU sensors and compared to the continuous-time and discrete-time (CT/DT) GPS-based SLAM system proposed in [114]. Since a comparison with the competing technique [114], combining GPS signals computed from the Vicon system measurements better emphasizes the findings of this ablation research, we chose the identical six Vicon room sequences from the EuRoC benchmark they used in their evaluation.

The GPS readings for EuRoC sequences are generated with the same realistic model and parameters given in [114] that gives a real-world accuracy but does not suffer from limitations as multipath effects [115]. CARLA GPS sensor is modeled as most commercial sensors containing a particular bias with a random noise seed and a zero mean Gaussian noise added to every reading. The most prominent conclusion from Figure 3.11 is that as the GPS rate increases, the CT-GPS interpolation is better with a low degree (n) manifold, and vice-versa, and our GPS-aided initialization method can still be valid with the lowest GPS frequency ($f = 1\,Hz$).

The quantitative analysis in Table 3.6 shows that our level 1 estimations, with no depth information, can efficiently estimate a metric-scaled trajectory that can bootstrap level 2 and outperform other well-established VIO systems in terms of accuracy. We also notice an improvement in estimation accuracy with adding a sensor modality (IMU/GPS), given that at least one visual sensor is present in the system. Another conclusion is that a GPS can be sufficient with

the optical sensor to get a reliable trajectory estimate in a tightly-coupled fusion scheme. For a loosely-coupled fusion scheme (proposed Lvl.1), adding a gyroscope increases the confidence of the optimizer to converge to reasonable values.

3.4.3 Algorithm's In-depth Behavioural Insights

In this sub-section, we enclose a more in-depth quantitative analysis for our calibration and pose estimation results (dashed lines) compared to the ground truth values (solid lines) when available or to a baseline algorithm—for example, DUI-VIO is used as GT with the VCU-RVI handheld (hand-eye) calibration sequence.

CARLA datasets used in this experiments link:

`https://drive.google.com/drive/folders/1aL4JNtUfshEw-nilLsgefijOSecUqwtf?usp=sharing`

3.4.3.1 CARLA and VCU-RVI Quantitative Analysis

Figure 3.12 reports the 2D-XY trajectories with more information, including the trajectory estimated with each level of the optimization process, the KLT-VO (in red) up-to-scale trajectory, and both the DT-GPS and CT-GPS trajectories.

Figure 3.13 illustrates the quality of the optimization process. Starting in the first column with the RK4 gyroscope integration technique showing insights into the level 1 optimization quality. Then, the velocity estimation is one of the important parameters to verify the Bundle-Adjustment optimization quality because it is initialized with zero values. The estimated value completely depends on the IMU preintegration and bias factors, and the IMU still calibration noise standard deviation values.

Finally, the last column reports the translations and rotations Relative Pose Error (RPE) for both the optimization levels (1 in red and 2 in blue). The translations RPE values (top) are reported in [cm] and for the rotations (bottom) in [degrees]. For the RK4 evaluation; rows 1-3: Roll ϕ, Pitch θ, and Yaw ψ angles in [rad]. For the velocity estimations; rows 1-3: V_x, V_y, and V_z in [m/s].

The main conclusion from the quantitative evaluation results on both VCU-RVI and CARLA sequences is that the RK4 integration scheme generates highly stable orientations with smooth transitions. i.e., when gyroscope sensor measurements have an immediate transient impact, the RK4 integration scheme can filter the noisy measurements. One limitation of the RK4 integration scheme is the high bias of the sensor as simulated in CARLA with the roll ϕ angular velocity (this phenomenon is not witnessed with the real-world VCU-RVI sequence with the Bosch BMI085 IMU sensor).

3.4.3.2 EuRoC Quantitative Analysis

Figure 3.14 compares our level 1 optimizer estimated trajectory to the ground truth for all the V1- and V2- sequences starting with the easiest in the top row, then medium in the middle, and the hardest in the bottom row. Then, in

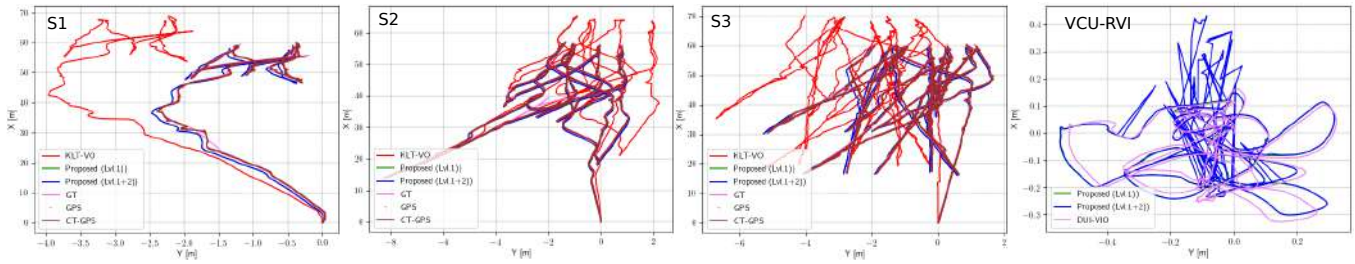


Figure 3.12: More quantitative evaluation on the 2D-XY estimated trajectories for the CARLA and VCU-RVI sequences.

Figure 3.15, we compare the level 1 estimated velocities to the ground truth velocities enclosed with the EuRoC sequences. Moreover, in Figure 3.16, we show the RK4 evaluation compared to the ground truth orientations. Figure 3.16 gives a more in-depth view of the insights of the RK4 integration scheme and its orientations estimation accuracy, especially with the medium (middle) and hard (bottom) sequences. Finally, the RPE evaluation results are reported in Figure 3.17 with the translation errors in [cm] to the left and the rotational errors in [degrees] to the right.

The main conclusion from the quantitative evaluation results on EuRoC sequences, the RK4 integration scheme can produce reasonable orientations estimations in the case of easy and medium sequences (V1-01, V1-02, V2-01, V2-02) due to the low number of rapid transient changes of the motion of the MAV. Whereas, for the hard sequences, the RK4 results are slightly degrading in the integration quality due to the high number of significant and rapid transient changes of the motion of the MAV in brief time lapses. Since the Pose Graph Optimization (PGO) factor accounts for the orientations increments, and the information matrix includes a standard deviation value that incorporates the noise to the orientations increments, this degraded quality with the hard sequences did not affect the overall quality of the level 1 optimization process.

3.5 Conclusion

This chapter proposes the first baseline method for robust RGB-D-IMU intrinsic and extrinsic calibration, addressing a critical challenge in the field of visual-inertial navigation for autonomous systems. Our novel approach begins with an RGB-GPS-Gyro optimizer bootstrapping technique that reliably estimates the metric-scaled target's pose, providing a strong foundation for the calibration process. Subsequently, we introduce a cloud-scale factor for spatially aligning untextured depth maps in RGB-D, which estimates the scale by incorporating the uncertainty of the initially reconstructed cloud.

Experimental results on both real-world and simulated sequences demonstrate the effectiveness of our method, which can be considered as the foundation for a cutting-edge RGB-D GPS-aided VI-SLAM system with a reliable online calibration algorithm. These promising results indicate that our method has the potential to significantly enhance the performance and reliability of visual-inertial navigation systems.

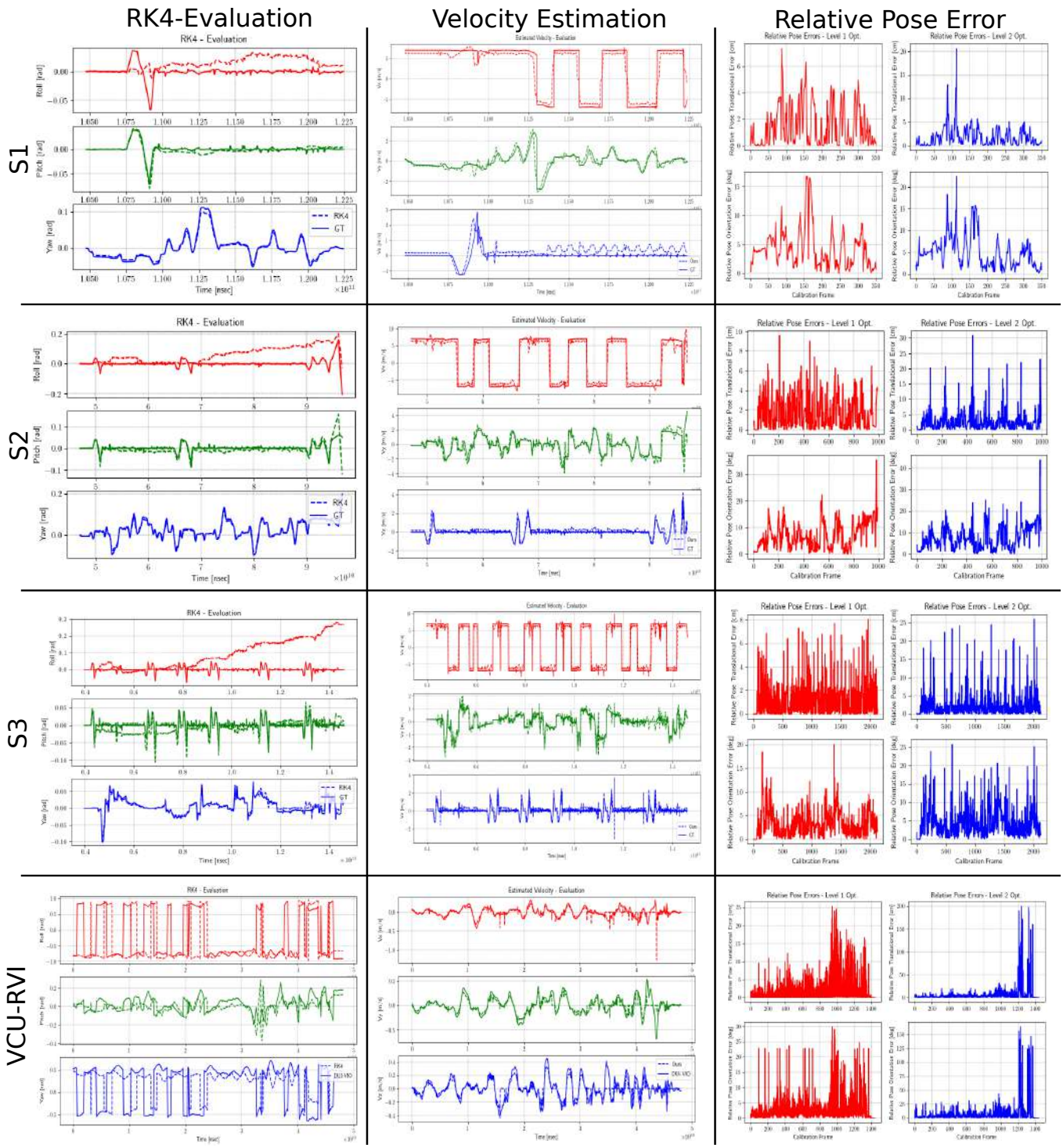


Figure 3.13: RK4 Evaluations, Velocities Estimations and Relative Pose Error Analysis.

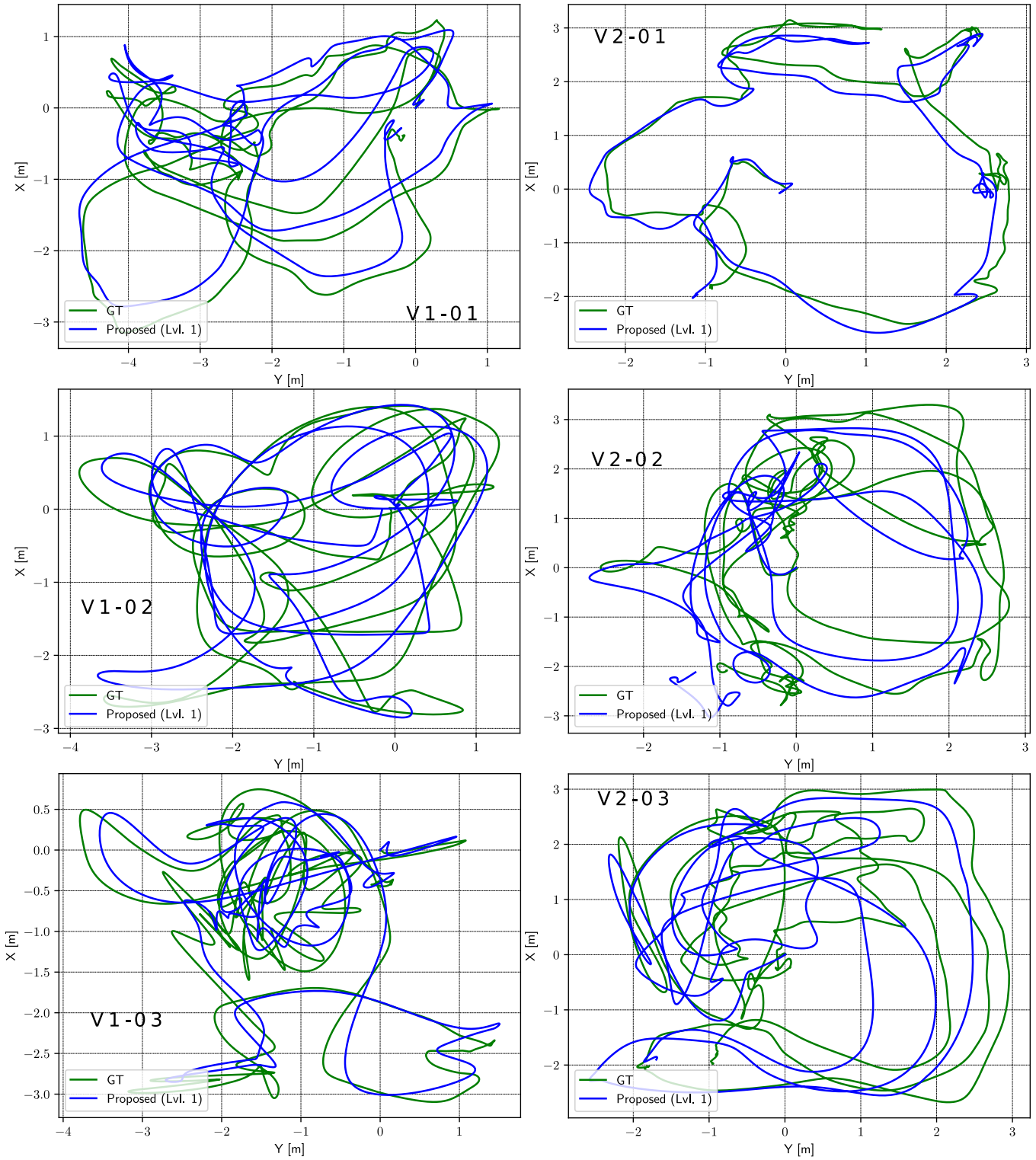


Figure 3.14: 2D-XY estimated trajectories for the EuRoC sequences.

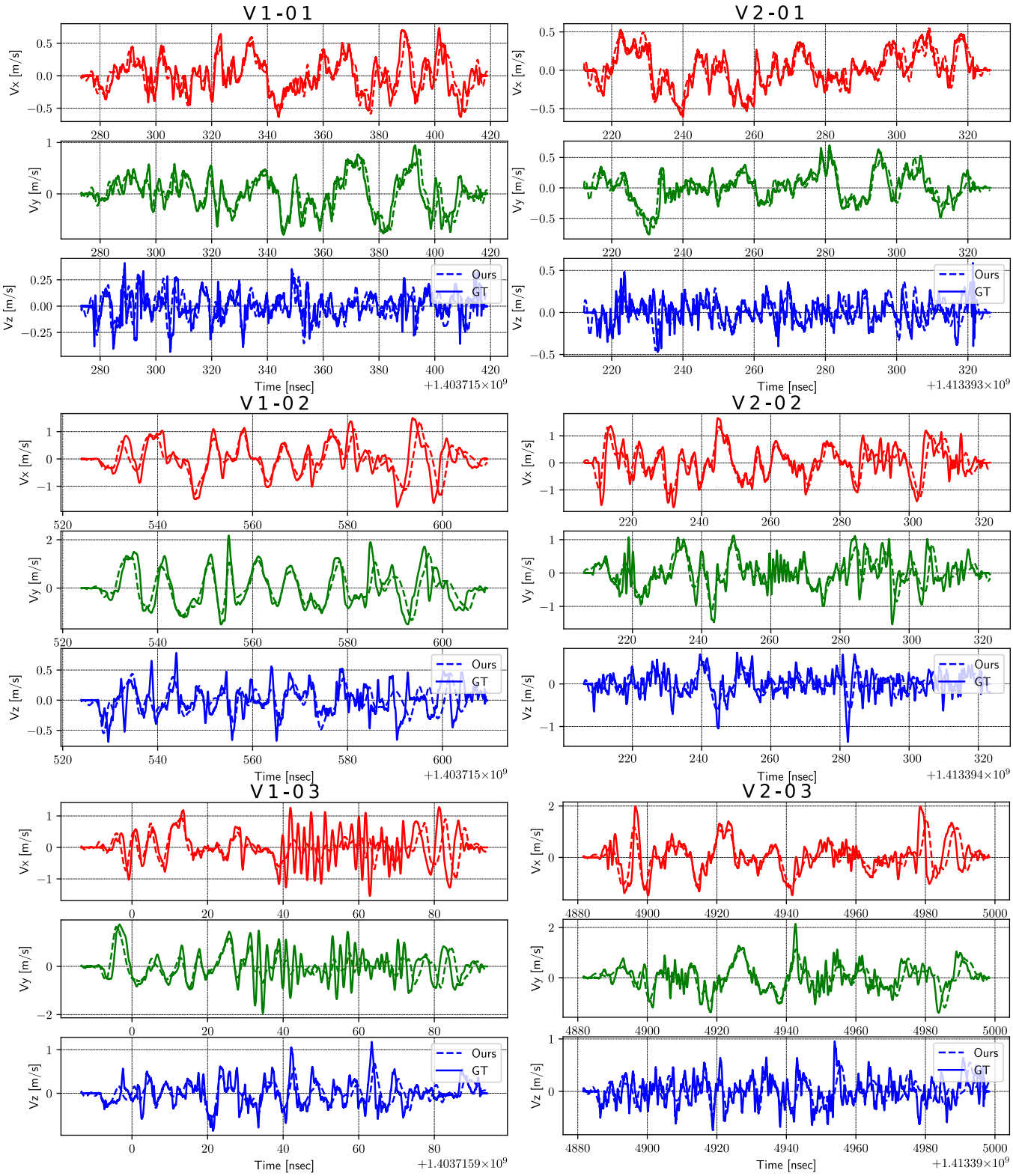


Figure 3.15: Velocities Estimation.

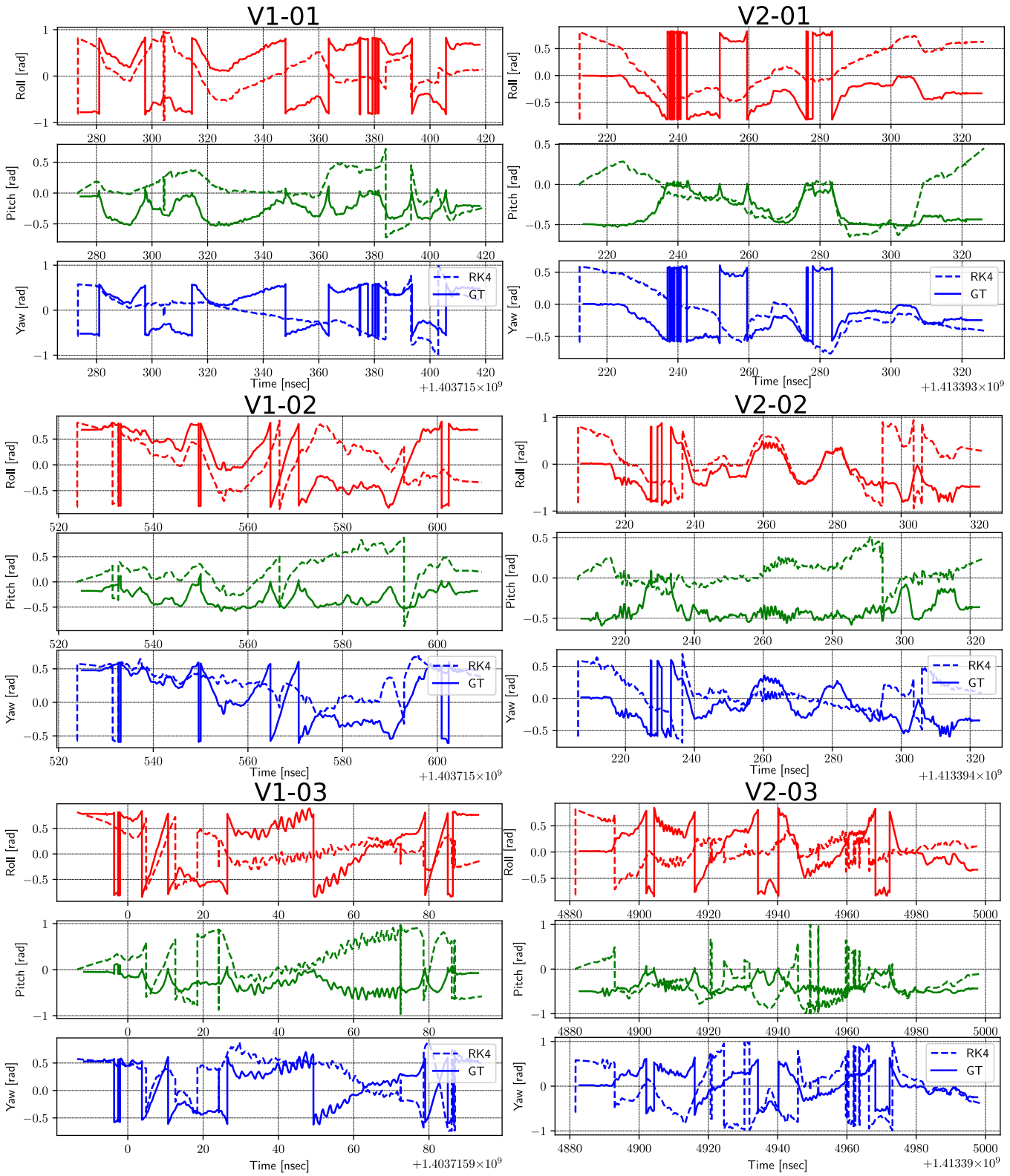


Figure 3.16: RK4 Integration Scheme Evaluation.

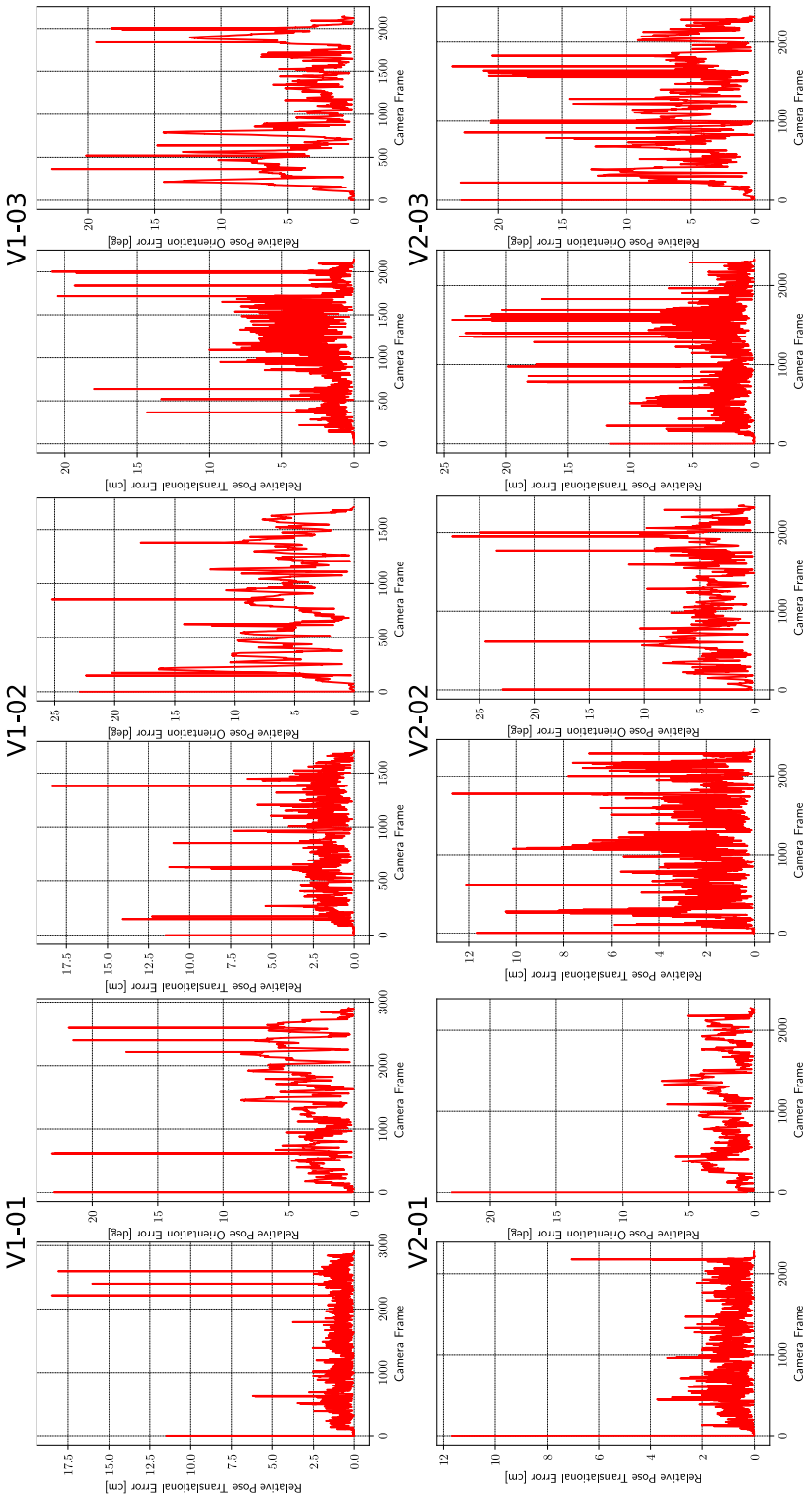


Figure 3.17: Relative Pose Error Analysis.

In future work, it will be crucial to address situations where GPS sensor limitations, such as multipath effects, cannot be simulated in the optimizer. This will further improve the robustness of our method and ensure its applicability in challenging environments. Additionally, it will be essential to generalize the Bundle Adjustment (BA) optimization problem to extend the algorithm's calibration capability to include multiple IMUs and vision sensors (RGB and depth), thus catering to more complex and diverse system configurations.

In conclusion, our proposed method represents a substantial contribution to the field of RGB-D-IMU intrinsic and extrinsic calibration, offering a reliable and effective solution to a critical problem. The promising experimental results obtained in this chapter underscore the potential of our method for real-world applications and set the stage for further advancements in this area. By addressing the calibration challenge, our work has the potential to significantly impact the development of more robust and accurate autonomous navigation systems.

4 Hybrid State Estimation

Abstract

In this chapter, a linear optimal state estimation approach is introduced for Micro Aerial Vehicles (MAVs) in order to achieve highly accurate localization while minimizing system delay. The proposed approach incorporates a decoupled optimization- and filtering-based sensor fusion technique, which aims to achieve both high estimation accuracy and minimal system complexity. The system utilizes real-world indoor and outdoor settings as experimental environments for conducting MAV localization studies. Through these studies, the proposed method's findings are validated and tested, assessing its effectiveness and performance in different scenarios. The chapter provides insights into the capabilities and limitations of the proposed approach, shedding light on its potential applications in MAV localization.

"If I have seen further than others, it is by standing upon the shoulders of giants."

Isaac Newton

4.1 Introduction

Robust localization of Micro Aerial Vehicles (MAVs) in uncharted large-scale areas can rely on complementary data gathered by many sensor modalities. The study of Simultaneous Localization And Mapping (SLAM), primarily used for MAV navigation in expansive and dynamic settings, may be enriched and expanded by using multi-modal datasets [1]. These settings have certain traits, such as the dynamic range of the scene's object intensities. For instance, mapping a small interior space with adequate illumination might be of more outstanding quality than mapping a rural area at night with heavy rain, wind, and fog (outdoors dynamic environment). The benefits of multimodal approaches become apparent when systems rely on sensors with high dynamic range and strong sensing capabilities, such as event cameras, LiDARs, or Radars, or typical inexpensive cameras fused with other sensor modalities such as the Inertial Measurement Units (IMUs) and GPS sensors. These multimodal approaches can fill indeed some lack of data during scene mapping and MAV localization.

Toward this aim, we develop a trustworthy (quick and precise) localization solution that utilizes information from three sensor modalities: camera frame data, IMU measurements, and GPS readings. Nevertheless, the GPS sensor readings are consistently slower and noisier than those from the IMU or camera modules, and they frequently experience signal loss in GPS-restricted locations. Therefore, a localization system that depends on GPS data must perform effectively when GPS readings are lost.

Visual-Inertial Odometry (VIO) is one of the most mature and well-established approaches in the localization field [116, 117, 118]. Efficient visual odometry can be achieved using a high-quality perception of the surroundings. Sensors performing this perception task can differ in their nature of data collection. On the one hand, the most common visual odometry sensors are cameras like RGB cameras [119], Event cameras [120], and RGB-D cameras [121]. On the other hand, using LiDAR sensor [7] can provide point clouds, and GPS sensor [122, 123] can locate the MAV using satellite signals triangulation as represented in Figure 4.1.

The accuracy of the state estimation process relies on an Error-State Extended Kalman Filter (ES-EKF) and the bootstrapping quality of its states. A well-established IMU-based state estimator initialization technique is discussed in [119]. In this bootstrapping method, the global metric scale of the trajectory and the IMU-camera gravity alignment is optimized using a specific amount of IMU readings preintegration combined with an initial up-to-scale trajectory estimated using the camera only. This bootstrapping process is prone to failure due to insufficient IMU excitation, especially when the MAV navigates in a planar terrain.

The MAV should contain a localization system that continually calculates the pose with high accuracy and low latency during search and rescue missions, for instance. The MAV is equipped with restricted resources regarding the data processing unit and the limited power source capacity for long-term navigation operations in large-scale situations. In light of this, the state estimate approach should consistently have low computational complexity and resist sensor readings that deviate from the norm.

Figure 4.1: An example for the on-map GPS readings of the large-scale environment of the **Fast Flight** dataset [124] sequences: gps175, gps15, gps10, and gps5. The sequence number denotes the maximum flight velocity of each sequence: 17.5, 15, 10, and 5 [m/sec], respectively. The color bar (bottom) denotes the map scale in [km] on the x-axis and the altitude of each sequence in [m] on the y-axis. In the blue dotted box: Comparing the maximum MAV's altitude at instance before the descent stage to the height of an aircraft hangar. The estimated airport asset height is 54.72 [m], corresponding to the maximum MAV altitude. Images are courtesy of Google Earth.

Our work's main contribution to tackle the aforementioned challenges is threefold:

- In case of state estimator initialization failure, we propose a unique instant bootstrapping technique based on continuous-time manifold optimization via Pose Graph Optimization (PGO) and Range factors, which depends on low-rate GPS signals.

- A closed-form estimation method without non-linear optimization during IMU/CAM fusion produces a reduced system latency with a constant CPU computing complexity. The mathematical modeling of a linear ES-EKF with a precise and quick gyroscope integration strategy accounts for the simplicity of our proposed localization solution.

- The EuRoC benchmark [26], for MAV localization assessment in indoor environments, and the Fast Flight dataset [124], for large-scale outdoor environments, are two real-world publicly available benchmarks on which our IMU/GPS-CAM fusion system is thoroughly tested. With thorough ablation investigations on the role of each sensor modality in the overall accuracy of the state estimation process, the assessment is conducted using the most recent state-of-the-art visual-inertial odometry methodologies.

4.2 Related Work

4.2.1 Sensor Fusion

Figure 4.2 presents a global overview of the current state-of-the-art approaches for localization. The ability to continually estimate the robot's ego-motion (position and orientation) over time is a significant difficulty in autonomous navigation, path planning, object tracking, and collision avoidance platforms [112]. The Global Positioning System (GPS) is a well-known localization method applied to several autonomous system domains. One kind of Global Nav-

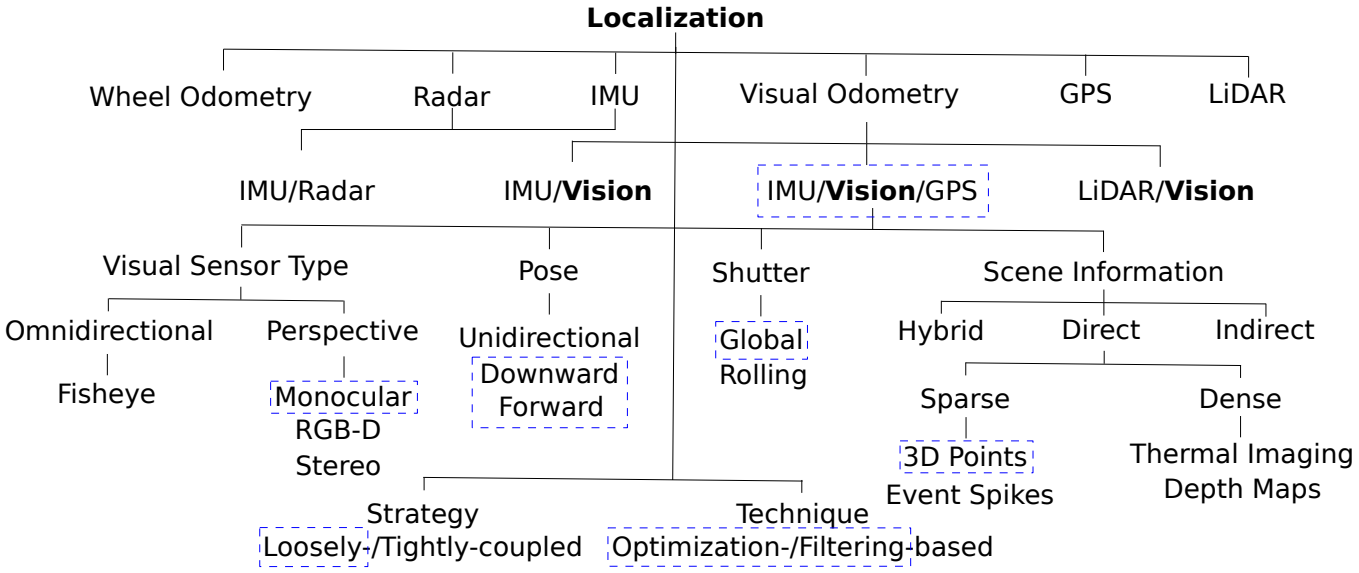


Figure 4.2: Visual odometry is generally categorized together with self-contained and global localization methods.

igation Satellite System (GNSS) is GPS [123]. GPS is used as a self-localization source, such as for MAVs security applications, and gives any user with a GPS receiver positional information with meter-level precision. The satellite signal blockage, high noise levels, multipath effects, and other issues with GPS, on the other hand, make it a less trustworthy alternative sensor for self-localization modules. However, RTK (Real-Time Kinematic) and PPP (Precise Point Positioning) [122], two GPS technologies that are rapidly developing, can provide locations with decimeter- or centimeter-level precision.

The effectiveness of GPS satellite signals depends heavily on the surrounding environment; it works best in locations with clear skies and is ineffective for inside navigation since walls and other obstacles impede it [125]. This makes the GPS module an unsuitable primary sensor for reliable autonomous vehicle localization in adverse weather and environmental conditions. Hence, the fusion of GPS signals with other inertial and/or visual sensors is indispensable for a reliable localization solution, especially in such environments. The state-of-the-art sensor fusion systems are differentiated into two prominent families: loosely- [126], and tightly-coupled [127] fusion strategies. In loosely-coupled fusion, the camera frames for pose estimation are processed as a black-box. A filter or an optimization model is developed to fuse the arbitrary-scaled poses from the visual sensor with the noisy metric-scaled pe-integrated IMU readings [128].

On the contrary, in the tightly-coupled approach, scene information from the visual sensor is fused with the IMU measurements (linear accelerations and angular velocities) using a fusion filter or an optimization model that estimates the metric-scaled pose, visual odometry scale factor, IMU biases, and visual drift between the IMU-camera inertial frames. One of the prominent advantages of a tightly-coupled fusion scheme is that it can estimate accurate scene information to reconstruct a precise scene map, along with providing the SLAM system with high confidence in loop closure during re-localization situations.

4.2.2 Fusion Strategies

The two sensor fusion strategies (loosely and tightly coupled) have two main execution techniques: filter-based and optimization-based. Some filter-based state-of-the-art approaches are deterministic such as MSCKF [41], S-MSCKF [124], S-UKF-LG/S-IEKF [129], and ROVIO [42]. At the same time, alternative strategies can be based on nondeterministic filters like particle filters [130], where a collection of Monte Carlo algorithms is used to address filtering issues in Bayesian statistical inference and signal processing.

Optimization-based methods such as VINS-Mono [5], OKVIS [110], ORB-SLAM [43], and BASALT [69], can be deterministic or nondeterministic based on the optimization strategy and the convergence constraints. The estimation and robustness of visual localization frameworks have advanced significantly over the past several decades, and this development may be furthered by tightly integrating visual and inertial data. Most methods integrate data utilizing optimization methods or filtering-based procedures.

Filtering approaches are ideally suited to real-time applications [131, 132], which is the main emphasis of this study. In contrast, optimization-based methods are more precise but often have more extensive processing complexity. The observability-constrained technique addresses the consistency issue, a shortcoming of traditional VIO filter-based algorithms [133]. The EKF/MSCKF and its cutting-edge variations are among the most widely used solutions because they effectively balance accuracy and computational complexity.

A recent study [134] shows that if the air mass's random character is considered, the EKF system states of a MAV are observable. The drag and lift forces on the MAV will directly impact the projected pose and velocity due to the nature of air mass randomization. To make an online update for the uncertainties brought on by these random effects on the precise position of the sensors' reference frames, we contribute with a visual drift augmentation technique to our EKF measurement model. The EKF's ability to tolerate significant disturbances in the MAV's velocity state variable and still converge to the undisturbed estimates is what we target.

4.2.3 Visual Odometry

The main objective of a visual odometry solution is to perform an accurate and precise localization of the robot (ground or aerial vehicle) to estimate its pose during the navigation task. Estimated poses can be on either discrete- or continuous-time manifolds. [114] studied the reliability of the estimated poses on both manifolds using IMU/Visual/GPS sensors. They came to an important conclusion: similar results are produced by the two representations when the camera and IMU are time-synchronized.

In [112], the sliding window pose-graph optimization of the most recent robot states uses global position data with poses predicted by a VIO method. Like [126], pose-graph optimization employs an independent VIO technique to generate pose estimations fused with GPS data. In contrast to [112], the pose-graph in [126] includes an extra node representing the local coordinate frame's origin to confine the absolute orientation. However, these methods

are loosely connected, meaning that a separate VIO algorithm generates the relative pose estimations. Inspired by [112], [126], we present a loosely-coupled strategy that considers the correlations between all measures by including them in a hybrid optimization and filtering problem.

It is demonstrated in [110] that considering all measurement correlations is essential for high-precision estimations in the visual-inertial situation. A tightly-coupled sliding window optimization for visual and inertial data with a loosely connected GPS refinement is presented in [125]. The GPS readings are given the same timestamp as the temporally nearest image to be included in the sliding window because it is believed they would only be accessible at low rates. As opposed to [125], we efficiently compute the global positional factors by closely coupling the global position measurements using the Runge-Kutta 4^{th}-order gyroscope preintegration scheme [103]. This enables the sliding window to incorporate numerous global parameters, each keyframe with barely any additional processing load.

4.2.4 Methodology Background

We highlight the methodology that inspires our study in blue dashed rectangles in Figure 4.2. Where the loosely-coupled fusion strategy [135] is adopted to keep constant computational complexity for real-time performance, along with adding a reset mode for the framework as discussed in [136] as well as an online IMU-camera extrinsic calibration paradigm [118]. Integrating the IMU/GPS readings with the global shutter visual sensor monocular frames raises our localization solution's accuracy level, leveraging the MAV's inertial and global localization information.

Pushing the limits of the Extended Kalman Filter to raise the robustness of our localization solution towards a resilient system, we leverage the high accuracy of optimization to initialize the filter pose states using a novel instant approach utilizing the low-rate noisy GPS readings when available. Sensor fusion on continuous-time (CT) manifolds, such as B-splines [93], suffers from high execution complexity, especially with the time derivatives of high-order manifolds for integrating the IMU measurements in the estimation process. Hence, in our novel method, we avoid this dilemma with a simple spline-fitting approach for the GPS readings during the data pre-processing stage.

4.3 System Architecture

Our core sensor setup consists of an inertial navigation sensor (IMU), a global positioning sensor (GPS), and a monocular camera, as illustrated in Figure 4.3. The pipeline starts with the data acquisition and pre-processing for the initialization process, as discussed in Section 4.3.1. The initialization is an optimization-based phase (see Algorithm 2) with considerably low complexity and processing time whose output is an instant metric-scaled pose estimated from the camera, GPS, and gyroscope readings. Then, an ES-EKF (see Algorithm 3) whose dynamic model is given in Section 4.3.2 is applied to estimate all the system states, including the MAV's trajectory, velocity, and a scale factor

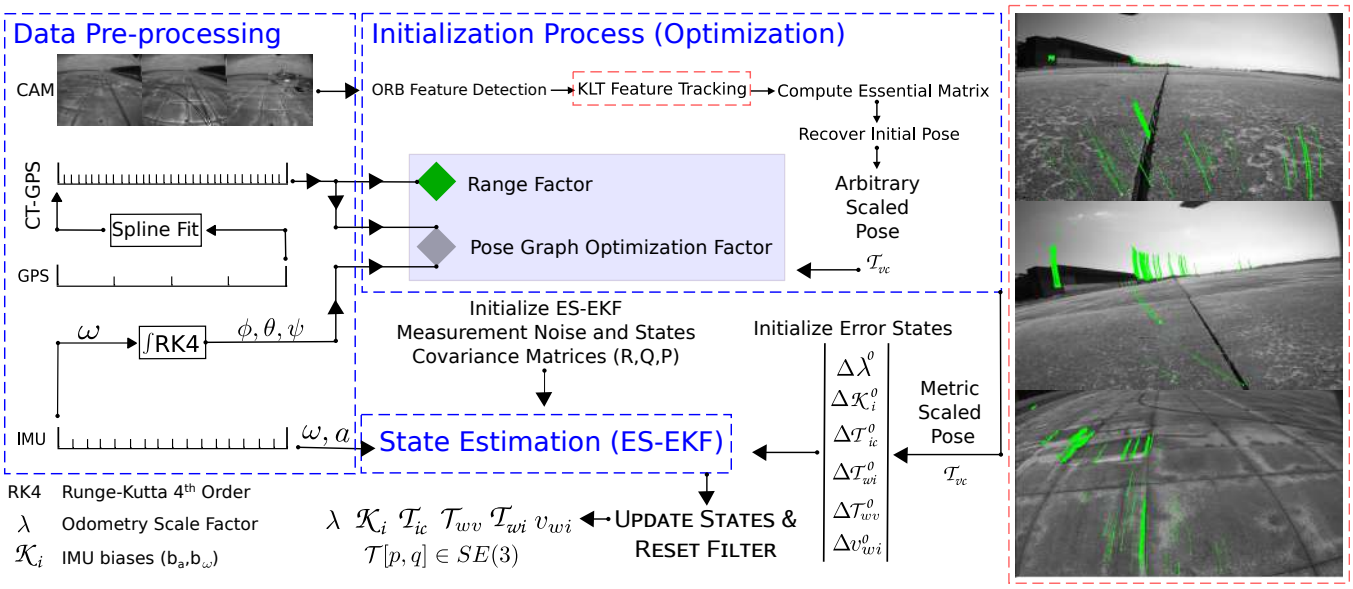


Figure 4.3: Overview of our proposed entire system architecture.

to recover the initially estimated trajectory in case of GPS readings loss. Finally, we present the measurement model in Section 4.3.3 with a novel false pose augmentation paradigm to ensure the observability of all the filter states as analyzed in Section 4.5.

The state representation is a 31-elements state vector $\mathcal{X}$:

$$\mathcal{X} = \begin{bmatrix} {p_w^i}^\top & {v_w^i}^\top & {q_w^i}^\top & {b_\omega}^\top & {b_a}^\top & \lambda & {p_i^c}^\top & {q_i^c}^\top & {p_v^w}^\top & {q_v^w}^\top \end{bmatrix}^\top , \tag{4.1}$$

where p_w^i is the position of the IMU in the world frame[1] (w), its velocity v_w^i, and its attitude rotation quaternion q_w^i describing a rotation from the IMU frame (i) into the world frame (w). b_ω and b_a are the gyro and acceleration biases along with the visual odometry scale factor λ. $R_{(q)}$ is the quaternion q rotational matrix, g is the gravity vector aligned with the world frame (w), and $\Omega(\omega)$ is the quaternion-multiplication matrix of ω.

The IMU/Camera calibration states are the rotation from the camera frame into the IMU frame q_i^c , and the position of the camera center w.r.t. the IMU frame p_i^c.

Finally, the visual attitude drifts between the black-boxed visual frame[2] (v) and the world inertial frame (w) are reflected in q_v^w and the translational ones in p_v^w. We assume that all the visual drifts are spatial without any temporal drifts,i.e., the IMU and the camera have synchronized timestamps.

The corresponding 28-elements error state vector is defined by:

$$\tilde{x} = \begin{bmatrix} {\Delta p_w^i}^\top & {\Delta v_w^i}^\top & {\delta\theta_w^i}^\top & {\Delta b_\omega}^\top & {\Delta b_a}^\top & \Delta\lambda & {\Delta p_i^c}^\top & {\delta\theta_i^c}^\top & {\Delta p_v^w}^\top & {\delta\theta_v^w}^\top \end{bmatrix}^\top , \tag{4.2}$$

[1] World frame is a gravity-aligned frame
[2] Vision frame is the frame to which the camera pose is estimated in the black-box vision framework

as the difference of an estimate $\hat{x}$ to its quantity x, i.e. $\tilde{x} = x - \hat{x}$. We apply this to all state variables except the error quaternions, which are defined by:

$$\delta q_y^x = q_y^x \otimes \hat{q_y^x} \approx \begin{bmatrix} \frac{1}{2}\delta\theta_y^x & 1 \end{bmatrix}^\top. \tag{4.3}$$

This error quaternion representation increases the numerical stability of the estimation process and handles the quaternion in its minimal representation [137].

4.3.1 State Estimator Initialization

An incremental Structure from Motion (SfM) algorithm [138] is applied to the acquired image frames, whose goal is to retrieve the camera poses and 3D structure of the scene, based on the five-point algorithm proposed in [64]. ORB features are detected, and the highest quality points are tracked between 10 consecutive frames using the KLT method [56].

To solve the arbitrary-scale problem of the camera trajectory only we follow the efficient level 1 optimization process modeled in Section 3.3.3 of Chapter 3, by applying an on-manifold cumulative B-spline[3] interpolation [93] to synthesize a very smooth continuous-time (CT) trajectory in R^3 from the low-rate noisy GPS readings.

Algorithm 2 Bootstrapping: Pose Graph Optimization and Range Factors

Input: RGB frames (c), Camera matrix ($\mathcal{K}_c$), GPS readings (DT-GPS), IMU readings ($\mathcal{I}$)
Output: Metric-scaled Trajectory ($\mathcal{T}_{vc}[p_v^c,\ q_v^c] \in \mathrm{SE}(3)$)

1: $\mathcal{T}_{vc}^0 \Leftarrow$ KLT-VO (c, $\mathcal{K}_c$) ▷ Arbitrary-scaled pose
2: $p(u) \Leftarrow$ spline_fit (DT-GPS) ▷ CT-GPS by Equation (3.28)
3: $[\phi, \theta, \psi] \Leftarrow$ RK4 ($\mathcal{I}_{gyro}(\omega)$) ▷ Initial orientations
4: **while** *not converged* **do** ▷ Initial Trajectory Optimization
5: $\quad \mathcal{T}_{vc} \Leftarrow$ optimize ($\mathcal{T}_{vc}^0$, $p(u)$, $[\phi, \theta, \psi]$) ▷ Equation (3.30)
6: **end while**

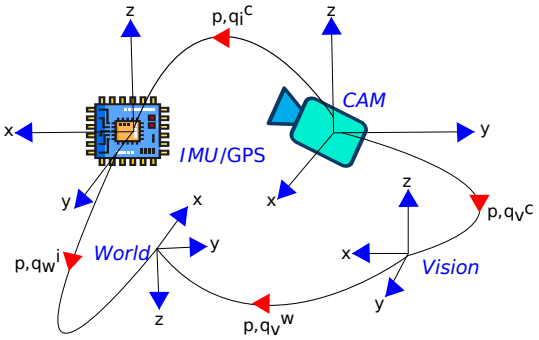


Figure 4.4: The frames of reference annotations.

[3] https://github.com/AbanobSoliman/B-splines

4.3.2 Dynamic Model

The core state estimation is performed by fusing the RGB camera frames and the IMU reading using an Error-States Extended Kalman Filter (ES-EKF). Figure 4.4 illustrates the inter-sensor extrinsic relation between the IMU/GPS sensors and a monocular camera.

To use the linear states estimator, we assume that the IMU measurements contain a particular bias $b_a \in \mathcal{N}(0, \sigma_{ba})$, $b_\omega \in \mathcal{N}(0, \sigma_{b\omega})$ and a white Gaussian noise $n_a \in \mathcal{N}(0, \sigma_a)$, $n_\omega \in \mathcal{N}(0, \sigma_\omega)$.

Thus, the real angular velocities ω and accelerations a in the IMU body frame (i) can be written as:

$$\omega = \omega_m - b_\omega - n_\omega \quad and \quad a = a_m - b_a - n_a, \tag{4.4}$$

where the subscript m denotes the measured value. The dynamics of the non-static biases are modeled as a random process:

$$\dot{b_\omega} = n_{b_\omega}, \quad \dot{b_a} = n_{b_a}. \tag{4.5}$$

The standard deviation $\sigma_{b_\omega}, \sigma_{b_a}, \sigma_w, \sigma_a$ values are generally given by the IMU manufacturer's data in Allan deviation plots. For discrete time steps, as it will be applied in the filter. We need to convert these values according to their units:

$$d\sigma^2_{\omega,a} = \frac{\sigma^2_{\omega,a}}{\nabla t}, \quad d\sigma^2_{b_{\omega,a}} = \sigma^2_{b_{\omega,a}} * \nabla t. \tag{4.6}$$

The following differential equations govern IMU states propagation:

$$\begin{aligned}
\dot{p^i_w} &= v^i_w, \\
\dot{v^i_w} &= R^\top_{(q^i_w)}(a_m - b_a - n_a) - g, \\
\dot{q^i_w} &= \tfrac{1}{2}\Omega(\omega_m - b_\omega - n_\omega)q^i_w, \\
\dot{b_\omega} &= n_{b_\omega},\ \dot{b_a} = n_{b_a},\ \dot{\lambda} = 0, \\
\dot{p^c_i} &= 0,\ \dot{q^c_i} = 0,\ \dot{p^w_v} = 0,\ \dot{q^w_v} = 0,
\end{aligned} \tag{4.7}$$

For the quaternion integration inside the ES-EKF, we use the first order integrator defined in [137] as:

$$\begin{aligned}
\bar{w} &= \tfrac{\omega_{k+1}+\omega_k}{2}, \quad \kappa = \tfrac{1}{2}\Omega(\bar{\omega})\Delta t, \\
\hat{q^i_{w\,k+1}} &= [e^\kappa + \tfrac{\Delta t^2}{48}(\Omega(\omega_{k+1})\Omega(\omega_k) - \Omega(\omega_k)\Omega(\omega_{k+1}))]\hat{q^i_{w\,k}}.
\end{aligned} \tag{4.8}$$

where the hat term $\hat{}$, means the estimated value. The exponential term e^κ is expanded by the Maclaurin series.

The states transition matrix F_d is modeled as:

$$F_d = \begin{bmatrix} I_{d_3} & \Delta t & A & B & -R^{\top}_{(\hat{q_w^i})}\frac{\Delta t^2}{2} & 0_{3\times 13} \\ 0_3 & I_{d_3} & C & D & -R^{\top}_{(\hat{q_w^i})}\Delta t & 0_{3\times 13} \\ 0_3 & 0_3 & E & F & 0_3 & 0_{3\times 13} \\ 0_3 & 0_3 & 0_3 & I_{d_3} & 0_3 & 0_{3\times 13} \\ 0_3 & 0_3 & 0_3 & 0_3 & I_{d_3} & 0_{3\times 13} \\ 0_{13\times 3} & 0_{13\times 3} & 0_{13\times 3} & 0_{13\times 3} & 0_{13\times 3} & I_{d_{13}} \end{bmatrix}. \tag{4.9}$$

Then, we apply the small angle approximation for which $|\omega| \rightarrow 0$, apply de l'Hopital rule and obtain a compact solution for the six matrix blocks A, B, C, D, E, F [137]:

$$\begin{aligned} A &= -R^{\top}_{(\hat{q_w^i})} \lfloor \hat{a} \rfloor_{\times} (\tfrac{\Delta t^2}{2!} - \tfrac{\Delta t^3}{3!} \lfloor \hat{\omega} \rfloor_{\times} + \tfrac{\Delta t^4}{4!} \lfloor \hat{\omega} \rfloor^2_{\times}), \\ B &= -R^{\top}_{(\hat{q_w^i})} \lfloor \hat{a} \rfloor_{\times} (\tfrac{-\Delta t^3}{3!} + \tfrac{\Delta t^4}{4!} \lfloor \hat{\omega} \rfloor_{\times} - \tfrac{\Delta t^5}{5!} \lfloor \hat{\omega} \rfloor^2_{\times}), \\ C &= -R^{\top}_{(\hat{q_w^i})} \lfloor \hat{a} \rfloor_{\times} (\Delta t - \tfrac{\Delta t^2}{2!} \lfloor \hat{\omega} \rfloor_{\times} + \tfrac{\Delta t^3}{3!} \lfloor \hat{\omega} \rfloor^2_{\times}), \\ D &= -A, \\ E &= I_{d_3} - \Delta t \lfloor \hat{\omega} \rfloor_{\times} + \tfrac{\Delta t^2}{2!} \lfloor \hat{\omega} \rfloor^2_{\times}, \\ F &= -\Delta t + \tfrac{\Delta t^2}{2!} \lfloor \hat{\omega} \rfloor_{\times} - \tfrac{\Delta t^3}{3!} \lfloor \hat{\omega} \rfloor^2_{\times}, \end{aligned} \tag{4.10}$$

with $\hat{\omega} = \omega_m - \hat{b_\omega}$, $\hat{a} = a_m - \hat{b_a}$ and $\lfloor \hat{\omega} \rfloor_{\times}$, $\lfloor \hat{a} \rfloor_{\times}$ the skew-symmetric matrices for IMU readings.

We can now derive the discrete-time input noise covariance matrix Q_d as:

$$Q_d = \int_{\Delta t} F_d(\tau) G_c Q_c G_c^{\top} F_d(\tau)^{\top} d\tau, \tag{4.11}$$

where Q_c is the CT process noise covariance, and G_c is calculated in the form:

$$G_c = \begin{bmatrix} 0_3 & 0_3 & 0_3 & 0_3 \\ -R^{\top}_{(\hat{q_w^i})} & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & I_{d_3} & 0_3 \\ 0_3 & 0_3 & 0_3 & I_{d_3} \\ 0_3 & -I_{d_3} & 0_3 & 0_3 \\ 0_{13\times 3} & 0_{13\times 3} & 0_{13\times 3} & 0_{13\times 3} \end{bmatrix}. \tag{4.12}$$

The closed-form solution of the complete derivation of the Q_d covariance matrix is given in detail in Appendix C.

Finally, the propagated state covariance matrix computation is defined as:

$$P_{k+1|k} = F_d P_{k|k} F_d^\top + Q_d. \tag{4.13}$$

4.3.3 Measurement Model

The main contribution of our measurement model for an observable ES-EKF is the false relative pose augmentation methodology of the visual drift quaternion state at the previous time step (k) updated with the current camera measurement at a time (k+1) and modeled as:

$$q_v^w(k) = \hat{q}_w^i(k)^{-1} \otimes \hat{q}_i^c(k)^{-1} \otimes q_v^c(k+1). \tag{4.14}$$

The camera position measurement model yields the position of the camera w.r.t. the vision frame p_v^c. The error in measurement modeled as $\tilde{z}_p$ and linearized as $\tilde{z}_{pL}$:

$$\tilde{z}_p = z_p - \hat{z}_p = p_v^c - R_{(\hat{q}_v^w)}^\top (\hat{p}_w^i + R_{(\hat{q}_w^i)}^\top \hat{p}_i^c)\hat{\lambda} \doteq \tilde{z}_{pL} = H_p \tilde{x}, \tag{4.15}$$

with

$$H_p^\top = \begin{bmatrix} R_{(\hat{q}_v^w)}^\top \hat{\lambda} \\ 0_{3\times 3} \\ -R_{(\hat{q}_v^w)}^\top R_{(\hat{q}_w^i)}^\top \lfloor \hat{p}_i^c \rfloor_\times \hat{\lambda} \\ 0_{6x3} \\ R_{(\hat{q}_v^w)}^\top R_{(\hat{q}_w^i)}^\top \hat{p}_i^c + R_{(\hat{q}_v^w)}^\top \hat{p}_w^i \\ R_{(\hat{q}_v^w)}^\top R_{(\hat{q}_w^i)}^\top \hat{\lambda} \\ 0_{6x3} \\ -R_{(\hat{q}_v^w)}^\top \left\lfloor (\hat{p}_w^i + R_{(\hat{q}_w^i)}^\top \hat{p}_i^c)\hat{\lambda} \right\rfloor_\times \end{bmatrix}, \tag{4.16}$$

using the definition of the error-quaternion

$$\begin{aligned} q_w^i &= \delta q_w^i \otimes \hat{q}_w^i, \\ R_{(q_w^i)} &\approx (I_{d_3} - \lfloor \delta\theta_w^i \rfloor_\times) R_{(\hat{q}_w^i)}. \end{aligned} \tag{4.17}$$

The vision algorithm yields the rotation from the camera frame into the vision frame q_v^c . We can model the error measurement as,

$$\tilde{z}_q = z_q - \hat{z}_q = q_i^c \otimes q_w^i \otimes q_v^w \otimes (q_i^c \otimes \hat{q}_w^i \otimes \hat{q}_v^w)^{-1}. \tag{4.18}$$

Finally, the measurements Jacobian H in $\tilde{z} = H.\tilde{x}$ is calculated based on the method in [136], and can be stacked

together in the form,

$$\begin{bmatrix} \tilde{z}_p \\ \tilde{z}_q \end{bmatrix} = \begin{bmatrix} H_p & & & & & \\ 0_{3x6} & \tilde{H}_q^{wi} & 0_{3x10} & \tilde{H}_q^{ic} & 0_{3\times3} & \tilde{H}_q^{vw} \end{bmatrix} \tilde{x}. \tag{4.19}$$

with the Jacobian matrices $\tilde{H}_q^{xy}$, known as the right Jacobian of $\mathrm{SO}(3)$, and are defined as:

$$\begin{aligned} \tilde{H}_q^{xy} &= J_r(\theta_x^y) = \lim_{\delta\theta\to 0} \tfrac{Log(Exp(\theta)\otimes Exp(\theta+\delta\theta))}{\delta\theta}, \\ J_r(\theta_x^y) &= I_{d_3} - (\tfrac{1-cos\|\delta\theta\|}{\|\delta\theta\|^2}) \lfloor \delta\theta_x^y \rfloor_\times + (\tfrac{\|\delta\theta\|-sin\|\delta\theta\|}{\|\delta\theta\|^3}) \lfloor \delta\theta_x^y \rfloor_\times^2 . \end{aligned} \tag{4.20}$$

4.3.4 States Update

To update the framework for the current time step (k+1), we compute the innovation term S, Kalman gain K, and the states correction vector $\hat{\tilde{x}}$ defined as:

$$S = HPH^\top + \mathcal{R}, \quad K = PH^\top S^{-1}, \quad \hat{\tilde{x}} = K\tilde{z} \ . \tag{4.21}$$

The error state covariance is updated as follows:

$$P_{k+1|k+1} = (I_{d_{28}} - KH)P_{k+1|k}(I_{d_{28}} - KH)^\top + K\mathcal{R}K^\top, \tag{4.22}$$

where $\mathcal{R}_{[6x6]} = diag(\mathcal{R}_{position}, \mathcal{R}_{orientation})$ is the measurement noise covariance matrix.

The error quaternion is calculated by (4.3) to ensure its unit length, then update the states vector: $\mathcal{X}_{k+1} = \mathcal{X}_k + \hat{\tilde{x}}$.

For quaternions state update:

$$\hat{q}_{k+1} = \frac{\begin{bmatrix} 1 & \frac{1}{2}\delta\theta_{k+1}^1 & \frac{1}{2}\delta\theta_{k+1}^2 & \frac{1}{2}\delta\theta_{k+1}^3 \end{bmatrix} \otimes \hat{q}_k}{\left\| \begin{bmatrix} 1 & \frac{1}{2}\delta\theta_{k+1}^1 & \frac{1}{2}\delta\theta_{k+1}^2 & \frac{1}{2}\delta\theta_{k+1}^3 \end{bmatrix} \otimes \hat{q}_k \right\|}, \tag{4.23}$$

where $\delta\theta_{k+1}^i$ is the i^{th} error state of this quaternion.

4.3.5 Reset Mode

The ES-EKF reset mode is performed by setting $\hat{\tilde{x}} \leftarrow 0$ and $P \leftarrow GPG^\top$, where G is the Jacobian matrix defined by,

$$\begin{aligned} G &= diag(I_{d_6}, J_{r_{wi}}, I_{d_{10}}, J_{r_{ic}}, I_{d_3}, J_{r_{vw}}), \\ J_{r_{xy}} &= \tfrac{\partial\delta\theta_x^{y^+}}{\partial\delta\theta_x^y} = I_{d_3} - \tfrac{1}{2}\left\lfloor \hat{\delta\theta_x^y} \right\rfloor_\times . \end{aligned} \tag{4.24}$$

Algorithm 3 End-to-End State Estimation Scheme

Input: IMU Readings, Initial Optimized Trajectory $\mathcal{T}_{vc}$
Output: FilterStates $\mathcal{X} = \{\lambda, \mathcal{K}_i[b_a,\ b_\omega], \mathcal{T}_{ic}, \mathcal{T}_{wv}, \mathcal{T}_{wi}, v_{wi}\}\ ,\ \forall T[p,q] \in \mathrm{SE}(3)$

$P, Q_c, \mathcal{R}$_initialization, FilterStates_initialization
ErrorStates_initialization=0
while imuRead **do**
    Read LastStep (k) P, FilterStates, ErrorStates
    Read LastStep (k) IMU (Accel, Gyro) values
    Read Current (k+1) IMU (Accel, Gyro) values
    Step 1: Propagate IMU states ▷ Equation (4.7)
    Step 2: Calculate F_d and Q_d ▷ Equations (4.9),(4.11)
    Step 3: Compute P state covariance matrix ▷ Equation (4.13)
    if camRead **then**
        Read Current (k+1) CAM $\mathcal{T}_{vc}$ values ▷ Metric-scaled pose
        Step 4: Estimate False Pose ▷ Equation (4.14)
        Step 5: Calculate $\tilde{z}$, H ▷ Equation (4.15)
        Step 6: Calculate S, K, ErrorStates $\hat{\tilde{x}}$, P ▷ Equations (4.21),(4.22)
        Step 7: Update: FilterStates += ErrorStates
        Step 8: RESET $\hat{\tilde{x}} = 0$, P ▷ Equation (4.24)
    end if
end while

4.4 Experiments

4.4.1 Setup

An extensive quantitative and qualitative evaluation is carried out to validate all the state estimation process aspects. This thorough performance analysis is run on the EuRoC benchmark [26] for indoor system global positioning evaluation in low-speed flights and on the Fast Flight dataset [124] for outdoor experimentation at relatively high-speed flights. For a fair comparison, all the pipeline processing stages in both Algorithms 2,3 are performed on a 16 GB RAM laptop computer running 64-bit Ubuntu 20.04.3 LTS with AMD(R) Ryzen 7 4800h ×16 cores 2.9 GHz processor and a Radeon RTX NV166 Renoir graphics card. In Table 4.1, we represent quantitative insights of our experiments settings regarding the benchmarks statistical data and the sensors parameters in-detail.

The front-end of the pipeline, including both the data acquisition and pre-processing steps, is developed as Python API that sends the optimization variables to the factor graph implemented in C++ using the Ceres solver [109] to achieve the lowest possible system latency before the state estimation process. The Sparse Normal Cholesky linear solver by the Ceres solver is employed to solve the least-squares convex optimization problem formulated in Equation (3.30) along with the Levenberg-Marquardt trust region strategy with the automatic differentiation tool for Jacobian calculations. The sparse Schur linear method is applied to utilize the Schur complement for a more robust and fast optimization process. The pipeline's back-end for the state estimation process is developed entirely in MATLAB[4] and all the initialization parameters are given explicitly in Table 4.2.

[4] https://github.com/AbanobSoliman/VIO_RGB_IMU

Table 4.1: Insights of our experiments statistical information and sensor settings.

	Parameter	EuRoC Benchmark [26]				Fast Flight Dataset [124]			
Stats	Total processed sequences	6 (Vicon room)				4 (Airport runway)			
	Total sequences duration	11.6111 minutes				8.8867 minutes			
	Total sequences length	411.5425 meters				2539.0599[1] meters			
	Maximum speed	2.3 [m/s]				17.5 [m/s]			
Camera	Total processed frames	13736				21312			
	Frame Resolution	752×480 pixels				960×800 pixels			
	Intrinsics (f_x, f_y, c_x, c_y)	458.65	457.30	367.22	248.38	606.58	606.73	474.93	402.28
	Distortion (k_1, k_2, p_1, p_2)	-0.2834	0.0739	0.0001	1.8e-5	-0.0147	-0.0058	0.0072	-0.0046
	Camera-IMU p_i^c(x,y,z,1) [m]	-0.0216	-0.0647	0.0098	1.0000	0.1058	-0.0177	-0.0089	1.0000
	Camera-IMU q_i^c(x,y,z,w) [-]	-0.0077	0.0105	0.7018	0.7123	-1.0000	0.0042	-0.0039	0.0015
	Frame rate	20 [Hz]				40 [Hz]			
IMU	Gyroscope noise density (σ_{n_ω})	$1.6968 \times 10^{-4}\ [rad/s/\sqrt{Hz}]$				$6.1087 \times 10^{-5}\ [rad/s/\sqrt{Hz}]$			
	Gyroscope random walk ($\sigma_{n_{b_\omega}}$)	$1.9393 \times 10^{-5}\ [rad/s^2/\sqrt{Hz}]$				$9.1548 \times 10^{-5}\ [rad/s^2/\sqrt{Hz}]$			
	Accelerometer noise density (σ_{n_a})	$2.0000 \times 10^{-3}\ [m/s^2/\sqrt{Hz}]$				$1.3734 \times 10^{-3}\ [m/s^2/\sqrt{Hz}]$			
	Accelerometer random walk ($\sigma_{n_{b_a}}$)	$3.0000 \times 10^{-3}\ [m/s^3/\sqrt{Hz}]$				$2.7468 \times 10^{-3}\ [m/s^3/\sqrt{Hz}]$			
	Data rate ($1/\Delta t$)	200 [Hz]				200 [Hz]			
GPS	Type / Operation	Indoors / Vicon System				Outdoors / Satellite Triangulation			
	Readings	X [m], Y [m], Z [m]				Long. [deg], Lat. [deg], Alt. [m]			
	Data rate	1 [Hz] (Down-sampled)				5 [Hz]			

[1] Denotes the exact value of the total trajectories lengths for all Fast Flight dataset sequences shown on the x-axis of Figure 4.1 (≈2.5 [km]).

Table 4.2: The ES-EKF initialization parameters for both EuRoC and Fast Flight sequences.

Parameter Initialization	EuRoC Benchmark [26]	Fast Flight Dataset [124]
28-elements Error States Vector ($\hat{\tilde{x}}$)	$0_{28\times1}$	$0_{28\times1}$
31-elements States Vector[1] ($\mathcal{X}$)	$\begin{pmatrix}0_{3\times1}\ 0_{3\times1}\ \bar{q}^\top\ 0_{3\times1}\ 0_{3\times1}\ 1\ p_i^{c\top}\ q_i^{c\top}\ 0_{3\times1}\ \bar{q}^\top\end{pmatrix}^\top$	
States Propagation Covariance (P)	$10^{-7} \times I_{d_{28}}$	$10^{-12} \times I_{d_{28}}$
CT Process Noise Covariance[2] (Q_c)	$diag(d\sigma_{n_a}^2 I_{d_3}, d\sigma_{n_{b_a}}^2 I_{d_3}, d\sigma_{n_\omega}^2 I_{d_3}, d\sigma_{n_{b_\omega}}^2 I_{d_3})$	
Measurement Noise Covariance ($\mathcal{R}$)	$diag(0.01, 0.01, 0.03, 10^{-4}, 10^{-4}, 10^{-4})$	

[1] $\bar{q}$ denotes the unity quaternion [0,0,0,1].

[2] IMU noise density values for each dataset are from Table 4.1 and discretized using Equation (4.6).

The performance analysis is done using the two trajectory evaluation metrics: Root Mean Square Error (RMSE) for the Fast Flight dataset compared to the GPS trajectory p_{gps}, and the RMS Absolute Trajectory Error (ATE) for the EuRoC benchmark compared to the ground truth trajectory T_{gt} provided with Vicon room sequences. The positional RMSE metric for the Fast Flight sequences is chosen because ground truth GPS trajectories exist with unknown ground truth orientations. Whereas, for EuRoC sequences, we select the RMS ATE metric for two reasons: 1. the Vicon system provides ground truth poses (positions and orientations), and 2. to ensure a fair comparison with the latest state-of-the-art methods based on the same error metric. The two trajectory evaluation metrics are formulated as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||\hat{p}(i) - p_{gps}(i)||^2}, \;\; \text{ATE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \left|\left|p(T_{gt}^{-1}(i) T_{rel} \hat{T}(i))\right|\right|^2} \; [m], \tag{4.25}$$

where $\hat{p}$ is the estimated translation vector of the $\hat{T} \in \text{SE}(3)$ trajectory. $p(.)$ is the translation vector of the $T \in \text{SE}(3)$ pose, and T_{rel} is rigid-body transformation corresponding to the least-squares solution that maps the $\hat{T}$ trajectory onto the T_{gt} trajectory calculated by optimization. We set it constant for all sequences that belong to the same benchmark.

4.4.2 The EuRoC MAV Benchmark

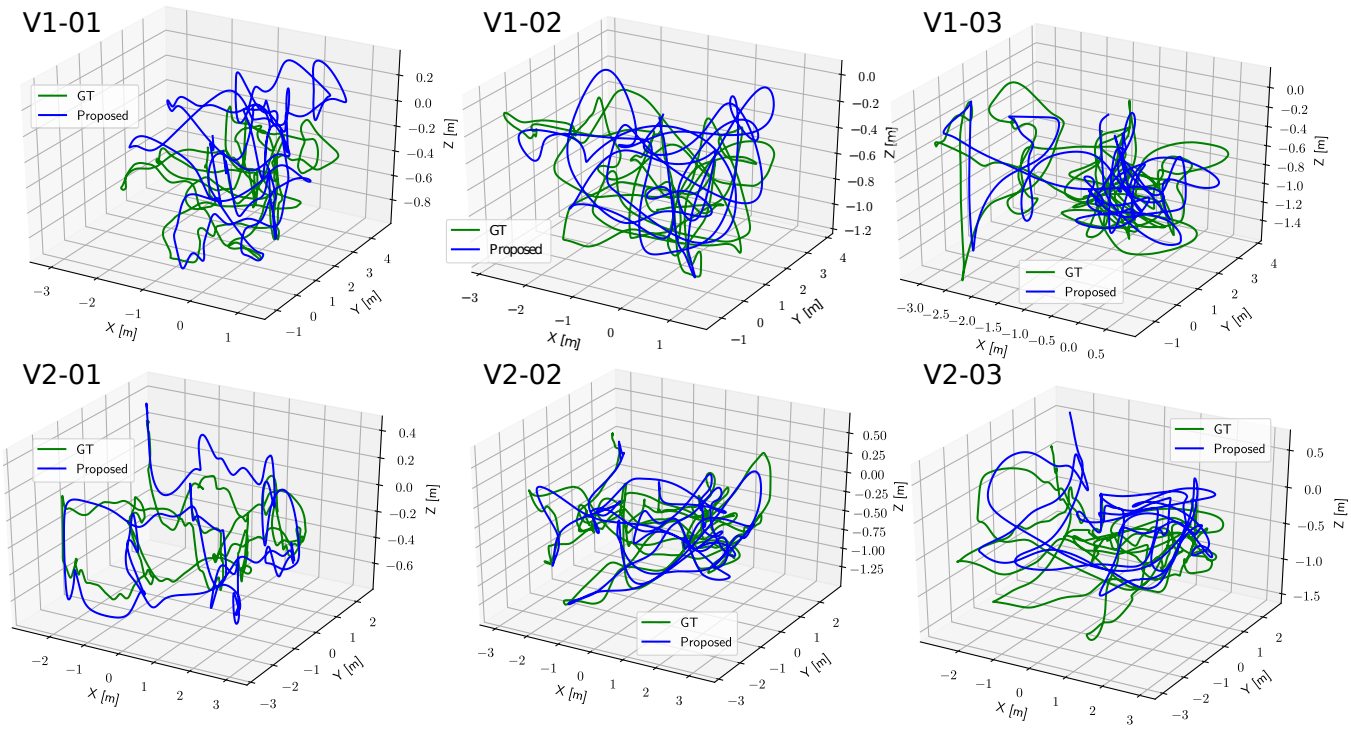


Figure 4.5: EuRoC 3D trajectory estimation compared to the ground truth.

The two main characteristics of the EuRoC MAV sequences are the complex combined 6-DoF motions and the

Table 4.3: Ablation study on the contribution of the GPS sensor on the system accuracy. The latest state-of-the-art (monocular/stereo) VI-SLAM systems are compared to our proposed trajectory initialization (PGO factors) and ES-EKF state estimation methods. **Bold** denotes the most accurate.

	Method	EuRoC Benchmark [26] (RMS ATE [m])						Avg.
		V1-01	V1-02	V1-03	V2-01	V2-02	V2-03	
Mono-VI	OKVIS [110]	0.090	0.200	0.240	0.130	0.160	0.290	0.185
	ROVIO [42]	0.100	0.100	0.140	0.120	0.140	0.140	0.123
	VINS-Mono [5]	0.047	0.066	0.180	0.056	0.090	0.244	0.114
	OpenVINS [111]	0.056	0.072	0.069	0.098	0.061	0.286	0.107
	CodeVIO[1] [89]	0.054	0.071	0.068	0.097	0.061	0.275	0.104
	[2] [127]	0.034	0.035	0.042	0.026	0.033	0.057	0.038
Stereo-VI	VINS-Fusion [112]	0.076	0.069	0.114	0.066	0.091	0.096	0.085
	BASALT [69]	0.040	0.020	0.030	0.030	0.020	0.050	0.032
	Kimera [113]	0.050	0.110	0.120	0.070	0.100	0.190	0.107
	ORB-SLAM3 [43]	0.038	0.014	0.024	0.032	0.014	0.024	0.024
Mono-(V/I/G)[3]	CT (V+I+G) [114]	0.024	0.014	**0.011**	0.012	0.010	**0.010**	0.014
	CT (V+G) [114]	0.011	0.013	0.012	0.009	**0.008**	0.012	**0.011**
	CT (I+G) [114]	0.062	0.102	0.117	0.112	0.164	0.363	0.153
	DT (V+I+G) [114]	0.016	0.024	0.018	0.009	0.018	0.033	0.020
	DT (V+G) [114]	0.010	0.025	0.024	0.010	0.012	0.029	0.018
	DT (I+G) [114]	0.139	0.137	0.138	0.138	0.138	0.139	0.138
	Ours (PGO)	**0.008**	0.017[4]	0.023[4]	**0.008**	0.022	0.025[4]	0.017
	Ours (ES-EKF)	0.009	**0.012**	**0.011**	0.010	0.011	**0.010**	**0.011**

[1] Denotes the only learning-based baseline in the table and incorporates point clouds using LiDAR.

[2] Denotes values from the original work with four GPS readings connected to each optimization state.

[3] V,I,G: Vision, IMU, and GPS (generated from the Vicon system readings).

[4] Denotes KLT-VO tracks features in 5 consecutive frames instead of 10 due to the rapid movement of the MAV.

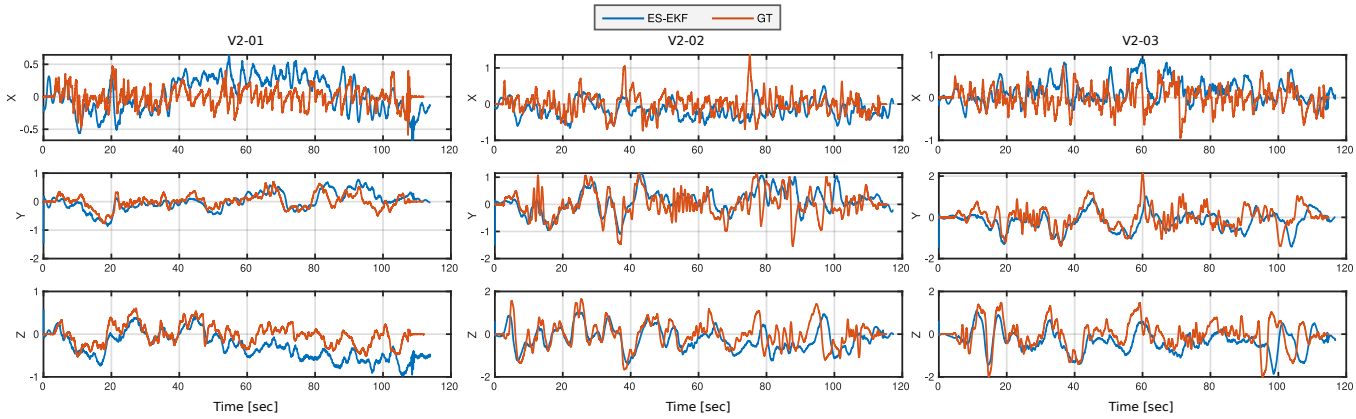


Figure 4.6: Estimated velocity profile validation with the ground truth. Comparison of sample sequences from EuRoC benchmark.

relatively low speeds compared to the Fast Flight sequences. These prominent characteristics allow an accurate evaluation of the ES-EKF marginally stable states, such as the velocity and the visual drift. In Table 4.3, we report the ATE values as an evaluation parameter for the trajectory estimation accuracy compared to the ground truth. Moreover, Table 4.3 shows an ablation study that investigates the contribution of the GPS sensor to the overall estimation accuracy, especially for the monocular vision-based optimization methods: ours (PGO) and the recent work of [114]. The selection of the six Vicon room sequences from the EuRoC benchmark is because a comparison with an alternative method such as [114] incorporating GPS signals simulated from the Vicon system readings, better emphasizes the findings of this ablation study.

A prominent finding of this ablation study is that vision is the most significant type of sensor. In most sequences, the lowest ATE is obtained by fusing the camera trajectory from the vision KLT-based SfM algorithm to a gravity-aligned frame using the noisy simulated GPS data, and adding inertial measurements does not provide a measurable benefit in this case. However, adding the gyroscope measurements to the visual-GPS fusion has led to the least ATE achieved by our PGO model compared to all other discrete-time (DT) methods. Figures 4.5,4.6 show our trajectory and velocity estimations after incorporating the accelerometer readings in the ES-EKF model resulting in the lowest achievable errors that can compete with the continuous-time optimization model in [114].

4.4.3 The Fast Flight Dataset

Table 4.4: Ablation study on the effect of the high MAV speed on the accuracy of the filtering approaches compared to optimization approaches. The first sub-section compares monocular (VINS-Mono and Ours) to stereo (OKVIS) optimization-based VI systems. The second sub-section compares stereo filtering-based approaches to our proposed method. **Bold** denotes the most accurate in each sub-section.

Method	Fast Flight [124] (RMSE [m])				Avg.
	gps5	gps10	gps15	gps175	
OKVIS [110]	3.224	4.987	3.985	4.535	4.183
VINS-Mono [5]	5.542	8.753	2.875	3.452	5.156
Ours (PGO)	**0.417**	**0.759**	**0.180**	**0.927**	**0.571**
S-MSCKF [124]	4.985	2.751	**4.752**	**7.852**	**5.085**
S-UKF-LG [129]	4.875	2.589	5.128	7.865	5.114
S-IEKF [129]	4.986	**2.544**	5.124	8.152	5.201
Ours (ES-EKF)	**4.751**	7.924	7.221	9.488	7.346

The main observation, which is validated upon both the EuRoC and Fast Flight sequences (see Table 4.4 and Figure 4.7), is that for velocities less than 5 [m/s], the monocular loosely-coupled ES-EKF can achieve considerably low estimation errors concerning the other filter- or optimization-based methods. For velocities more than 5 [m/s], our proposed optimization-based initialization scores the lowest RMSE compared to all other methods in comparison in Table 4.4. On the contrary, the monocular ES-EKF scores the lowest RMSE, especially for velocities more than 10

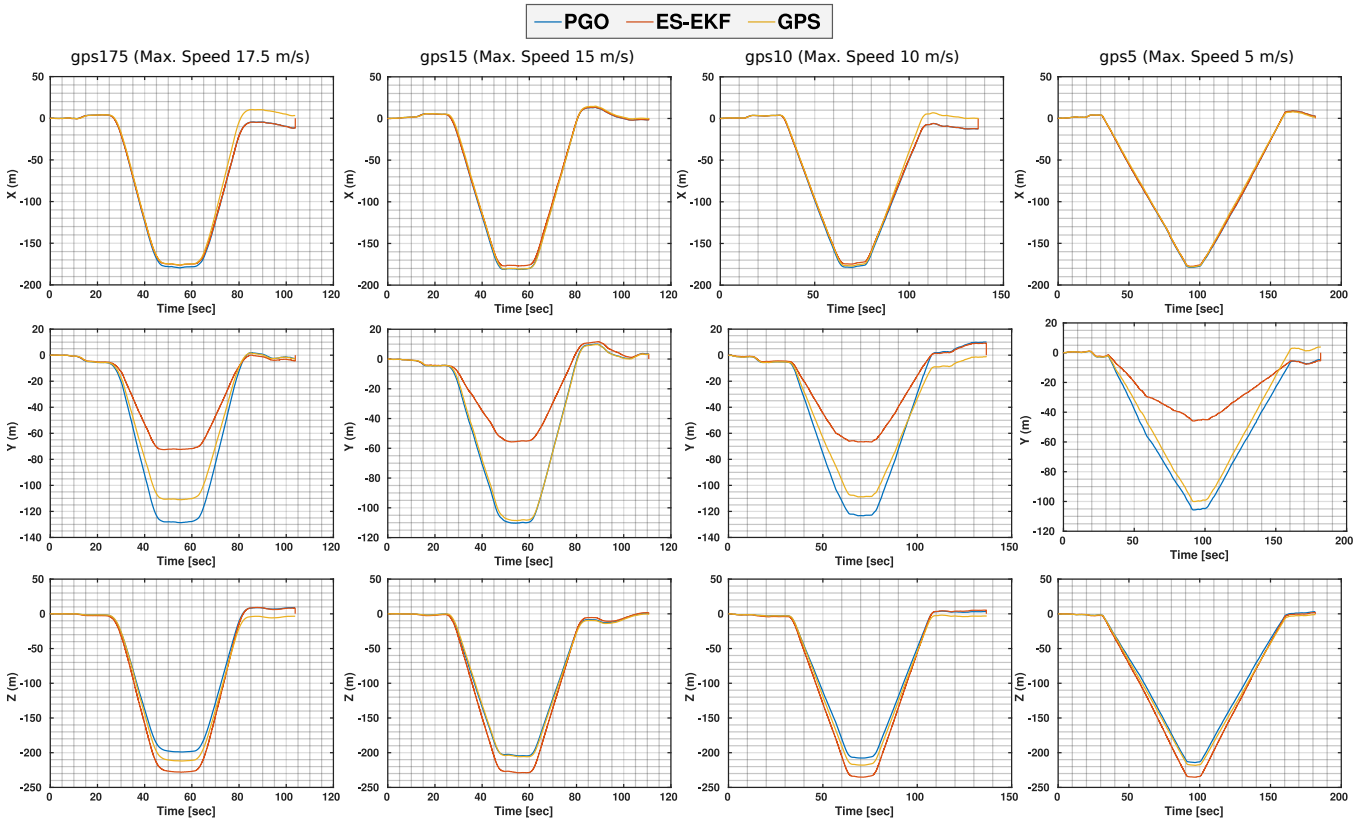


Figure 4.7: Fast Flight (X (top) - Y (middle) - Z (bottom)) trajectory estimation compared to the GPS readings.

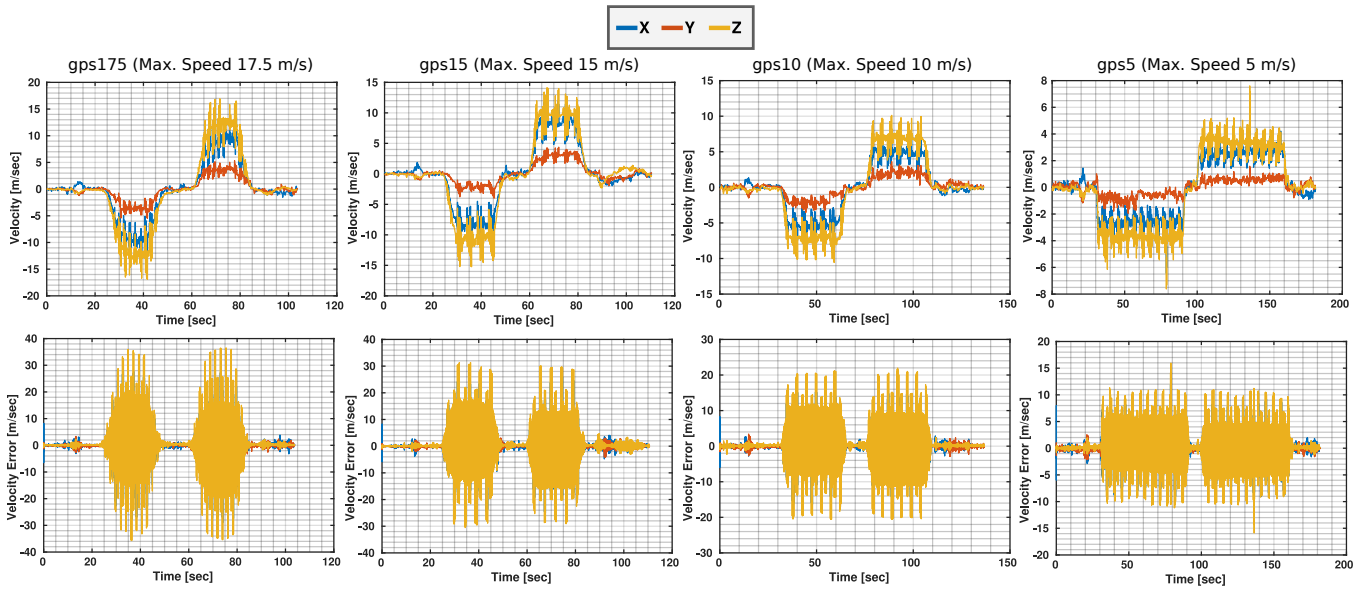


Figure 4.8: **Top**: Fast Flight velocity profile validation with the top speed of each sequence. **Bottom**: velocity error states in the ES-EKF.

[m/s], compared to the best-performing Kalman filter stereo model of the S-MSCKF.

Since the maximum achieved velocity of the EuRoC MAV is nearly 2.3 [m/s], the quantitative results in Table 4.3 further support this conclusion, where our ES-EKF scores the best performance compared to the other state-of-the-art methods. In-depth reasoning for this degraded performance at high speeds (more than 5 [m/s]) can be clarified based on the hardware characteristics of the MAV sensors' properties, such as the data rate, latency, and noise effects at high speeds. Our optimization-based (PGO) initialization outperforms all other optimization- or filtering-based methods with high-rate visual-inertial sensors.

An insightful overview of the velocity profiles estimated by our ES-EKF is represented in Figure 4.8. The main conclusion is that the estimated velocity profile during the planar motion of the MAV in the X-Z plane optimally fits the upper and lower bounds of the top speed for each sequence. Towards an in-depth investigation to understand the high perturbations in the estimated velocity when approaching the maximum limit, we plot the velocity error states in the ES-EKF showing a high error at the instances when approaching top speeds due to strong vibrations in the MAV structure affecting the IMU readings.

The high estimation accuracy of our ES-EKF model compared to GPS readings and the PGO optimization-based initialization process is further verified by Y-axis trajectory estimation in Figure 4.7. The maximum estimated altitude for all sequences by the ES-EKF is nearly 60 [m], whereas both the GPS readings and the initialization optimizer estimate a maximum altitude of nearly 100 [m]. To physically validate which is a more accurate altitude estimation, we took snippets of the scene at a time instance in the exact halfway of all trajectories as shown in Figure 4.1. We can observe that the MAV is nearly on the same level as the roof of a commercial aircraft hangar, which is in the range of 30 [m] to 66 [m]. This observation validates the high estimation accuracy of the altitude using our ES-EKF.

4.4.4 Real-time Performance Analysis

The filter-based approaches are more advantageous for real-time onboard applications because they use the CPU more efficiently than monocular and stereo optimization-based methods. Due to its computationally intensive front-end pipeline for both temporal and stereo matching, OKVIS uses more CPU than VINS-Mono. Additionally, OKVIS's back-end operates at a speed that is much faster than the set 10 [Hz] rate of VINS-Mono. Around 90% of the work in our back-end, ES-EKF, is brought on by the front-end, which includes ORB feature detection, KLT-based tracking, and matching. At 200 [Hz], the filter uses around 10% of a core. Our suggested technique offers the maximum estimation frequency, which provides the optimal balance between precision and computing cost.

Figure 4.9 contrasts how much CPU time various VIO solutions used on the EuRoC benchmark and the Fast Flight dataset. Since V2-03 has considerable scale drift with S-IEKF and S-UKF-LG techniques and hence has significantly worse accuracy when compared to other methods, the CPU consumption of V2-03 is excluded from the comparison. According to the testing, the ES-EKF achieves the lowest CPU consumption while retaining a similar level of accuracy in comparison to other methods. We notice that the proposed method puts more computing work into the image

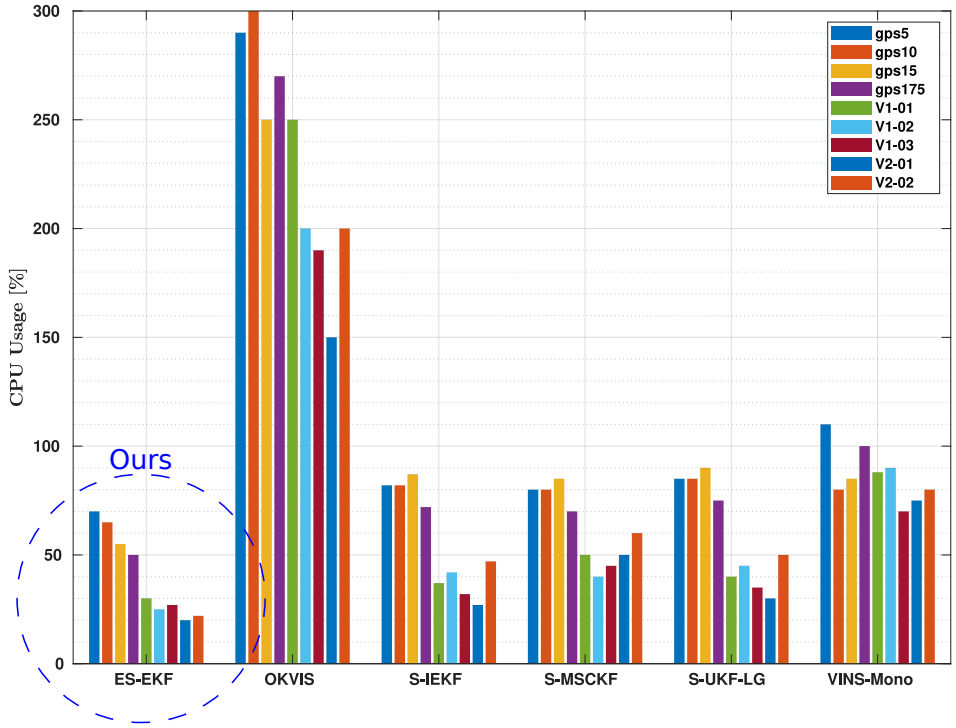


Figure 4.9: CPU usage as a real-time performance analysis indicator.

processing front-end than the tests using the EuRoC dataset. Higher imaging frequency and resolution are one explanation, while Fast Flight results in a shorter feature lifetime, necessitating frequent new feature identification, is another reason.

4.5 Observability Analysis

The EKF-based VIO for 6-DOF motion estimate contains four unobservable states corresponding to the global position and rotation around the gravity axis, or yaw angle, as demonstrated in [139]. A simple EKF VIO implementation will gather false information about yaw. The different processes and measurement's linearizing point causes this unobservability. To ensure that the uncertainty of the current camera states in the state vector is not impacted by the uncertainty of the current IMU state during the propagation step, in our implementation, camera poses in the state vector can be represented with respect to its inertial frame (v) instead of the latest IMU frame. Besides the efficient gyroscope RK4 integration scheme during the initialization process, our ES-EKF implementation minimizes the effect of the unobservable modes of the basic EKF. Figure 4.10 shows the IMU intrinsics, IMU-CAM extrinsic parameters, and odometry scale ES-EKF states plotted for sample EuRoC and Fast Flight sequences.

The main observation from Figure 4.10, is that when the motion of the MAV is smooth with no abrupt rotations and translations, our optimization-based initialization estimates an optimal metric-scaled trajectory with $\lambda = 1$. Moreover, we also observe that when the IMU-camera setup is not accurately calibrated, the ES-EKF can optimally

align the sensor setup in a robust online calibration process. Furthermore, the estimated IMU biases using our ES-EKF model are accurate and in a sensible range. One crucial observation is the estimated attitude visual drift of the visual sensor and the detection of consistent drift patterns based on the MAV speed (Fast Flight sequences) and abrupt motions (EuRoC sequences). These observations validate the contribution of the ES-EKF to the sustainability of the proposed method to achieve a resilient system that observes all the state vector parameters besides all the 6-DoF of the MAV trajectory. Finally, after the initial trajectory optimization, the filtering process is indispensable to estimate the false camera poses during long-term navigation caused by the visual attitude drifts.

4.6 Conclusion

Our work aimed to provide an accurate and computationally inexpensive localization solution for Micro Aerial Vehicles (MAVs) during long-term navigation in large-scale environments. To achieve this goal, we developed a loosely-coupled IMU/GPS-Camera fusion framework with a pose failure detection methodology. Furthermore, we proposed a novel decoupled optimization- and filtering-based sensor fusion technique that offers superior estimation accuracy and minimal system complexity compared to existing methods in the literature. We conducted extensive experiments using real-world indoor and outdoor settings for MAV localization studies to validate and test the findings of our proposed method.

We began our evaluation by examining the vision-based black-box pose estimation accuracy in a controlled laboratory Vicon room of the EuRoC benchmark. The results confirmed the system's reliance on monocular vision and its ability to perform accurately in such settings. Subsequent experiments on EuRoC and Fast Flight sequences demonstrated remarkable accuracy in trajectory estimation studies, further strengthening the effectiveness of our approach. Additionally, we assessed the proposed scheme in terms of computational complexity, measured by CPU usage. Our monocular-vision optimization/filtering solution consistently outperformed all competing techniques, showcasing its efficiency.

These conclusions emphasize our work's significant contributions toward providing a reliable (fast and accurate) sensor fusion solution for challenging and large-scale environments. This paves the way for enhancing the performance and robustness of MAVs in various applications, including surveillance, search and rescue, and environmental monitoring.

Looking forward, it will be essential to address situations where GPS sensor constraints, such as multipath effects on the optimizer, impact the performance of the proposed solution. This will ensure its robustness and applicability in even more challenging environments. Lastly, further generalizing the optimization problem will be necessary to extend the algorithm's pose estimation capability to include multiple vision sensors, such as stereo RGB. By doing so, the versatility and adaptability of our solution will be enhanced, making it suitable for a wider range of autonomous navigation tasks.

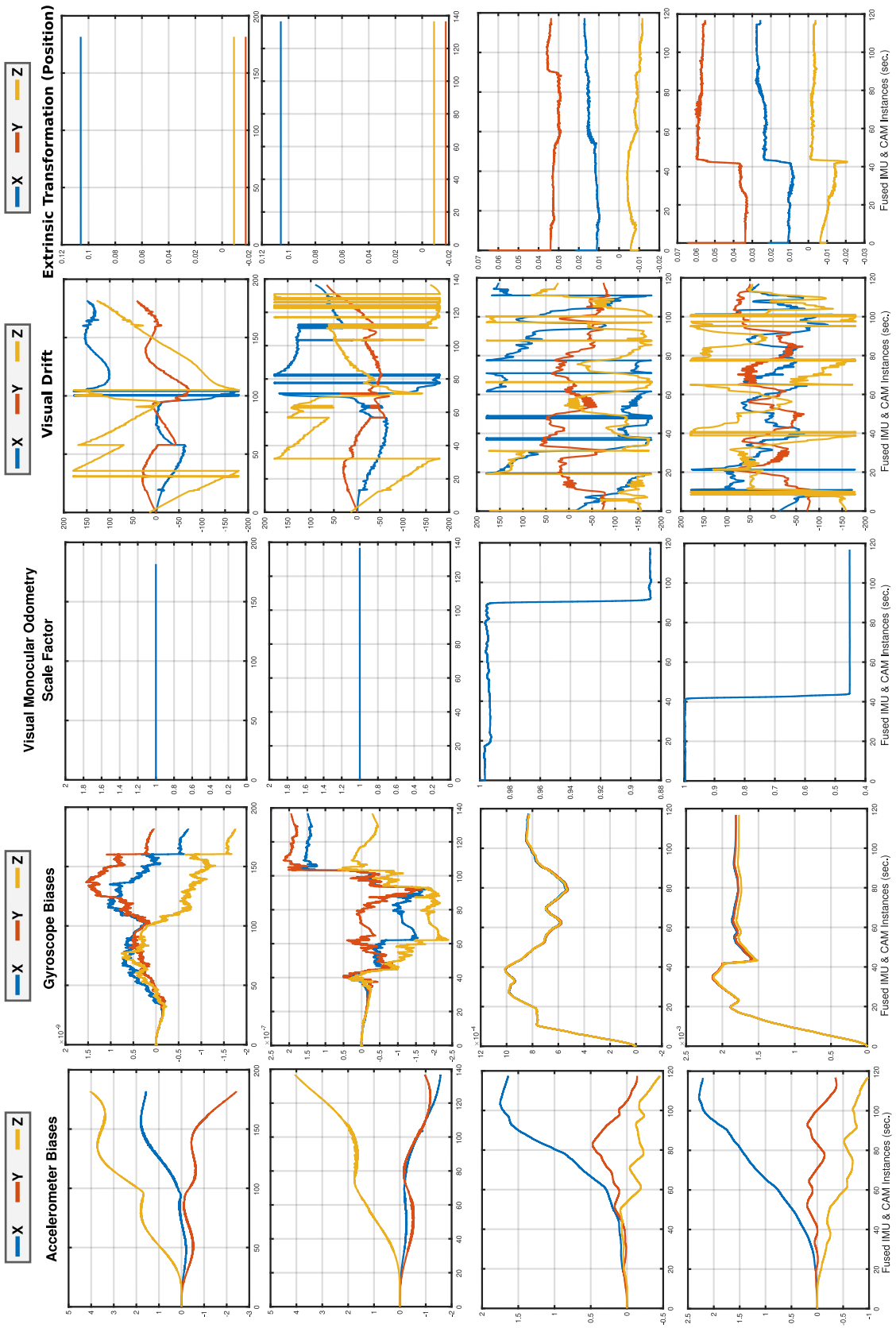


Figure 4.10: Our ES-EKF estimated states. Columns from left to right: IMU (accelerometer/gyroscope) biases b_a, b_ω, odometry scale factor λ, visual drift orientations q_v^w, and IMU-camera translation online calibration p_i^c. Rows 1,2 for sample FAST Flight sequences (gps5,gps10) and rows 3,4 for sample EuRoC sequences (V2-02, V2-03), respectively.

5 Hybrid Visual Odometry

Abstract

In this chapter, the DH-PTAM system is introduced as a solution for achieving robust parallel tracking and mapping in dynamic environments, utilizing stereo images and event streams. The system takes advantage of the strengths of heterogeneous multi-modal visual sensors and incorporates deep learning-based feature extraction and description techniques to enhance its robustness. Through CPU-/GPU-based experiments, it is demonstrated that the DH-PTAM system outperforms existing visual-inertial SLAM methods, particularly in challenging scenarios characterized by fast motion, High Dynamic Range (HDR), and occlusions. This showcases the system's ability to handle difficult conditions and produce superior results. The proposed system not only offers scalability and accuracy in 3D reconstruction and pose estimation but also provides a research-based Python API, which is publicly available on GitHub. This allows for further research and development, encouraging collaboration and innovation in the field.

"As far as the laws of mathematics refer to reality, they are not certain, and as far as they are certain, they do not refer to reality."

Albert Einstein

5.1 Introduction

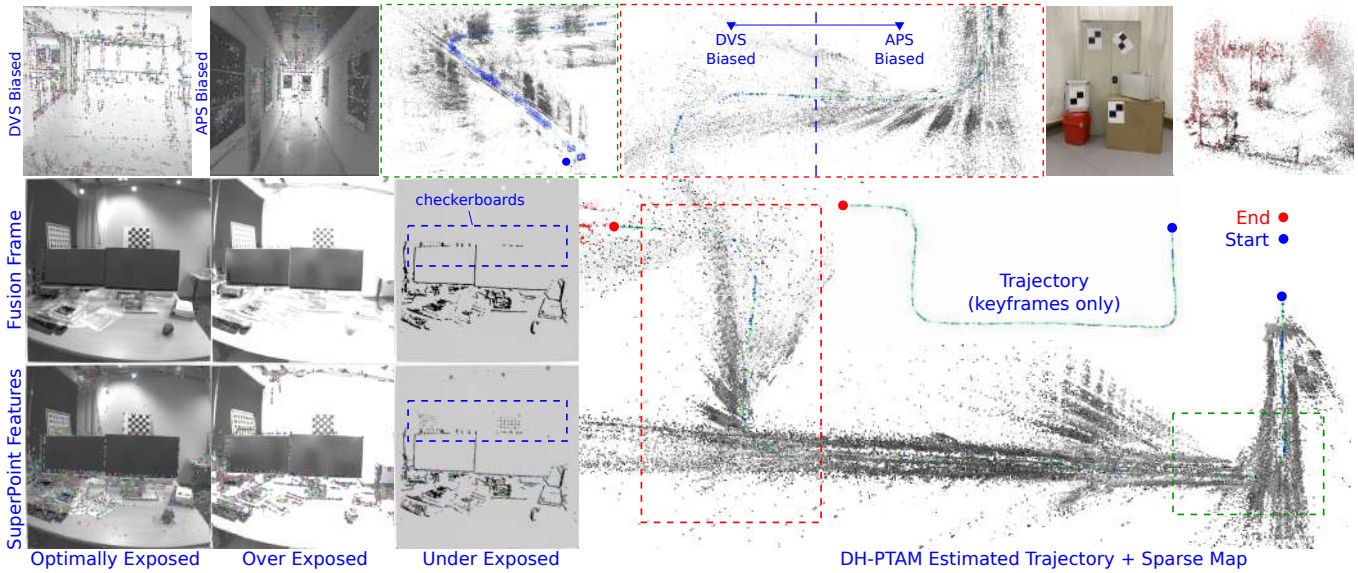


Figure 5.1: Top & Bottom (right): snippets of experiments on school-scooter and corner-slow sequences from the VECtor dataset that show the estimated trajectory with the constructed scene map (green dotted rectangle). Red dotted rectangle highlights an HDR use-case where DH-PTAM estimates the trajectory continuously based on the two fusion modes (Dynamic Vision Sensor (DVS) or Active Pixel Sensor (APS) biased). Bottom (left): snippets of an experiment on a small-scale (mocap-desk2) sequence from the TUM-VIE dataset that show the capability of the proposed events-frames fusion method to maintain and track features in dimmed and bright scenes where grayscale-only frames fail. APS: denotes the standard camera's global shutter frames.

Sensor fusion [140] combines data from multiple sensors to improve a system's accuracy, reliability, and robustness. It can also reduce computational costs by eliminating the need for redundant sensor data. Different types of sensors can be fused, such as cameras, lidars, radars, and ultrasonics. The algorithm used for fusion can vary, and it typically requires online calibration to ensure accurate and consistent data.

Visual Odometry (VO) is a method that utilizes sensor fusion to estimate the motion of a camera by analyzing the changes in visual features between consecutive frames. Still, it has challenges, such as difficulties in feature matching when the scene has little texture, the need for a robust feature detector and descriptor, and the problems of scale ambiguity and drift [141]. Scale ambiguity refers to the problem of determining the actual scale of the scene. In contrast, drift refers to the accumulation of errors over time that causes the estimated position to deviate from the true position. These challenges and limitations must be considered when applying frame-based visual odometry in practical applications [142].

An event camera [143], known as an asynchronous or dynamic vision sensor (DVS), operates on a fundamentally different concept than traditional frame-based cameras. Instead of capturing frames at a constant rate, event cameras output a stream of "events" that indicate the brightness changes in each pixel. This allows event cameras to operate at high speed, in very low-light conditions, and more resistant to motion blur [144]. The event-based nature

of these cameras also makes them highly suitable for tasks that involve fast-moving objects or scenes with high dynamic range. These characteristics make them an excellent complementary sensor to frame-based visual odometry in adverse conditions such as fast motion, high dynamic range, and low-light environments, where traditional cameras may struggle.

Deep learning-based features are more robust than traditional methods [145, 146], as they can learn from large amounts of data and generalize well to unseen data (for example, the checkerboard features in Figure 5.1). They are also more invariant to changes in viewpoint and lighting, making them suitable for real-world applications. Recently, pre-trained models have been widely adopted in computer vision and have achieved state-of-the-art performance in object detection, semantic segmentation, and image classification tasks.

Overall, a deep hybrid stereo events-frames parallel tracking and mapping system can significantly improve simultaneous localization and mapping accuracy and robustness in dynamic environments. This system combines the advantages of stereo standard and event cameras, which can capture visual information at high temporal resolution. The use of deep learning techniques in this system allows for the extraction of robust features from the stereo hybrid image and event frames, which can improve the accuracy of the feature-matching process and the estimation of the camera pose. Towards a robust metric-scaled tracking and mapping system that performs efficiently in adverse conditions, we contribute with the following:

- We propose an end-to-end parallel tracking and mapping (PTAM) approach based on a novel spatio-temporal synchronization of stereo visual frames with event streams (see Fig. 5.1).

- We propose a simple mid-level feature loop-closure algorithm for prompt SLAM behavior based on a learning-based feature description method to maximize robustness.

- DH-PTAM's effectiveness is evaluated in both stereo event-aided and image-based visual SLAM modes, achieving improved accuracy when incorporating event information, shown in an ablation study on the CPU versus the GPU of a consumer-grade laptop.

This chapter is organized as follows: Section 5.2 gives a brief overview of the state-of-the-art SLAM methods. Section 3.3 provides an in-detail overview of the proposed method and offers insights into the novel parts of the algorithm. Section 2.4 comprehensively evaluates the algorithm on the most recent VECtor [147] and TUM-VIE [148] benchmarks, along with defining the limitations. Section 3.5 summarizes the experiments' main observations, the proposed method's behavioral aspects, and the start points for future works.

5.2 Related Work

5.2.1 Conventional visual-SLAM

Simultaneous Localization and Mapping (SLAM) problem has been widely studied in the literature [149], and various techniques have been proposed to solve it. Deep learning has also been applied to SLAM [150] in recent years. Learning-based features extraction and description [145, 146] have been used to improve the SLAM robustness.

One of the most popular SLAM techniques is the filter-based SLAM using an extended Kalman filter (EKF) [140], or a particle filter [151]. These filters use probabilistic frameworks to estimate the robot's pose and map. They can handle non-linearities and uncertainties in the system, making them useful for large-scale and highly dynamic environments. Filter-based SLAM has been widely used in applications [152] such as mobile robots, UAVs, and autonomous vehicles.

Another important class of SLAM is graph-based SLAM [153], which uses a factor graph data structure to represent the robot's poses and the map. Graph-based SLAM requires Sparse Bundle Adjustment (SBA), which uses a non-linear least squares optimization to estimate the robot's poses and a graph to represent the map. These methods are robust to changes in lighting and viewpoint, making them well-suited for real-world applications. Some popular graph-based SLAM methods include ORB-SLAM [43], Basalt [69], and VINS-Fusion [112].

Loop-closure detection is a fundamental approach to minimize drifts in visual-SLAM, as it allows a system to recognize when it has returned to a previously visited location. Two common approaches to loop-closure detection are mid-level features [154] and bag-of-words [155] representations. Mid-level features are more abstract than low-level features, such as edges and corners, but are not as high-level as object recognition. Deep learning descriptors [156] can be considered mid-level features as they can extract higher-level information from raw data compared to low-level features, such as pixel values, but are not as high-level as features directly related to the task at hand, such as object labels.

5.2.2 Event-aided visual-SLAM

Event-based VO is an emerging form of localization solution that uses event-based cameras to generate measurements of the environment. While traditional SLAM is limited by the number of frames sampled, event-based SLAM provides high temporal resolution by generating an abundance of measurements, allowing for improved localized 3D and 6D pose estimation. Indirect methods, like frame-based approaches, extract keypoints from the input data in the front-end before passing them to the back-end. On the other hand, direct methods try to process all available events without any intermediate filtering. Table 5.1 compares the latest event-based and event-aided VO solutions concerning the sensor setup, events pre-processing layer (EPL), direct or indirect event processing, and the loop-closure capability to minimize visual drifts.

Table 5.1: Direct and Indirect (D/I) Visual Odometry methods based (B) on events and/or aided (A) by events

Method	B/A	D/I	EPL[a]	LC[b]	More Information
[157]	B	D	LIR	×	3 EKFs + Image reconst.
[158]	B	D	EI	×	Monocular (PTAM)
[46]	B	D	TS	×	Stereo (PTAM)
[159]	A	I	×	×	Event-aided Tracking
[160]	A	I	MEF	×	Mono + IMU (front-end)
[143]	A	D	EGM	×	Monocular Odometry
Proposed	A	D	E3CT	✓	Stereo (PTAM) + DL[c]

[a] denotes an Event Pre-processing Layer. [b] denotes Loop-Closure capability. [c] denotes the only method incorporating Deep Learning-aided features.

Event-aided systems leverage the high-quality representations that events can produce after processing, especially in dynamic and dimmed environments where standard camera frames fail. Some of the well-known event representations are event image (EI) [158], event frame [161], event count image [162], voxel grid [163], Time Surfaces (TS) [164], Event Spike Tensor (EST) [21], and recently Event 3-Channel Tensor (E3CT) [1]. Others [143] build the front-end on an Event Generation Model (EGM) [165] that generates a brightness increment model for the standard frame, which is fused with an event representation. Others [160] develop a front-end method to construct motion-compensated event frames (MEF) aided by a gyroscope and median scene depth information with no fusion between events and standard camera images. Towards a traditional frame reconstruction from events, [157] propose a Log Intensity Reconstruction (LIR), a model-based method, and [166] propose Spade-e2vid, a learning-based method.

5.3 Methodology

5.3.1 System Overview

Stereo Parallel Tracking and Mapping (Stereo PTAM) system is an extension of the original Mono PTAM, a real-time simultaneous localization and mapping (SLAM) algorithm for autonomous robots or devices. Stereo PTAM leverages the additional depth information from stereo cameras to improve the system's performance and robustness.

Figure 5.2 illustrates the main components and the process of DH-PTAM. The system establishes a global reference frame based on the camera position in the initial frame. A preliminary map is created by identifying and triangulating distinctive points in the first stereo image. For subsequent frames, the tracking thread calculates the 6D pose of each stereo frame by minimizing the discrepancy between the projected map points and their matches. The system chooses a subset of keyframes used in another thread to update the map at a slower pace.

Map points are derived from the stereo matches of each keyframe and added to the map. The mapping thread constantly improves the local discrepancy by adjusting all map points and stereo poses using Bundle Adjustment.

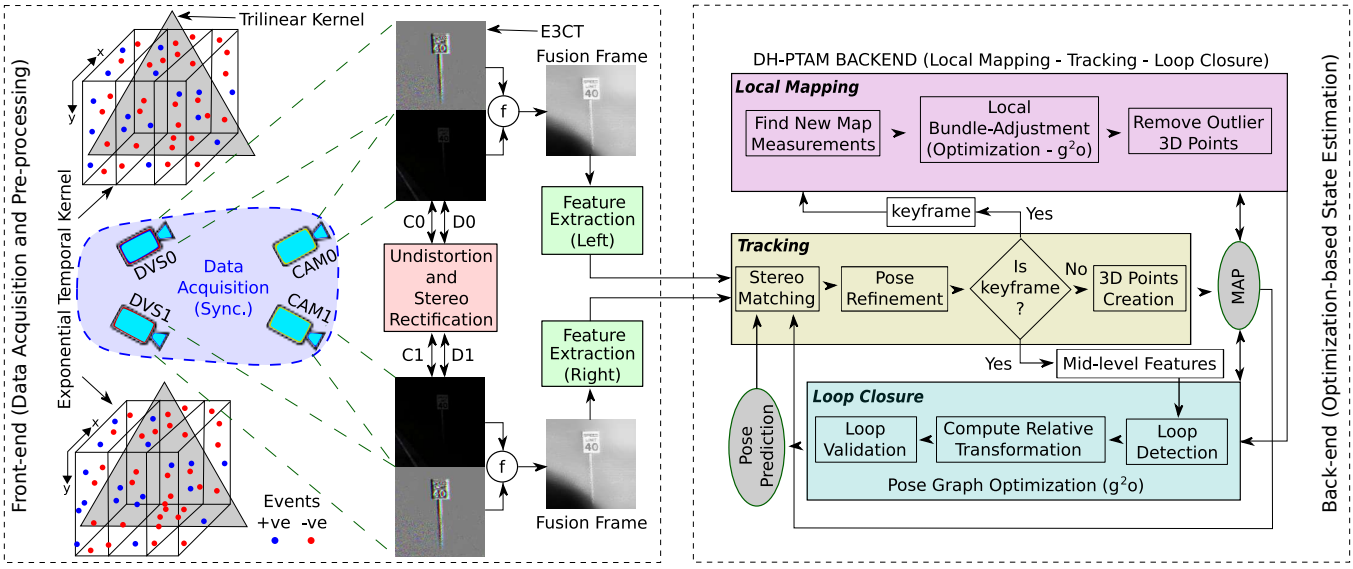


Figure 5.2: Block diagram of the proposed event-aided hybrid stereo odometry approach (DH-PTAM). DVS denotes "Dynamic Vision Sensor" (event camera).

A pose graph is utilized to preserve the global consistency of the map. The map is a shared resource among the tracking, mapping, and loop-closing threads. Point correspondences are actively searched between keyframes to strengthen the constraints of the pose graph optimization smoothing process.

Notations. The odometry state representation comprises the 3D points X_w^k and a 7-increment vector $\mu \in \mathfrak{se}(3)$, which is the current pose of the left fusion frame at time k:

$$\mu^k = [\delta x \ \delta y \ \delta z \ \delta q_x \ \delta q_y \ \delta q_z \ \delta q_w]^\top \ , \tag{5.1}$$

where $[\delta x \ \delta y \ \delta z]^\top$ is the incremental translation vector and $[\delta q_x \ \delta q_y \ \delta q_z \ \delta q_w]^\top$ is the incremental quaternion vector.

In Stereo PTAM, the state vector represents the system's current state, including both the camera pose and the 3D map points. Here's a breakdown of the components of the state vector:

- **Camera pose**: The camera pose is represented by a 6-DoF (Degrees of Freedom) transformation, which includes 3D position (X, Y, Z) and orientation (roll, pitch, yaw) of the camera in the world coordinate frame. The pose can also be expressed as a combination of rotation matrix (R) and translation vector (t) or as a quaternion (q) and translation vector (t).

- **3D Map points**: The 3D map points are the positions of the salient features observed by the stereo camera in the world coordinate frame. These points are used to create a map of the environment, which can be used for navigation and localization. Each map point is represented by its 3D coordinates (X, Y, Z).

The state vector in Stereo PTAM is typically represented as a concatenation of the camera pose and the 3D map points. For example, if there are N map points, the state vector would have a length of $6 + 3 * N$, where the first

six elements represent the camera pose, and the remaining $3 * N$ elements represent the 3D coordinates of the N map points.

The Stereo PTAM system uses the state vector in its localization and mapping processes, where it updates the camera pose and map points based on the new observations from the stereo camera. The state vector is essential for tracking the camera motion and maintaining an accurate and consistent map of the environment.

5.3.2 Spatio-temporal Synchronization

Spatio-temporal synchronization of events with global shutter frames is an essential aspect of vision systems that deal with dynamic scenes, particularly in applications such as robotics, autonomous vehicles, and sports analytics. Global shutter cameras capture the entire scene simultaneously, unlike rolling shutter cameras, which capture different parts of the scene at slightly different times. This feature allows for the precise alignment of spatial and temporal information, ensuring that all points in the scene are registered simultaneously. By leveraging global shutter frames, the spatio-temporal synchronization of events can be significantly improved, resulting in a more accurate representation of the scene's dynamics. This accurate representation is critical for reliable motion estimation, object tracking, and scene understanding in real-time applications. Furthermore, global shutter cameras reduce motion artifacts and distortions, which is common in rolling shutter cameras, ensuring that the captured images are more sensitive to the true nature of the observed events (see Figure 5.3).

Our spatio-temporal synchronization approach (see Figure 5.4) considers the general case of global shutter cameras where the exposure time $t_{exp_{0,1}}$ is known. We adopt the constant-time $\Delta t^k_{0,1}$ events accumulation window k approach in our spatio-temporal events-frames synchronization method.

As soon as stereo standard camera frames are received at timestamps $t_{C_{0,1}}$, we calculate the fusion frames timestamps assuming the hardware synchronization of stereo standard images and stereo event streams, using:

$$t_{f_{0,1}} = t_{C_{0,1}} + \frac{t_{exp_{0,1}}}{2} \ , \ \Delta t^k_{0,1} = t^k_{f_{0,1}} - t^{k-1}_{f_{0,1}} \ , \tag{5.2}$$

where t_{C_0} is the selected stereo keyframe timestamp.

5.3.2.1 The Event 3-Channel Tensor (E3CT)

Starting with the fundamental definition of an event. The event camera has a pixel array that triggers asynchronous firings called "events" with every luminosity (log brightness) change in the scene, according to the following formula:

$$L(x, y, t) - L(x, y, t - \Delta t) \geq pC, \tag{5.3}$$

where C is a contrast threshold, $p \in \{-1, +1\}$ is the polarity of a decreasing or increasing scene luminosity, Δt is

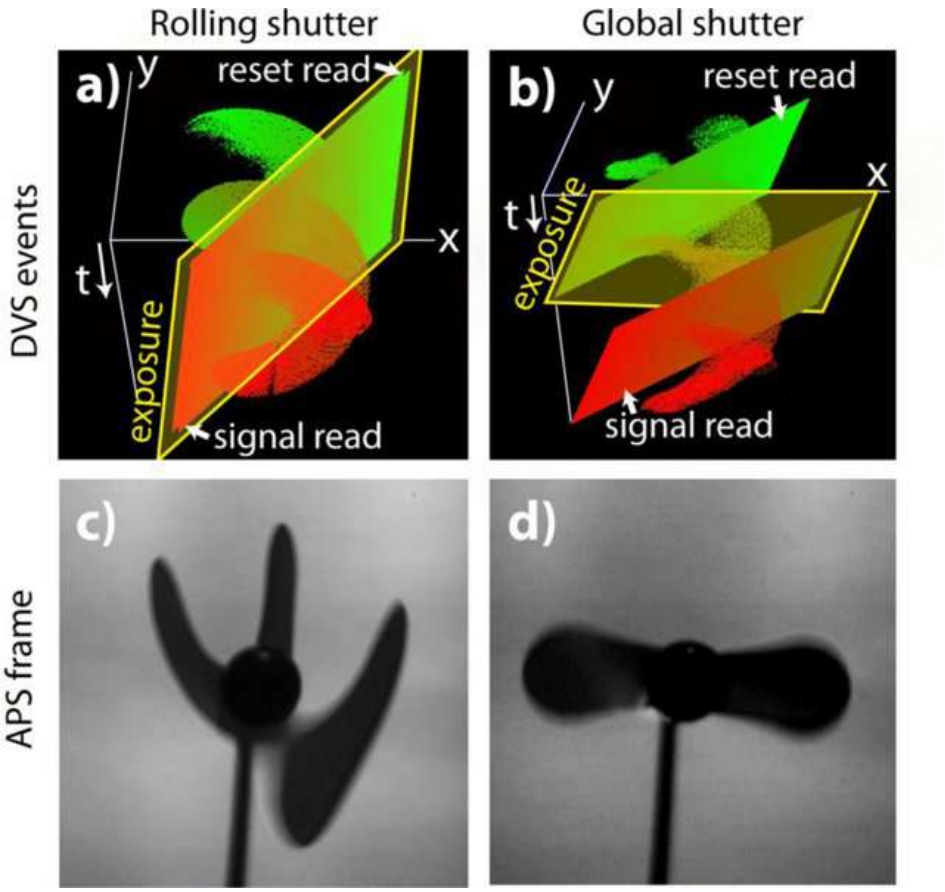


Figure 5.3: Comparison of rolling shutter and global shutter DAVIS readout: Visualization of the DVS events and APS frames generated by a 50 Hz rotating fan in space-time. Figures (a) and (b) display the data in space-time, with APS sample readouts appearing as slanted planes. DVS events and APS samples are represented as dots, with recent events in red and older ones in green. The exposure time is indicated by a yellow rectangle marked "exposure". Figure (c) presents the output of the rolling shutter readout, while (d) demonstrates the global shutter readout. The illustration figure is courtesy of [167].

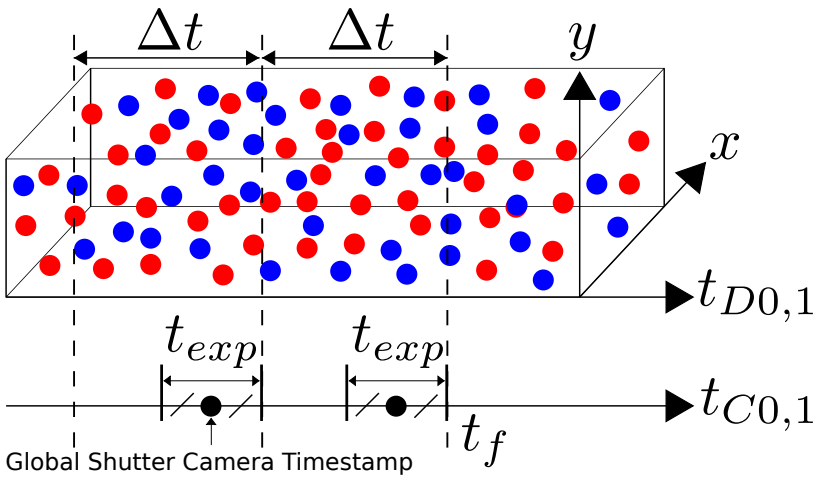


Figure 5.4: Spatio-temporal synchronization scheme. t_{exp} is the global shutter camera exposure time. Δt is the event representation (E3CT) volume accumulation time. t_f is the fusion frame calculated timestamp. $t_{D,C}$ are the DVS events, and standard camera frames timestamps, respectively.

the time-lapse between two event firings at $(x, y)^\top$. For a temporal interval $\Delta\tau$, the event camera triggers an array of 4D-tuples:

$$\mathcal{E} = \{e_k\}_{k=1}^{N} = \{(x_k, y_k, t_k, p_k)\}_{k=1}^{N}. \tag{5.4}$$

Owing to their asynchronous characteristics, events are depicted as a collection. To employ events in conjunction with convolutional neural networks or visual SLAM systems, it is essential to transform the event collection into a grid-like format. Consequently, we need to establish a mapping $\mathcal{M} : \mathcal{E} \to \mathcal{T}$ between the set $\mathcal{E}$ and a tensor $\mathcal{T}$. Ideally, this mapping should maintain the structure (spatiotemporal proximity) and the information contained within the events.

Towards a generalized mapping $\mathcal{M} : \mathcal{E} \to \mathcal{T}$. In the article of [21], the authors present an innovative approach to learning event representations from asynchronous event-based data. The event representation is derived from two distinct fields, the event field (5.5) (events of both polarities are represented as Dirac pulses in time) and the event-assigned measurement field (5.6) (events are grouped according to their polarity, normalized timestamp, or count). The event field captures the spatial and temporal characteristics of the events, while the event measurement field assigns specific measurements to these events. To represent the event membrane potential (5.7), the authors introduce the concept of kernels (alpha, exponential and trilinear voting), which are responsible for capturing the spatial and temporal dependencies of the events. By leveraging these kernels, the researchers construct the Event Spike Tensor (EST) (5.8), a novel generalized data structure that efficiently encodes the asynchronous event-based data in spatio-temporal bins. The EST enables extracting meaningful features from the event-based data, paving the way for end-to-end learning of representations in various event-driven applications. Figure 5.5 shows graphical illustrations for these concepts.

- Event Field (Spatio-temporal Dirac Pulses $\delta(x, y, t)$):

$$S_{\pm}(x, y, t) = \sum_{e_k \in \mathcal{E}_{\pm}} \delta(x - x_k, y - y_k)\delta(t - t_k). \tag{5.5}$$

- Event (Assigned) Measurement Field:

$$S_{\pm}(x, y, t) = \sum_{e_k \in \mathcal{E}_{\pm}} f_{\pm}(x_k, y_k, t_k)\delta(x - x_k, y - y_k)\delta(t - t_k),\ f_{\pm}(.) = \begin{cases} \pm 1 & \text{Event Polarity} \\ \frac{t - t_0}{\Delta t} & \text{Normalized Timestamp} \\ 1 & \text{Event Count} \end{cases} \tag{5.6}$$

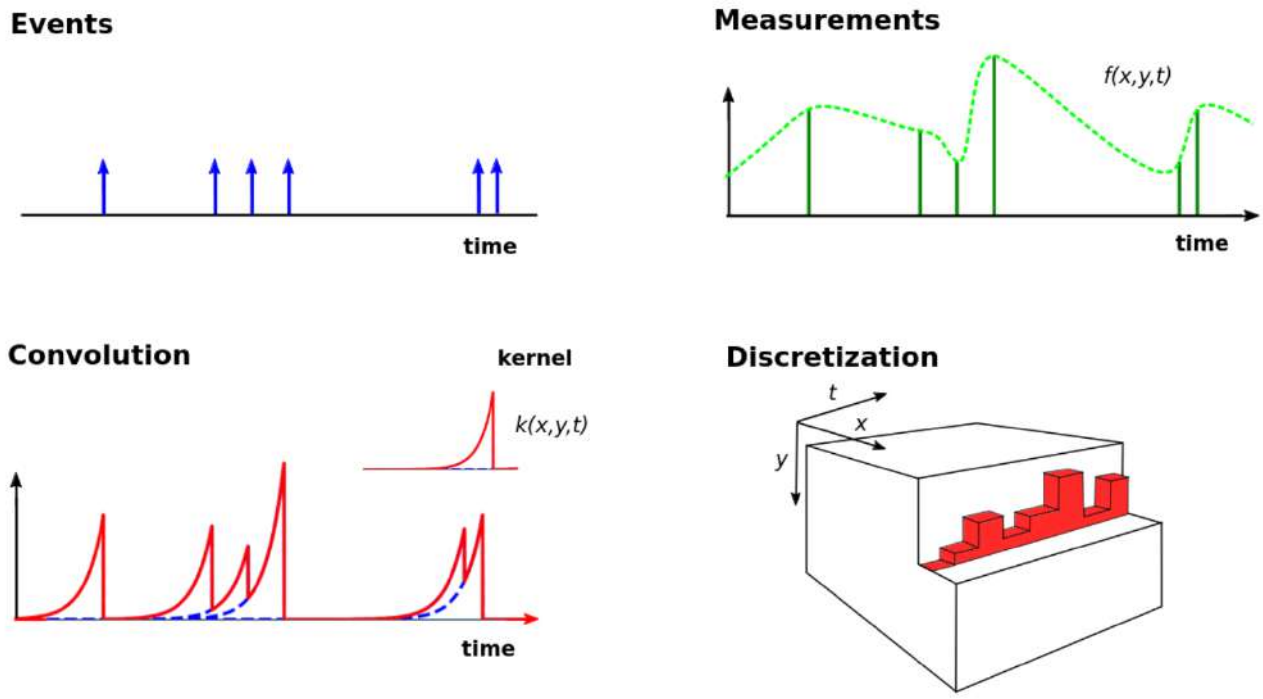


Figure 5.5: A synopsis of the recommended framework is provided. Each event is linked to a measurement (indicated in green) that undergoes convolution with a potentially learned kernel. The convoluted signal is then sampled on a uniform grid. Different representations can be generated by executing projections along the temporal axis or across polarities. Illustrative figures are courtesy of [21].

- Membrane Potential (Spatio-temporal and Voting Kernel Convolutions $k(x, y, t)$):

$$(k * S_{\pm}(x, y, t)) = \sum_{e_k \in \mathcal{E}_{\pm}} f_{\pm}(x_k, y_k, t_k) k(x - x_k, y - y_k, t - t_k),\ k(.) = \begin{cases} \delta(x, y) \frac{\alpha t}{\tau} \exp -\frac{t}{\tau} & \text{Alpha} \\ \delta(x, y) \frac{1}{\tau} \exp -\frac{t}{\tau} & \text{Exponential} \\ \delta(x, y) \max(0, |\frac{t}{\Delta t}|) & \text{Trilinear} \end{cases} \tag{5.7}$$

- Generalized Representation (Event Spike Tensor, EST):

$$S \pm [x_l, y_m, t_n] = (k * S_{\pm}(x_l, y_m, t_n)) = \sum_{e_k \in \mathcal{E}_{\pm}} f_{\pm}(x_k, y_k, t_k) k(x_l - x_k, y_m - y_k, t_n - t_k), \tag{5.8}$$

$$\text{Discretized Spatio-temporal Bins (n) on a Voxel Grid} \in \begin{cases} x_l \in \{0, 1, \cdots, W - 1\} \\ y_m \in \{0, 1, \cdots, H - 1\} \\ t_n \in \{t_0, t_0 + \Delta t, \cdots, t_0 + n\Delta t\} \end{cases} \tag{5.9}$$

Inspired by the generalized representation. We propose a novel DVS sensor pre-processing layer called the Event 3-Channel Tensor (E3CT) based on the Event Spike Tensor (EST) representation method [21]. The E3CT is a pre-processing layer that combines the benefits of both EST and Histograms of Averaged Time Surfaces (HATS)

[164] concepts. This allows its modeling to be simple with reliable time information. The E3CT concept is illustrated in Algorithm 4, where in line (19), an event volume $\mathcal{V}_0(x, y, t)$ is created in the form of a 4D tensor (n,2.c,h,w), then in line (20) we update every event's e(x,y,p,t) linearly weighted histogram. Where $t_i = \frac{t}{\delta}$ is the sequential number of the current event time, $c_i = ([\frac{t \times c}{\delta}] \mod c)$ the sequential number of the current event nano-time bin, $[.]_c$ is the closest nano-time bin number to the right and left, $t^* = c \times \frac{t_i - t_0}{\delta} - 0.5$ is the relative temporal distance to the center of the corresponding nano-time bin, $t_i = \frac{t}{\delta}$ is the sequential number of the current nano-time bin, t_0 is the initial event timestamp [nsec], $\delta = \frac{\Delta t}{n}$ is the nano-time bin interval [nsec].

Algorithm 4 Event 3-Channel Tensor (E3CT) Pre-processing Layer for Frame-based Systems

Input: Packets of Events Arrays $@f_{ep}\ Hz$
Output: E3CT (RGB Frame) $@f_{ec}\ Hz$

1: $hot\_pixels \Leftarrow Hot\ Pixel\ Array$ ▷ Figure 2.9
2: $n \Leftarrow 24$ ▷ #Temporal bins
3: $c \Leftarrow 3$ ▷ #Channels
4: $h \times w \Leftarrow 1024 \times 1024$ ▷ Frame dimensions
5: $\Delta t \Leftarrow 1e9$ (1 sec) ▷ Sampling duration [nsec]
6: $e_l \Leftarrow [t, x, y, p]$ ▷ Events List
7: **for each** $packet \in event\_packets$ **do**
8: **if** $\#packets < f_{ep}/f_{ec}$ **then**
9: Load events in the current packet
10: **if** $length(packet) > 2 \times h \times w$ **then**
11: Remove hot_pixels from packet
12: Add packet to e_l
13: **else**
14: Add packet to e_l
15: **end if**
16: $t_f \Leftarrow e_l[t][-1]$
17: packet += 1
18: **else**
19: $\mathcal{V}_0 \Leftarrow \mathcal{V}(e_l, n, 2 * c, \Delta t)$ ▷ Event Volume Construction
20: $\mathcal{V}_1 \Leftarrow \mathcal{V}_0(e_l, n, 2 * c, \Delta t)$ ▷ Alpha Temporal Exponential Kernel
21: $\mathcal{V}_2 \Leftarrow \mathcal{V}_1(t_i, c_i, y, x)$ += $\max(0, 1 - \|t_c^* - t^*\|)$ ▷ Trilinear Voting Kernel
22: $\mathcal{V}_{Total} = \sum_{n=0}^{24} \mathcal{V}_2(t_i, c_i, y, x)$
23: visualize ($\mathcal{V}_{Total}$) ▷ Synthetic 3-Channel RGB Frame Construction
24: **end if**
25: **end for**

The E3CT events pre-processing layer is adopted and modeled as two consecutive filtering kernel convolutions on the event volume $\mathcal{V}_0(x, y, t)$ of the temporal width of Δt^k. The first kernel to filter the time decaying events in the volume is the α-exponential time decay kernel and modeled as:

$$\mathcal{V}_1(x, y, t) \doteq \exp\left(-\alpha \left(\frac{\mathcal{V}_0(x, y, t) - \eta/2}{\eta/6}\right)^2\right), \tag{5.10}$$

where $\alpha = 0.5$ and the decay rate $\eta = 30$ [ms] for our model. Followed by a trilinear voting kernel to stack the events in the three channels tensor so that each event contributes to two consecutive channels depending on their location from a vertex of this trilinear kernel. An event near the top contributes a higher weight to the current channel and a

lower weight to the neighboring ones. These contribution weights of the three channels can represent a percentage of an R-G-B color map; hence, the E3CT can be considered a synthetic RGB frame of events. The trilinear voting kernel can be modeled as follows:

$$\mathcal{V}_2(x, y, t_i) \doteq \max\left(0,\, 1 - \left|\frac{\mathcal{V}_1(x, y, t_i)}{\delta t}\right|\right) \,, \tag{5.11}$$

where δt is the temporal bin i size as discussed in [21].

After applying the trilinear temporal voting kernel on the exponential-decay time surface, we stack the 3-channel tensor temporal bins together, resulting in a synthetically colored 2D frame called the Event 3-Channel Tensor (E3CT). In Figures 5.7,5.2, we can observe that the constructed synthetic colors are always consistent, meaning that the stereo left and right constructed E3CTs have identical colors for the same scene.

For a visualized illustration for the nano-time bins concept see Figure 5.6, we show an example of two positive events fall with the only two possibilities. The first falls exactly on the nano-time bin edge between the two nano-tbin n, n+1. This event will contribute equally to both first and middle channels of this E3CT. The second falls totally in the n-1 nano-tbin, so it will have a high contribution weight to the last channel and a low contribution weight to the middle and no contribution to the first channel. As, the number of nano-tbins increase the number of contribution slices will increase and the quality of the E3CT will improve.

In Figure 5.8, we represent six cases to compare DVS sensor event arrays preprocessed using the E3CT and E2VID methods without any post processing. These corner cases show the the effectiveness of the E3CT to construct frames that preserve the scene artifacts even in the harshest weather conditions. This is due to the high sampling duration (1 sec) set during the E3CT construction. Selecting high sampling duration along with a hot pixel filter can efficiently suppress the rain and fog events contributions to the E3CT as seen with `Cases 3-6`. Figure 5.7 gives an in-depth illustration to the quality of the constructed E3CTs on sample artifacts from IBISCape sequences.

The only case where the E2VID can evaluate event-based SLAM systems performance in trajectory estimations is clear weather with a low dynamic scene level as in the `Clear 1` sequence. A rapid change in the scene can cause an instantaneous map loss that affects the whole trajectory estimation even if the weather is clear, as in the `Clear 2` sequence. IBISCape ESVI sequences are provided in raw `.npz` (NumPy arrays) and bag formats for the stereo-DVS events, along with `timestamps.csv` file includes the start and end timestamps for every time surface. Besides, the E2VID grayscale frames reconstruction results and the locations of the hot pixels for the stereo-DVS are also provided.

5.3.3 Events-Frames Hybridization Front-end

One of the main advantages of our front-end fusion modeling is that it does not rely on any online probabilistic photo-metric matching or alignment approach using filters or cost functions and considers all events polarities $p \in$

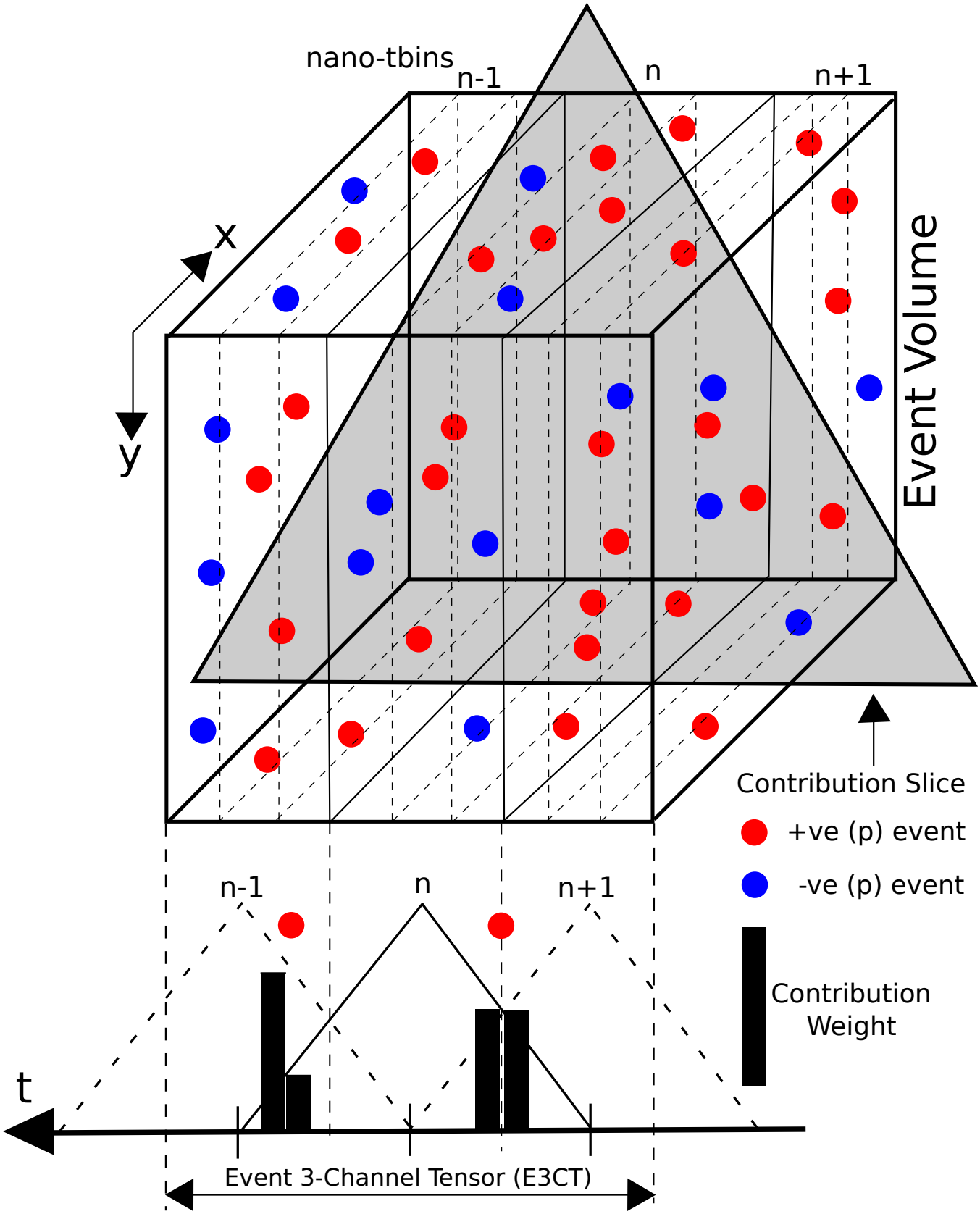


Figure 5.6: Graphical illustration of E3CT nano-time bins and events contribution to each channel. Nano-tbins are replaced with micro-tbins for real-world DVS sensors. The contribution slice is mathematically a trilinear voting kernel. The event volume is mathematically an exponentially decaying time surface of events with both polarities.

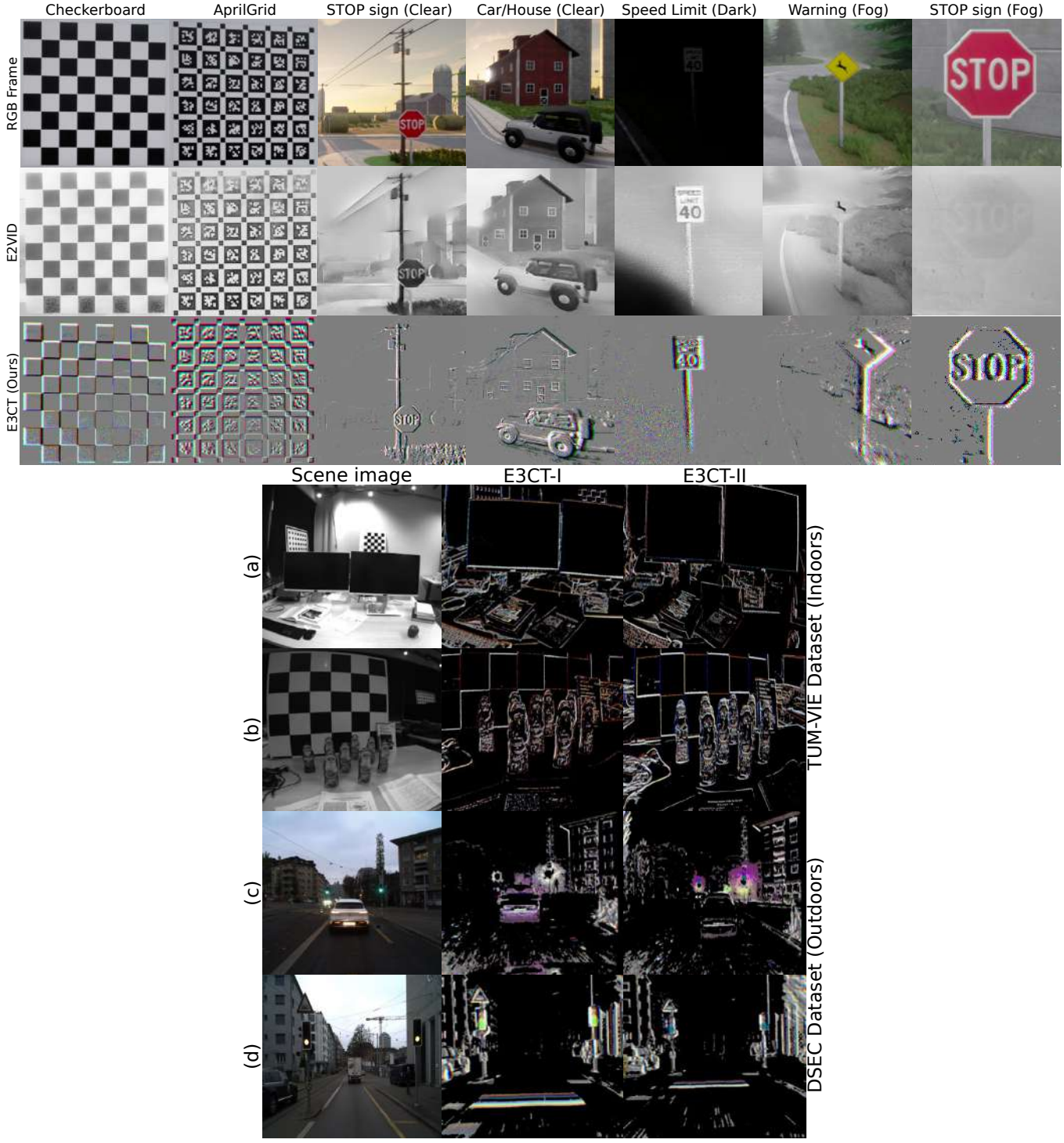


Figure 5.7: E3CT qualitative analysis in simulated (top) IBISCape [1] and real-world (bottom) TUM-VIE and DSEC [37, 13] scenes. **Top**: Event 3-Channel Tensor precision testing on multiple IBISCape artifacts compared to RGB and E2VID frames. **Bottom**: (a) Sequence: TUM-VIE Dataset (mocap-desk2), edge events are color-encoded with bright (red/blue) gradient pixels in good light. (b) Sequence: TUM-VIE Dataset (mocap-6dof), edge events are color-encoded with dark (red/blue) gradient pixels in dimmed light. (c) Sequence: DSEC Dataset (Zurich_city_00_a), the green traffic lights (dark - far) and the car rear lights triggering color-encoded events. (d) Sequence: DSEC Dataset (Zurich_city_04_a), the green traffic lights (bright – near / dark - far) are clearer than the RGB frame.

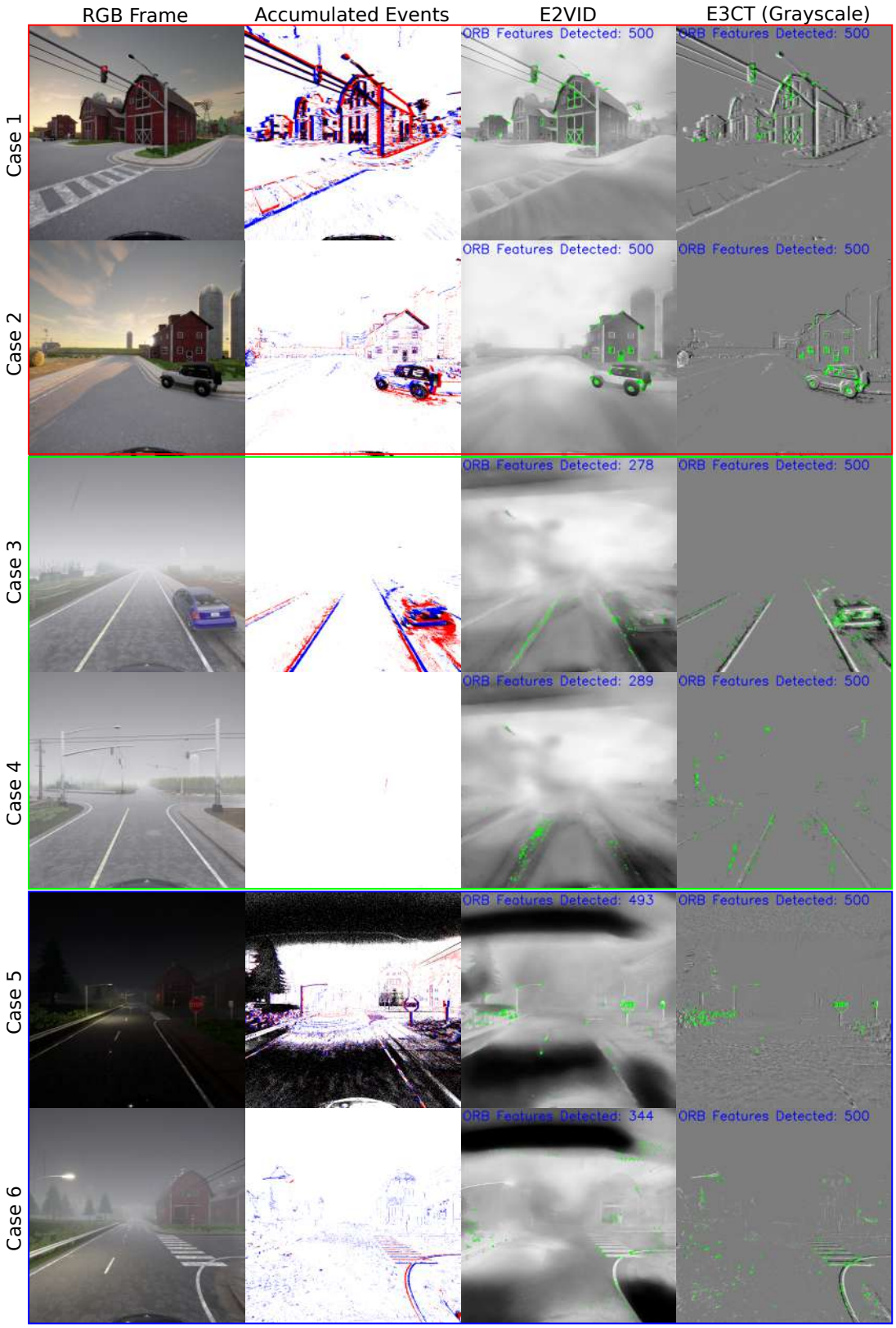


Figure 5.8: The effect of adverse weather conditions on DVS events and ORB feature extraction. E3CT preserves both the high dynamic range property with the pixel temporal information of the DVS sensor and the high quality with rich information (3 channels) of the RGB frames in all weather conditions in both static and dynamic scenes. Six cases are tested with an ascending difficulty from clear to adverse weather and static to the dynamic scene.

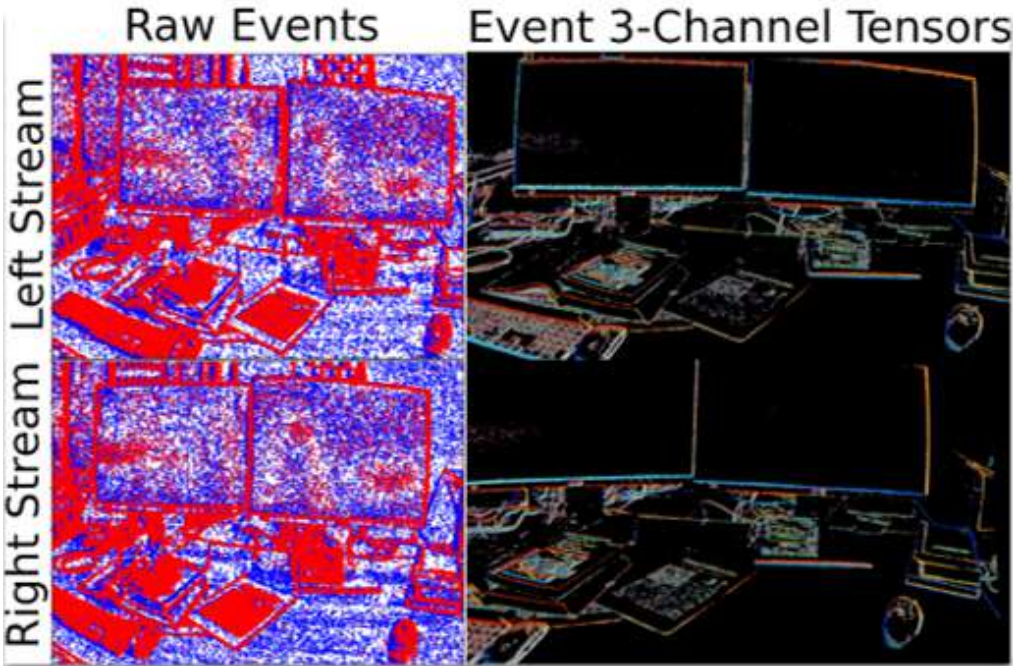


Figure 5.9: E3CT after the post-processing operations.

$\{+1, -1\}$. Hence, the computational load of our method lies mainly on the PTAM modules of the optimization-based back-end. We employ our novel event pre-processing layer, the Event 3-Channel Tensor (E3CT), that is thoroughly discussed in Chapter 2.

Conventional frame-based post-processing operations can be applied to the constructed E3CTs, such as adaptive threshold, contrast stretch, color correction and balance, and denoising functions. Figure 5.9 shows the effect of the post-processing operations on the E3CT compared to a conventional event accumulation frame. We consider a fully calibrated stereo standard and event cameras stack as represented in Figure 5.10, so that the rigid-body transformations $\mathcal{T}_{cd_{0,1}} = [R_{cd_{0,1}}|t_{cd_{0,1}}]_{3\times4}$ and the cameras intrinsic parameters $\mathcal{K}_{c_{0,1}}, \mathcal{K}_{d_{0,1}}$ are known.

Given that the same post-processing operations are applied on the current stereo E3CT frames, the 2D-to-3D-to-2D consecutive inverse-forward projections of the pixels on the E3CT frames $P^h_{d_{0,1}}$ to the standard camera frames $P^h_{d\in c_{0,1}}$ can be performed as follows:

$$P^h_{d\in c_{0,1}} \approx \mathcal{K}_{c_{0,1}}\, \mathcal{T}_{cd_{0,1}}\, [(\mathcal{K}_{d_{0,1}})^{-1}\, P^h_{d_{0,1}} \quad 1]^\top + \delta P^h_{align}\ , \tag{5.12}$$

where $(.)^h$ denotes the pixel location in homogeneous coordinates. The term δP^h_{align} denotes the pixel location alignment correction factor for the standard and event frames (see Figure 5.11) so that the same 3D world point $X^h_{w_{0,1}}$ should correspond exactly to the pixel locations $P^h_{d\in c_{0,1}}$, $P^h_{c_{0,1}}$. This alignment term is observed to be constant for the same sensor rig with non-varying intrinsic and extrinsic parameters. δP^h_{align} value can be estimated using an offline optimization process only once on a selected number of frames (the more the accurate) with high confidence feature matches, and this value is given in Section 5.4 for both VECtor and TUM-VIE sequences.

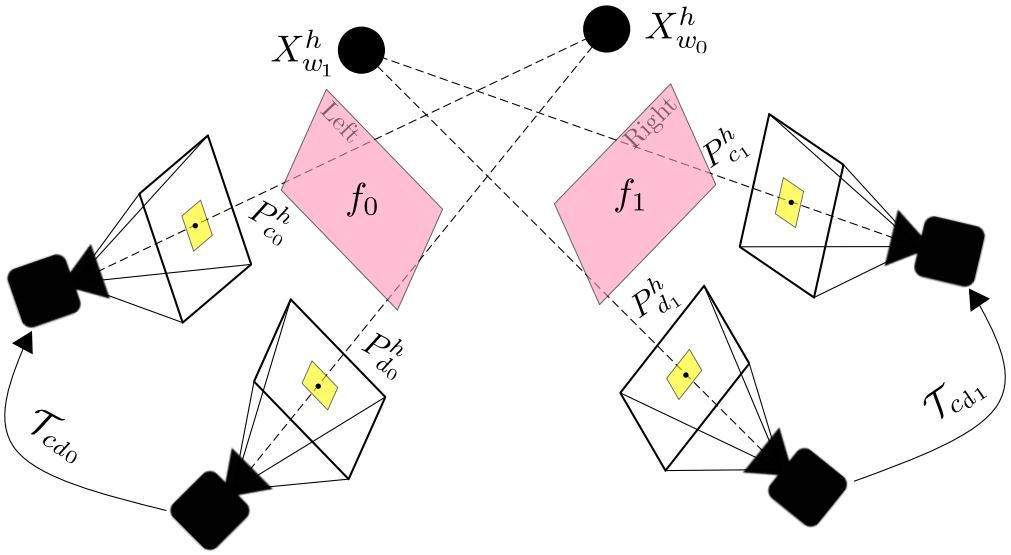


Figure 5.10: Geometry of the stereo hybrid event-standard cameras stack.

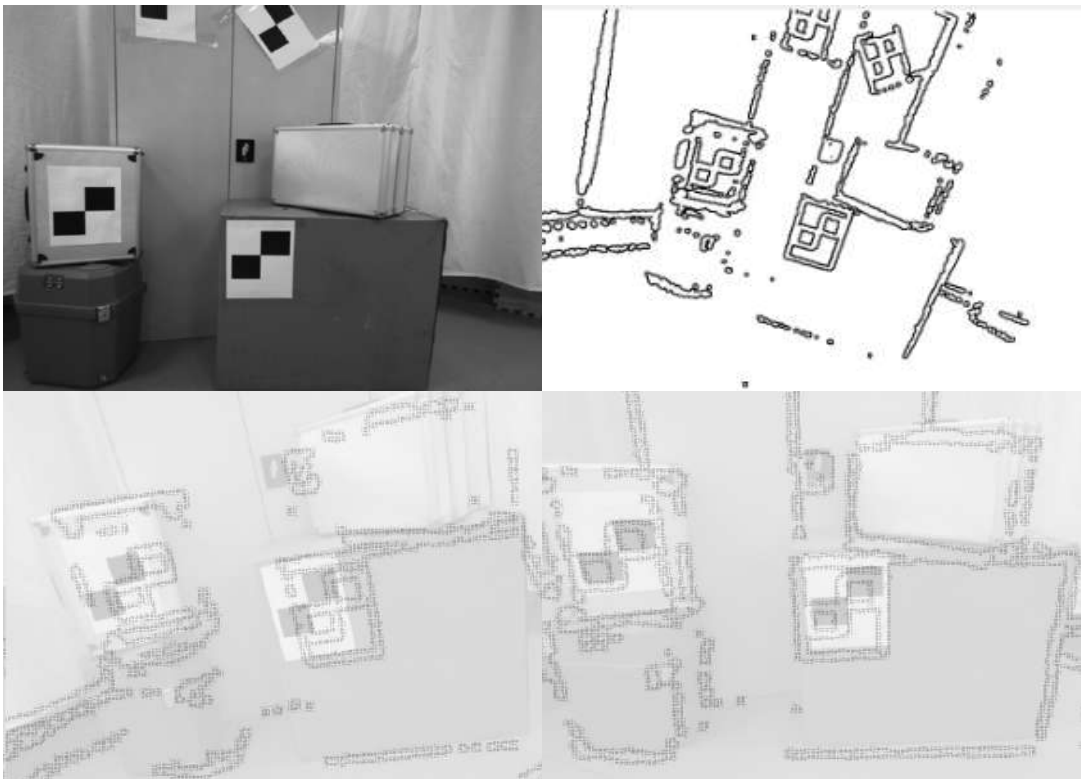


Figure 5.11: E3CT alignment with the standard camera frame. Top: standard camera frame (left) and E3CT post-processed frame (right). Bottom: E3CT-standard camera fusion frames before (left) and after (right) δP^h_{align} correction.

Finally, the fusion function (and frame) $f(.)$ performs a temporal cross-dissolve (linear blending) between both the left (D_0, C_0) and right (D_1, C_1) E3CTs and standard camera frames, respectively, and is formulated as:

$$f_{0,1}(C_{0,1}, D_{0,1}) \; = \; (1-\beta) * C_{0,1} + \beta * D_{0,1} \; , \tag{5.13}$$

where $\beta \in [0, 1]$ is the E3CT contribution weight in the current fusion frame. β value is dynamic and depends on the scene lighting and texture conditions. It should be set to high values $\beta = \max(\frac{\bar{C}_{0,1}}{C^{max}_{0,1}}, \, 1 - \frac{\bar{C}_{0,1}}{C^{max}_{0,1}})$ when the standard camera frame fails to detect features due to adverse conditions and low-textured scenes, and this is the DVS-biased fusion mode. For situations where standard camera frames can detect reliable scene features with good lighting and enough texture, the β value should be low $\beta = \min(\frac{\bar{C}_{0,1}}{C^{max}_{0,1}}, \, 1 - \frac{\bar{C}_{0,1}}{C^{max}_{0,1}})$ to reduce the amount of extracted features to maintain the back-end processing complexity and latency in reasonable ranges, and this is the APS-biased fusion mode.

Dynamic scenes with challenging and adverse conditions can easily trigger rapid switching between these two fusion modes during estimation (see Figure 5.12). This causes a critical problem during the feature tracking process using conventional low-level feature detectors, such as ORB, SIFT, SURF, BRIEF, and FAST. Accordingly, applying mid-level feature detectors that depend mainly on learning-based architectures could solve this fusion frame modes alternation problem. Hence, we employ the learning-based feature extractors and descriptors [145, 146] for their high robustness and feature detection speed.

5.3.4 Optimization-based Back-end

Inspired by the first work of the traditional S-PTAM system, all the optimization Jacobians mentioned in this section can be found with detailed proofs in [168]. All objective functions are minimized with the Levenberg-Marquardt algorithm implemented in the g^2o optimization library. We employ the Huber loss function for outliers rejection $\rho(.)$.

5.3.4.1 System bootstrapping

The bootstrapping process in a Stereo Parallel Tracking and Mapping (Stereo PTAM) system refers to the initialization phase. The system creates an initial map and estimates the initial camera pose based on the first few frames captured by the stereo camera. The bootstrapping process is essential for establishing a starting point for subsequent tracking and mapping updates. Here is a high-level overview of the bootstrapping process in Stereo PTAM:

1. **Capture stereo frames**: The stereo setup captures images (left and right) from the environment. These images are used to extract features and compute depth information.

2. **Feature extraction**: Features are detected and extracted from the left and right images using feature extraction algorithms, such as Scale-Invariant Feature Transform (SIFT), Oriented FAST and Rotated BRIEF (ORB), or

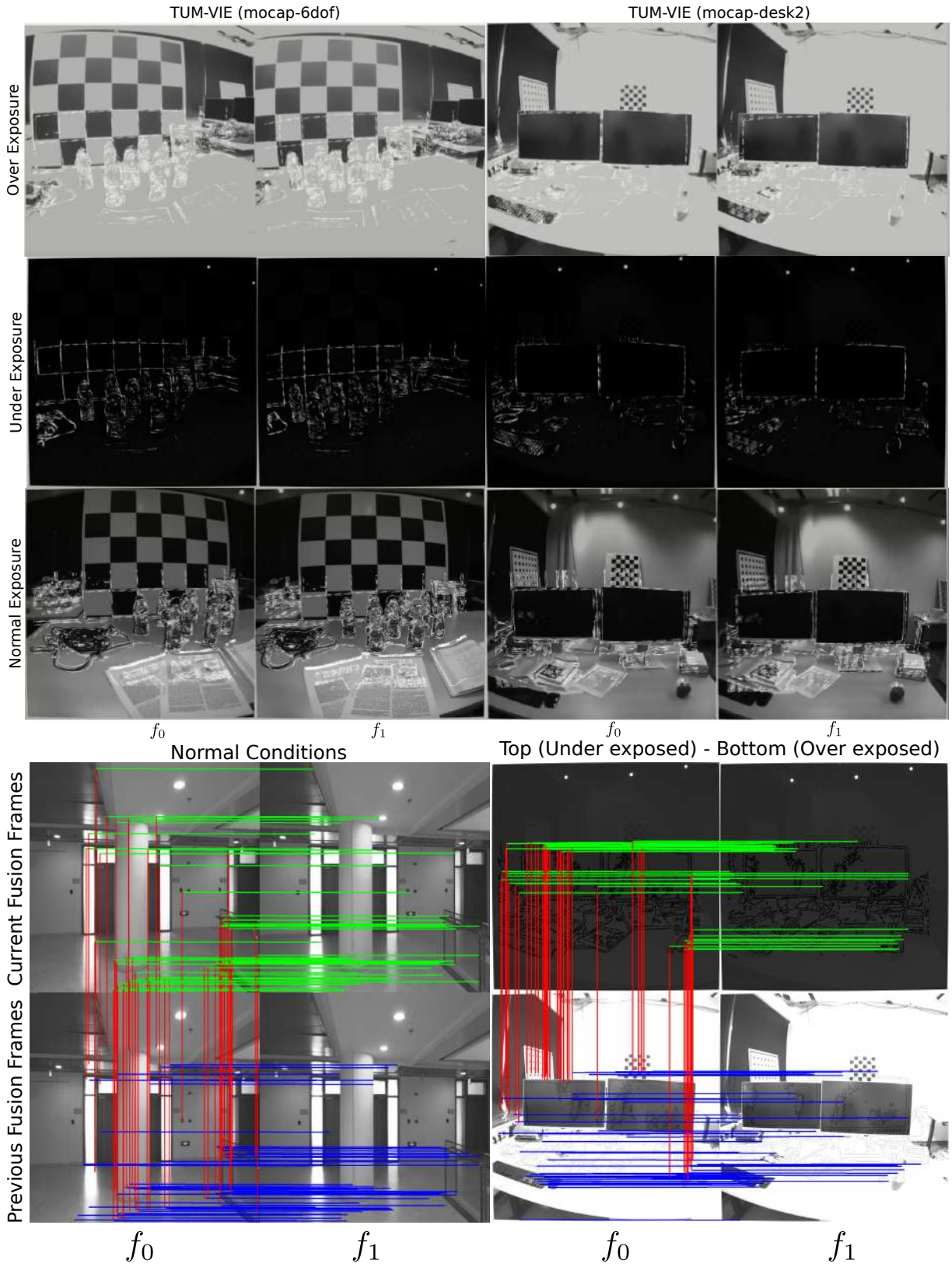


Figure 5.12: Spatio-temporal matching for SuperPoint features on two consecutive stereo fusion frames. A random batch of 50 matches is selected as a sample. The dynamic value of β opens new horizons for introducing a continuous-feature concept. This continuous-feature has a high-quality traceability as evident with the learning-based SuperPoint detector in these challenging situations.

learning-based methods. These features are matched between the left and right images to establish correspondences.

3. **Compute depth information**: By using the matched features and the known baseline distance between the two cameras, depth information can be calculated through triangulation. This process results in 3D coordinates for the matched features, which serve as the initial map points.

4. **Estimate initial camera pose**: The initial camera pose can be estimated by solving a Perspective-n-Point (PnP) problem using the 3D map points and their corresponding 2D image points in the left image. The PnP problem aims to find the camera pose that minimizes the re-projection error between the observed 2D image points and the 3D map points projected onto the image plane. The camera pose can be represented as a combination of rotation matrix (R) and translation vector (t) or as a quaternion (q) and translation vector (t).

5. **Initialize the map and tracking**: Once the initial map points and camera pose are estimated, the Stereo PTAM system initializes the mapping and tracking layers. The map is created by inserting the 3D map points, while the tracking is initialized with the estimated camera pose. The system is now ready for subsequent tracking and mapping updates based on new incoming stereo frames.

The bootstrapping process in Stereo PTAM is crucial for a successful operation. It provides the system with an initial map and camera pose that can be refined and expanded as new data is processed.

5.3.4.2 Pose tracking thread

Each map point is projected into the viewing frustum of the anticipated stereo position, and we then look nearby for the match. A valid prediction of the current pose is required for such a projection. By contrasting the descriptions, map points and features are matched. The L_2 norm is computed using the binary descriptors of SuperPoint and R2D2. The match is valid if the distance falls below a certain threshold; otherwise, it is ignored. The pose refinement is then applied to recover the current pose knowing the previous one using the following objective function:

$$L^{\text{refine}} = \arg\min_{\mu} \sum_{i \in N} \rho(||J_i^k \mu_k - \Delta z_i(\mu_{k-1}, X_w^i)||^2) \ , \tag{5.14}$$

where $N = \{z_1, \ldots, z_M\}$ and M is the number of matched measurements. The measurement $z = [u, v]^\top$ is a pixel 2D location of the forward projection of a 3D map point X_w using the pinhole model projection function $\pi(X_w^i) = \mathcal{K}_c \mathcal{T}_i^{f_0 w} X_w^i$. $J_i^k = \frac{\partial \Delta z_i(\mu)}{\partial \mu_k}$ is the re-projection error's Jacobian with respect to the current odometry state vector. Δz is the re-projection error of a matched set of measurements on the current k stereo fusion frames and is defined as:

$$\Delta z_i(\mu, X_w) = z_i - \pi(\exp(\mu) \mathcal{T}_{k-1}^{f_0 w} X_w^i) \ , \tag{5.15}$$

where the 3D point cloud X_w is considered a constant optimization parameter and not updated in the tracking thread and $\mathcal{T}_{k-1}^{f_0 w} = exp(\mu) \in SE(3)$ with $\exp(.)$ the exponential map in the $\mathfrak{Lie}$ group for the previous increment state vactor. If the number of observed points is less than 90% of the points recorded in the previous keyframe, a frame is chosen to be a keyframe after the current pose has been evaluated. Then, new map points are created by triangulating the stereo pair's remaining mismatched features. The keyframe is then placed in the local mapping thread for processing.

In the Stereo Parallel Tracking and Mapping (Stereo PTAM) system, the tracking thread estimates the camera's current pose using the incoming stereo frames and the existing 3D map. The tracking thread works in parallel with the mapping thread, which maintains and updates the 3D map of the environment. Accurate and efficient camera pose estimation is crucial for navigation, localization, and obstacle detection in autonomous systems. Here's an overview of the key tasks performed by the tracking thread:

1. **Feature extraction**: For each new incoming stereo frame, the tracking thread extracts features from the left image using feature extraction algorithms like Scale-Invariant Feature Transform (SIFT), Oriented FAST, and Rotated BRIEF (ORB), or other learning-based extractors/descriptors. These features are used to establish correspondences with the 3D map points.

2. **Feature matching and map point projection**: The tracking thread matches the extracted features from the left image with the 3D map points by projecting the map points onto the image plane using the camera's previous pose estimate. This process results in a set of 2D-3D correspondences that will be used to refine the camera pose.

3. **Camera pose estimation**: The tracking thread estimates the current camera pose by solving a Perspective-n-Point (PnP) problem using the 2D-3D correspondences. The PnP problem aims to find the camera pose that minimizes the reprojection error between the observed 2D image points and the 3D map points projected onto the image plane. The camera pose can be represented as a combination of rotation matrix (R) and translation vector (t), or as a quaternion (q) and translation vector (t).

4. **Pose refinement**: The initial pose estimate is typically refined using an iterative optimization algorithm, such as the Levenberg-Marquardt algorithm, to reduce the reprojection error further and improve the pose accuracy.

5. **Tracking quality assessment**: The tracking thread evaluates the quality of the estimated camera pose based on criteria like the number of inliers (i.e., the correspondences with low reprojection error) and the distribution of the inliers in the image. If the tracking quality is deemed insufficient, the system may trigger a re-localization process to recover the camera pose using other techniques, such as searching for more keyframes in the map.

6. **Update system state**: Once the camera pose is estimated, the tracking thread updates the system's state with the new pose, allowing the robot or device to use this information for navigation, localization, and other tasks.

The tracking thread in Stereo PTAM is essential for real-time camera pose estimation, enabling the system to navigate and interact with its environment effectively. It works closely with the mapping thread to ensure the camera pose and 3D map are consistent and up-to-date.

5.3.4.3 Mapping thread

A type of least squares estimation known as Bundle Adjustment (BA) is used to fine-tune the camera poses (keyframe map) and the 3D points (point cloud map). Local Bundle Adjustment minimizes the re-projection error of every point in every keyframe f_0^k. Given an initial set of N keyframe poses $\{\mathcal{T}_1^{f_0 w}, \ldots, \mathcal{T}_N^{f_0 w}\}$, an initial set of M 3D points X_w^i, and measurement sets $S \in \{S_1, \ldots, S_N\}$, where each set comprises the measurement z_i^k of the i^{th} point in the k^{th} keyframe, the local BA is performed using the following objective function on all keyframes in a pre-defined sliding-window size N:

$$L^{\mathsf{BA}} = \underset{\mu,\, X_w}{\arg\min} \sum_{k=1}^{N} \sum_{i \in S_k} \rho(||J_i^k \begin{bmatrix} \mu_k \\ X_w^i \end{bmatrix} - \Delta z_i(\mu_k, X_w^i)||^2) \;, \tag{5.16}$$

where the 3D point cloud X_w is considered a variable optimization parameter and is updated in the mapping thread. Hence, the $J_i^k = [\frac{\partial \Delta z_i(\mu_k, X_w^i)}{\partial \mu_k}, \frac{\partial \Delta z_i(\mu_k, X_w^i)}{\partial X_w^i}]$ is the re-projection error's Jacobian with respect to the current odometry state vector and the 3D point as well.

In the Stereo Parallel Tracking and Mapping (Stereo PTAM) system, the mapping thread is responsible for maintaining and updating the 3D map of the environment using the information from the stereo camera. The mapping thread works in parallel with the tracking thread, which estimates the camera pose based on the current frame and the existing map. The mapping thread is critical in ensuring an accurate and consistent representation of the environment. Here's an overview of the key tasks performed by the mapping thread:

1. **Keyframe selection**: The mapping thread selects keyframes from the incoming stereo frames. Keyframes are chosen based on criteria such as significant camera motion or many new features observed since the last keyframe. Keyframes provide a basis for updating the map and ensure that the map is updated only when necessary, reducing computational load.

2. **Feature extraction and matching**: For each new keyframe, the mapping thread extracts features from the left and right images using feature extraction algorithms like Scale-Invariant Feature Transform (SIFT), Oriented FAST and Rotated BRIEF (ORB), or other learning-based architectures. The features are matched between the left and right images to establish correspondences and compute depth information.

3. **Triangulation**: The mapping thread triangulates the matched features using the stereo camera's known baseline and the matched feature pairs' disparities. This results in new 3D map points added to the map.

4. **Bundle Adjustment (BA)**: BA is an optimization process that refines the camera poses (keyframe poses) and 3D map points by minimizing the reprojection error between the observed 2D image points and the 3D map points projected onto the image plane. This process helps improve the map's overall accuracy and consistency.

5. **Map management**: The mapping thread continuously updates the map, adding new 3D map points and keyframes while removing redundant or poorly estimated points. This process ensures that the map remains reliable and efficient for camera pose estimation and path planning.

6. **Loop closure detection and correction**: In some Stereo PTAM implementations, the mapping thread handles loop closure detection, identifying when the camera returns to a previously visited location. If a loop closure is detected, the system corrects any accumulated drift by adjusting the camera poses and map points to maintain a consistent map.

The mapping thread in Stereo PTAM is essential for maintaining an accurate and up-to-date environment map, which is crucial for navigation, localization, and obstacle detection in autonomous systems.

5.3.4.4 Loop-closure thread

In the domain of Simultaneous Localization and Mapping (SLAM), loop closure detection is crucial to maintaining a consistent and accurate environment map. Various methods have been developed to address this challenge, primarily focusing on techniques that efficiently and reliably recognize previously visited locations. Among these approaches, the bags-of-words (BoW) model and mid-level feature-based methods have gained significant attention due to their robustness and scalability. The BoW model represents images as sparse histograms of visual words and quantized feature descriptors from a pre-defined vocabulary. The BoW approach enables efficient matching and retrieval of similar images by comparing their histograms, thus facilitating loop closure detection in large-scale environments. On the other hand, mid-level features, also known as part-based or semantic features, capture the structural and semantic information of the scene by leveraging object recognition, segmentation, or higher-level abstractions. These features provide a more discriminative and invariant representation than low-level features, increasing robustness to viewpoint changes, occlusions, and dynamic objects. By incorporating mid-level features into the loop closure detection process, the system can better handle challenging scenarios and improve overall map consistency. Bags-of-words and mid-level feature-based approaches have demonstrated their effectiveness in loop closure detection tasks, contributing to developing more reliable and accurate SLAM systems in various applications, from robotics to augmented reality.

The advent of deep learning has led to the development of powerful learning-based feature extractors and descriptors that have proven highly effective in computer vision tasks. These learning-based methods, particularly convolutional neural networks (CNNs), have been increasingly employed as mid-level features for loop closure detection in SLAM systems. Unlike handcrafted feature descriptors, learning-based features can capture hierarchical

and semantically meaningful information from the raw image data. They are trained on large datasets, which enables them to generalize well and provide more robust and discriminative representations of the scene.

In loop closure detection, learning-based features can be extracted from intermediate layers of a pre-trained CNN or networks designed explicitly for place recognition, such as NetVLAD or DenseVLAD. These features offer several advantages, including increased invariance to changes in viewpoint, illumination, and occlusions, leading to improved performance in challenging environments. Furthermore, learning-based features can be combined with traditional BoW or mid-level feature-based approaches to enhance the overall loop closure detection performance. For instance, by incorporating semantic segmentation or object recognition into the process, the system can focus on the most informative parts of the scene, improving its ability to recognize previously visited locations.

Integrating learning-based feature extractors and descriptors in SLAM systems has shown great promise in advancing the state-of-the-art in loop closure detection, paving the way for more robust, accurate, and efficient mapping and localization solutions across various applications.

In the Stereo Parallel Tracking and Mapping (Stereo PTAM) system, the loop closure thread is responsible for detecting and correcting loop closures to maintain a consistent and accurate map of the environment. Loop closures occur when the camera returns to a previously visited location, and recognizing these events is essential for correcting accumulated drift in the estimated camera poses and 3D map points. The loop closure thread works in parallel with the tracking and mapping threads. Here's an overview of the key tasks performed by the loop closure thread:

1. **Candidate selection**: The loop closure thread continuously monitors the system's state and selects potential loop closure candidates based on the current camera pose and the existing keyframes in the map. Candidates can be selected using various criteria, such as proximity or similarity in appearance.

2. **Feature extraction and matching**: For each loop closure candidate, the thread extracts features from the current frame and the candidate keyframe using the mean of the mid-level learning-based feature descriptors (SuperPoint and R2D2) for each keyframe and assign this mean value as the embedding identity of each keyframe. The features are then matched between the two frames to establish correspondences.

3. **Geometric verification**: The loop closure thread performs geometric verification to confirm the loop closure by estimating a relative transformation between the current frame and the candidate keyframe. This can be achieved using techniques like RANSAC-based homography, fundamental matrix estimation, or solving a Perspective-n-Point (PnP) problem.

4. **Loop closure detection**: If the geometric verification is successful and the number of inliers (i.e., the correspondences with low reprojection error) exceeds a predefined threshold, the system considers the loop closure as detected.

5. **Loop closure correction**: Once a loop closure is detected, the system corrects the accumulated drift by adjusting the camera poses and 3D map points. This process typically involves a global optimization step, such as pose graph optimization or global Bundle Adjustment (BA), to minimize the overall reprojection error and ensure a consistent map.

6. **Map merging and optimization**: After the loop closure correction, the loop closure thread may perform additional map merging and optimization tasks, such as merging duplicated map points or optimizing local sub-maps, to maintain an efficient and accurate map representation.

The loop closure thread in Stereo PTAM is essential for maintaining a consistent and accurate environment map, enabling the system to navigate and interact with its environment more effectively. It works closely with the tracking and mapping threads to ensure that the camera poses and 3D map are consistent and up-to-date.

5.4 Evaluation

We perform a thorough, comprehensive evaluation during navigation in real-world, large-scale, and small-scale areas in challenging settings. In subsection 5.4.1, we compare DH-PTAM with other standard image-based and event-based/-aided methods on the HDR large-scale sequences of the publicly available dataset VECtor [147] due to its high-quality ground truth values and sensors calibration parameters. In subsection 5.4.2, we evaluate the small-scale (mocap-) sequences of TUM-VIE [148] to test the quality of the DH-PTAM spatio-temporal synchronization method with degraded event camera calibration parameters. Moreover, the first 45 frames of TUM-VIE sequences suffer a high over-/under-exposure global shutter alternation, which tests the DH-PTAM's pose estimation stability. We perform a comparative quantitative analysis to evaluate the accuracy of our system in Tables 5.2, 5.3 and a qualitative/quantitative analysis in Figures 5.13, 5.15, 5.14. The accuracy of DH-PTAM is measured with absolute trajectory error (ATE), and relative pose error (RPE) metrics calculated using the baseline SLAM evaluation tool [169].

To prevail the advantages of complementing the sensor stack with events information, we compare our event-aided stereo visual odometry solution (DH-PTAM) to the latest best-performing open-source visual-inertial systems in literature in Table 5.2. Table 5.4 gives the system parameters configuration for large-scale and small-scale sequences. We keep these parameters constant for all sequences of the same scale group without an online fine-tuning process.

All experiments are performed on the CPU and the GPU of a 16 GB RAM laptop computer running 64-bit Ubuntu 20.04.3 LTS with AMD(R) Ryzen 7 4800h ×16 cores 2.9 GHz processor and a Radeon RTX NV166 Renoir graphics card. Table 5.5 reports a detailed computational complexity analysis for our DH-PTAM system with minimal and maximal system requirements. The high CPU load observed when detecting SuperPoint and R2D2 features can be attributed to the algorithms' design, which prioritizes feature quality and robustness over computational efficiency. This trade-

Table 5.2: DH-PTAM Quantitative Comparison Against the best performing open-source State-of-the-art SLAM Systems based on the Absolute Trajectory Error (ATE [m]) metric. The upper sub-table is for Standard Stereo VIO Methods, the middle is for event-based VO/VIO Methods, and the lower is for DH-PTAM. **Bold** denotes best performing, Underline for second best performing, and ($\times$) denotes failure

Method	VECtor sequences [147]						TUM-VIE sequences [148]					Mean	Mean
	corridors dolly	corridors walk	units dolly	units scooter	school dolly	school scooter	mocap 1d-trans	mocap 3d-trans	mocap 6dof	mocap desk	mocap desk2	VECtor large-scale	TUM-VIE small-scale
ORB-SLAM3 (SVIO) [43]	**0.802**	1.031	18.063	14.504	**0.921**	0.752	0.007	0.012	0.018	**0.007**	0.025	6.012	0.013
BASALT (SVIO) [69]	1.625	2.152	11.151	13.256	1.852	1.482	**0.003**	0.009	**0.014**	0.016	0.011	5.253	**0.011**
VINS-Fusion (SVIO) [112]	1.464	**0.392**	10.391	11.471	1.791	**0.562**	0.011	0.011	0.017	0.058	0.013	4.345	0.022
EVO (Mono Events) [158]	×	×	×	×	×	×	0.075	0.125	0.855	0.541	0.752	×	0.470
ESVO (Stereo Events) [46]	×	×	×	×	13.710	9.830	0.009	0.028	0.058	0.033	0.032	11.77	0.032
Ultimate SLAM (EVIO)$^{+}$ [160]	×	×	×	×	×	6.830	0.039	0.047	0.353	0.195	0.341	6.830	0.195
DH-PTAM (Stereo Fusion)	1.884	1.299	**5.274**	8.433	1.093	0.796	0.103	0.007	0.024	0.016	0.015	3.130	0.033
(SuperPoint on CPU) - RPE (σ)	0.073	0.038	0.055	0.149	0.178	0.074	0.006	0.007	0.009	0.009	0.007	0.095	0.008
DH-PTAM* (Stereo Image)	1.841	1.543	5.738	**5.010**	1.559	0.877	0.099	**0.004**	0.045	0.011	**0.008**	**2.761**	0.033
(R2D2 on GPU) - RPE (σ)	0.116	0.141	0.134	0.308	0.202	0.331	0.014	0.020	0.022	0.023	0.021	0.205	0.020

$^{+}$ IMU sensor is included since it is integrated into the front-end and cannot be separated for a fair comparison with EVO, ESVO, and DH-PTAM (ours).
* in this ablation case study, the SuperPoint detector is replaced with the R2D2 detector (trained for SLAM tasks), leveraging the GPU performance.

Table 5.3: DH-PTAM Quantitative Analysis based on the Relative Pose Error (RPE [m]) metric (for more qualitative results insights, refer to Figure 5.14 and Figure 5.15). **Bold** denotes best performing.

Dataset	Sequence	SuperPoint on CPU Stereo Fusion	R2D2 on GPU Stereo Images*
VECtor	corridors-dolly	**0.073±0.073**	0.116±0.058
	corridors-walk	**0.038±0.034**	0.141±0.057
	units-dolly	**0.055±0.046**	0.134±0.065
	units-scooter	**0.149±0.099**	0.308±0.157
	school-dolly	**0.178±0.099**	0.202±0.107
	school-scooter	**0.074±0.043**	0.331±0.203
TUM-VIE	mocap-1d-trans	**0.006±0.004**	0.014±0.009
	mocap-3d-trans	**0.007±0.004**	0.020±0.009
	mocap-6dof	**0.009±0.005**	0.022±0.012
	mocap-desk	**0.009±0.006**	0.023±.009
	mocap-desk2	**0.007±0.003**	0.021±0.008
Mean		**0.055±0.038**	0.121±0.063

* No events to complement the scene in this ablation case-study.

Table 5.4: DH-PTAM Parameters Configuration

Parameter	VECtor sequences	TUM-VIE sequences
δP^{h}_{align} - Left	(-160, -235) [px]	(355, 40) [px]
δP^{h}_{align} - Right	(-160, -235) [px]	(375, 45) [px]
frustum_near	0.1 [m]	0.1 [m]
frustum_far	30.0 [m]	5.0 [m]
matching_cell_size	15 [px]	15 [px]
matching_neighborhood	2 [px]	1.8 [px]
matching_distance	25 [px]	15 [px]

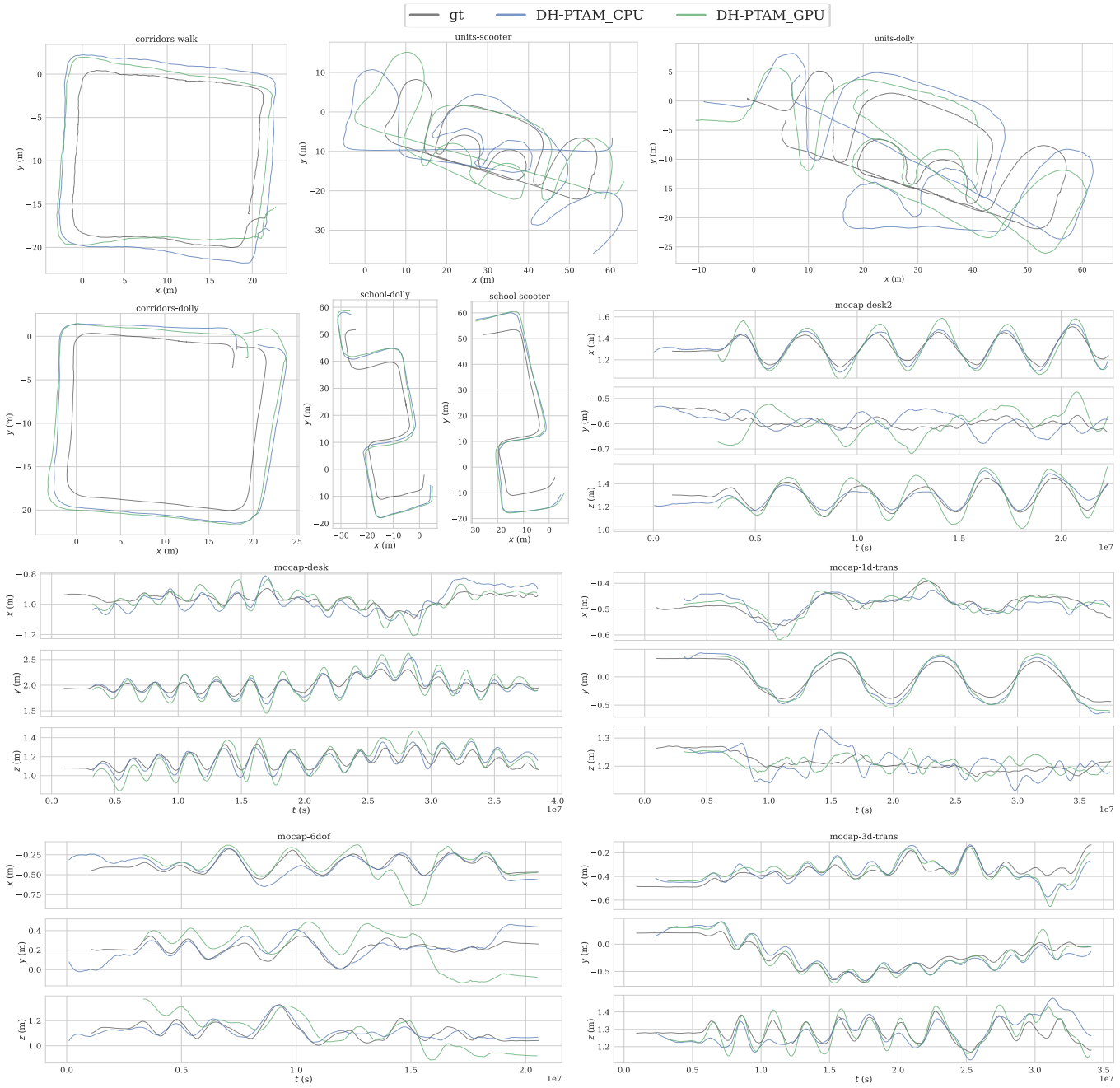


Figure 5.13: DH-PTAM (GPU (no events) vs. CPU (event-aided)) qualitative analysis. All trajectories are transformed to a reference frame as the ground truth poses using the extrinsic parameters, followed by an alignment with all poses by Umeyama's $\mathrm{SE}(3)$ method implemented by [169]. Large-scale trajectories show high-quality loop closure detection in the case of R2D2 on GPU. Small-scale trajectories show the high accuracy of the event-aided version of DH-PTAM.

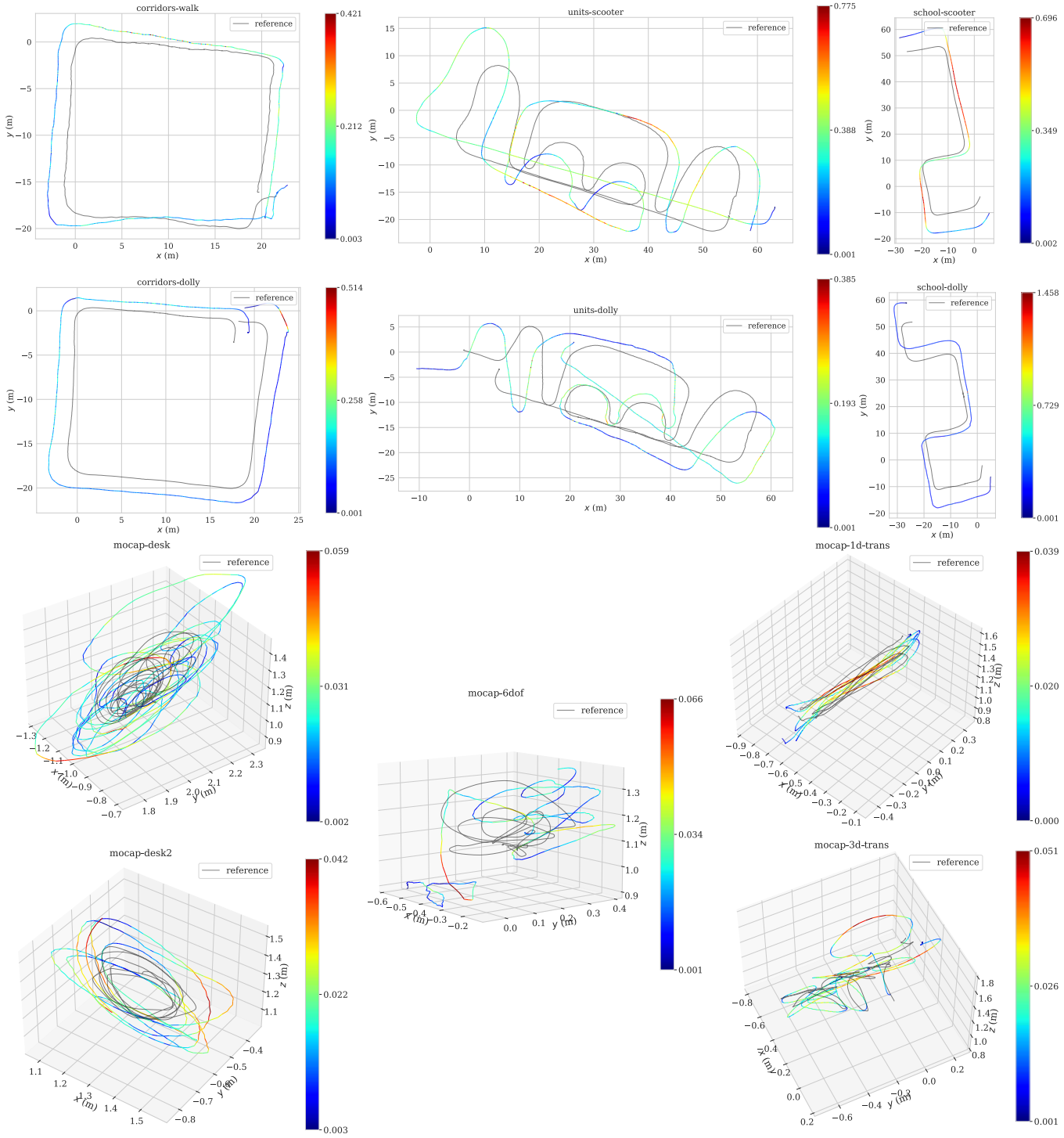


Figure 5.14: DH-PTAM (GPU (no events)) qualitative/quantitative analysis based on the positional relative pose error RPE [m] metric. The main observation from the low ATE and high RPE on the GPU, is due to the high-quality loop-closures detected using R2D2 features.

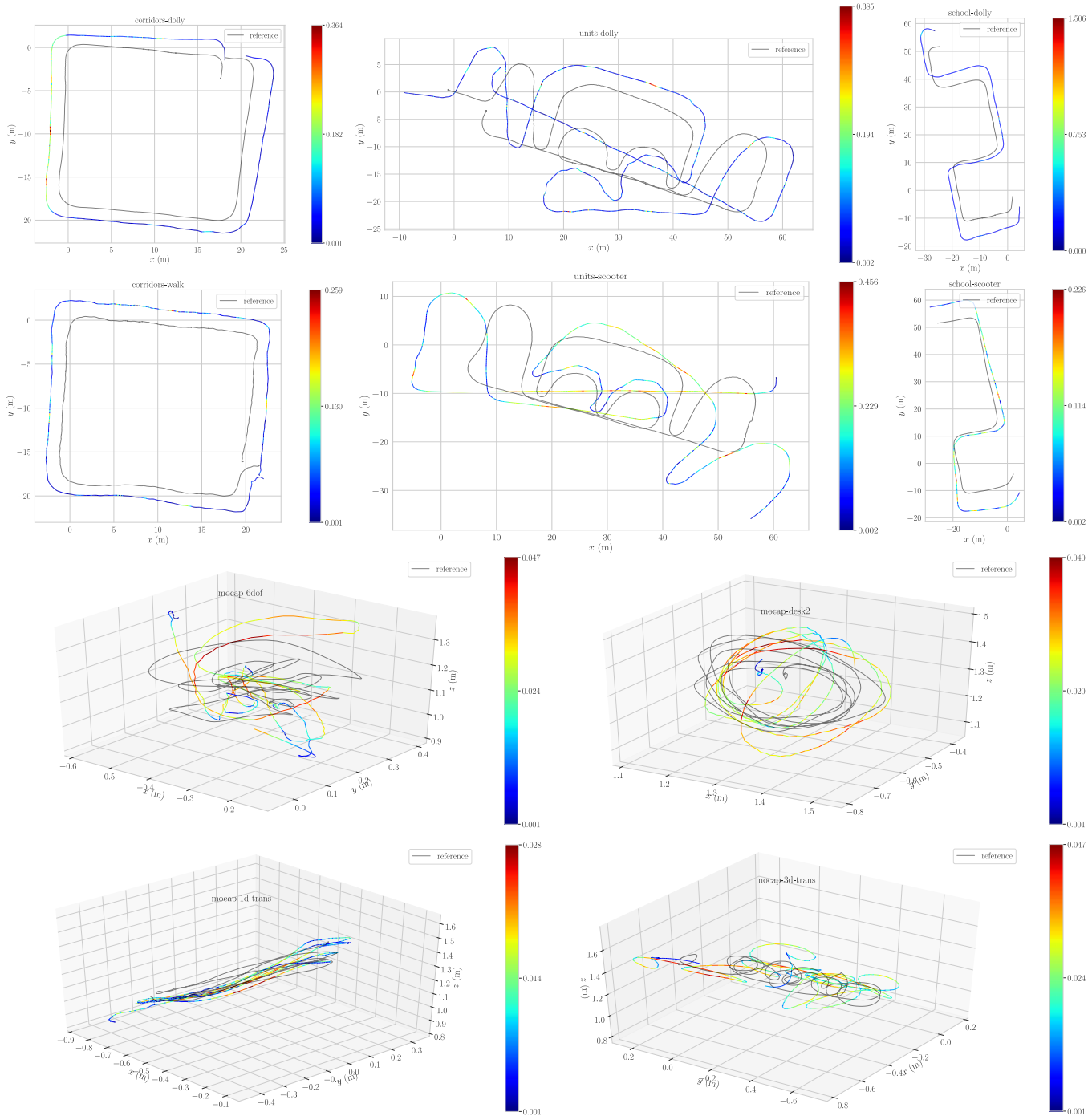


Figure 5.15: DH-PTAM (CPU (event-aided)) qualitative/quantitative analysis based on the positional relative pose error RPE [m] metric. The high visual drifts and the undetected loops with the large-scale sequences, is due to the low efficiency memory management in case of the learning-based features on the CPU leading to RAM overflow failures.

off is often necessary for computer vision research, where high-quality results are crucial for many applications but come at the cost of increased computational complexity. The back-end runs with real-time performance, and its recommended to run the front-end on a GPU to achieve a memory efficient, faster, and more stable performance.

No event streams ($\beta = 0$). In Tables 5.2, 5.3, we show an ablation study where we run DH-PTAM on stereo images. We notice estimation failure with all the conventional and learning-based feature detectors except R2D2. Although the ATE metric shows slightly better results without using events, the RPE metric shows much more accurate values when using events. These better ATE values are due to the high performance of the GPU in processing R2D2 feature detection (see Figure 5.13).

Table 5.5: Computational Complexity Analysis on CPU vs. GPU

Thread	#Tasks	Operation	CPU [ms]	GPU [ms]
Front-end	3	Stereo E3CT Construction	25-30	76-172
		Events-Frames Fusion	439-521	191-352
	($\approx$2$\times$10K)	SuperPoint Detection	2478-3256	521-1752
	($\approx$2$\times$4K)	R2D2 Detection	8532-8752	1067-2254
Bootstrapping	1	Initialize the Map	106-143	53-120
Tracking	2	Spatio-temporal Matching	161-215	142-172
	(10 iter.)	Pose Refinement	11-15	10-12
Mapping	2	Update Map	1-3	0.452-1
	(30 iter.)	Local Bundle-Adjustment	1-4	1-2
Loop-closing	3	Loop Detection	14-20	10-15
		Compute and Validate	2-5	1-3
	(30 iter.)	Pose Graph Optimization	1-3	0.524-2
End-to-End	11	SuperPoint Detector	3153-4208	356-1962
		R2D2 Detector	8560-9125	1226-2486

5.4.1 VECtor large-scale experiments

We notice a prominent estimation failure in Table 5.2 while evaluating the event-based methods EVO, ESVO and Ultimate SLAM on the large-scale sequences. Numerous factors may contribute to the failure of these systems, including stringent initialization requirements. For instance, the system EVO necessitates running in a sensor-planar scene for several seconds to bootstrap the system. Additionally, these systems are susceptible to parameter tuning, as demonstrated by using different parameters for different sequences in the same scenarios, even within their open-source projects.

Table 5.2 shows a good performance for DH-PTAM compared to the competing VI-SLAM systems. Although Figure 5.13 shows high visual drifts for our vision-only system in the case of units sequences, DH-PTAM could outperform

the VI-SLAM systems based on the ATE metric. Figure 5.13 gives an overview of the high-quality loop detection of DH-PTAM in the case of corridors sequences. Loop detection failure can be noticed only when the RAM overflows while running the system with enormous point clouds, as in the case of units sequences. We provide trajectory smoothing and post-processing script with our open-source implementation to join estimated trajectory increments in case of RAM overflow failures.

5.4.2 TUM-VIE small-scale experiments

As noticed in [170], the calibrationA (mocap-desk, mocap-desk2) sequences have more accurate depth estimation results than calibrationB (rest of mocap and TUM-VIE large-scale) sequences due to the significant calibration errors in the latter. Hence, we perform our comparative evaluation on TUM-VIE small-scale (mocap-) sequences using calibrationA parameters. Although the same high-quality calibrationA parameters apply to both desk2 and desk sequences with the same spiral motion, DH-PTAM performs the best with desk2 sequence but the worst with desk sequence. This occurs since the scene of the desk sequence is bounded by a close-by white wall that strict the depth, and hence DH-PTAM front-end detects low quality and fewer features for desk than desk2. Table 5.2 shows that the more DoF excited (6dof, desk2) and the consistent loops detection (1d-trans), the better the pose estimation quality based on ATE [m] metric.

5.5 Conclusion

This chapter presented the DH-PTAM system for robust parallel tracking and mapping in dynamic environments using stereo images and event streams. The proposed system builds upon the principles of S-PTAM and extends it with a deep learning-based approach to handle the sparse and noisy nature of event-based sensors while leveraging the rich information provided by fusion frames. Our experiments demonstrate that DH-PTAM outperforms state-of-the-art visual-inertial SLAM methods, particularly in challenging scenarios such as fast motion, HDR, and occlusions. The proposed system can achieve better performance on a GPU and provides a scalable and accurate solution for 3D reconstruction and pose estimation.

Our work has contributed significantly to the field of SLAM by developing a novel system that effectively combines the strengths of heterogeneous multi-modal visual sensors and employs deep learning-based feature extraction and description for estimation to enhance robustness. The DH-PTAM system has the potential to enable various applications in robotics, augmented reality, and autonomous driving, where robust and accurate 3D mapping and localization are critical for safety and efficient operation.

Future work includes investigating the potential of integrating inertial navigation sensors, such as IMUs, to further improve the system's robustness and accuracy in challenging environments. Additionally, exploring the integration of other deep learning components for feature extraction, matching, and loop-closure detection can potentially

enhance the overall performance and reliability of the system. Evaluating the DH-PTAM system in more diverse and challenging real-world scenarios will also be essential to validate its applicability and adaptability across a wide range of use cases.

In conclusion, the DH-PTAM system represents a significant advancement in the field of SLAM, offering a robust, scalable, and accurate solution that addresses the challenges associated with dynamic environments and heterogeneous sensor data. We believe that our work will pave the way for further research and development in this area, ultimately leading to more robust and efficient solutions for a variety of applications in robotics, autonomous navigation, and beyond.

6 Towards Event-based Dense SLAM

Abstract

In this chapter, we delve into unexplored territories within the realm of dense Simultaneous Localization and Mapping (SLAM) by focusing on utilizing Event cameras. Subsequently, we present our pioneering end-to-end approach for a dense event-based SLAM system. The proposed pipeline is constructed upon the open3D library, facilitating pose graph optimization. A straightforward loop-closure paradigm is employed based solely on the estimated hybrid point clouds. Rather than relying on the traditional Iterative Closest Point (ICP) method, we employ the efficient Teaser++ method for point cloud alignment and relative pose recovery, representing the current state-of-the-art approach. Lastly, we perform a proof-of-concept evaluation on DSEC and TUM-VIE, real-world public benchmarks. This evaluation demonstrates our proposed method's practical feasibility and effectiveness in a realistic setting, further solidifying its potential and value in the field.

"Imagination is more important than knowledge. For knowledge is limited, whereas imagination embraces the entire world, stimulating progress, giving birth to evolution."

Albert Einstein

6.1 Introduction and Related Works

Simultaneous Localization and Mapping (SLAM) is a fundamental problem in robotics and computer vision, which involves estimating the trajectory of a sensor-equipped agent while simultaneously constructing a map of the environment. Traditional SLAM approaches rely on visual sensors such as cameras, which provide dense pixel information. However, these methods often suffer from limitations such as high computational requirements and sensitivity to lighting conditions. In recent years, there has been a growing interest in utilizing event cameras, a type of sensor that captures asynchronous changes in pixel intensity, to overcome these challenges and enhance the performance of SLAM systems.

Event cameras offer several advantages over traditional cameras. They have a high temporal resolution, low power consumption, and a wide dynamic range, making them particularly suitable for high-speed and dynamic environments. Moreover, event cameras provide a sparse stream of events, which reduces the amount of data to process and enables real-time performance. These unique characteristics of event cameras make them promising candidates for dense SLAM applications.

Previous research in event-based SLAM [171] has predominantly focused on sparse mapping and tracking, neglecting the potential for dense reconstruction [59], [170]. While sparse methods have shown impressive results in terms of efficiency and accuracy, they suffer from limited environmental understanding due to the lack of dense geometric information. On the other hand, dense event-based SLAM aims to reconstruct a detailed and dense representation of the environment by leveraging the event stream.

Several approaches have been proposed to tackle the challenge of dense event-based SLAM. Some methods utilize traditional point cloud registration techniques from the Semi-global matching (SGM) algorithm [172] producing dense depth maps, such as Iterative Closest Point (ICP) [173], to align the event-based point clouds with the map. However, these methods may struggle with the sparsity and temporal nature of the event data, leading to suboptimal alignment results.

We propose a pioneering end-to-end approach for a dense, event-based SLAM system to address these limitations. Our pipeline builds upon the open3D library [174], which facilitates pose graph optimization. In contrast to traditional ICP-based methods, we employ the state-of-the-art Teaser++ method [58] for point cloud alignment and relative pose recovery. This method leverages efficient optimization techniques and has demonstrated superior performance compared to traditional methods.

This chapter presents the details of our proposed approach for dense event-based SLAM. We describe the pipeline, including the data preprocessing steps, the Teaser++ alignment method, and the loop-closure paradigm based on dense event-based point clouds. Furthermore, we evaluate our proposed method's practical feasibility and effectiveness through a proof-of-concept evaluation on real-world public benchmarks, namely DSEC [13] and TUM-VIE [37]. The results of this evaluation demonstrate the potential and value of our approach in a realistic setting,

paving the way for future advancements in dense event-based SLAM.

6.2 Methodology

The proposed event-based stereo-dense mapping pipeline 6.1 involves transforming stereo event streams into a dense 3D point cloud, followed by pose estimation and loop closure using the Teaser++ method. The methodology can be outlined as follows:

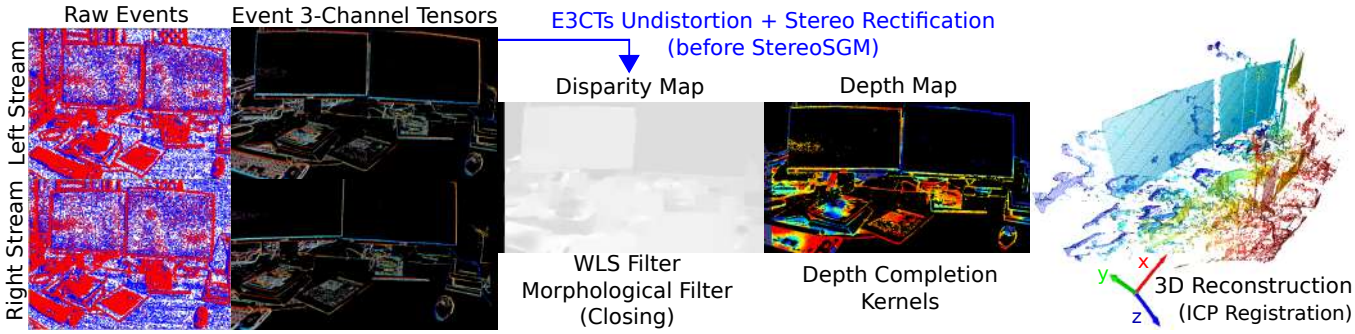


Figure 6.1: Our proposed event-based semi-dense SLAM system pipeline.

1. Acquisition of Stereo Event Streams:

   The system starts by acquiring stereo event streams from the event cameras. These cameras capture asynchronous changes in pixel intensity and provide a sparse stream of events, which form the basis of the subsequent processing steps.

2. Construction of Event 3-Channel Tensor (E3CT):

   To facilitate the processing of stereo events, the stereo event streams are converted into an Event 3-Channel Tensor (E3CT), denoted as $\mathbf{T}$. The E3CT represents the temporal information of events in a compact and structured format suitable for further computations.

3. Estimation of Disparity Map using Semi-Global Matching (SGM):

   The E3CT is used to estimate the disparity map, denoted as $\mathbf{D}$, which represents the pixel-wise disparity or depth difference between the left and right views of the stereo pair. The Semi-Global Matching (SGM) algorithm is applied to compute the disparity map by minimizing the energy function:

$$E(\mathbf{D}) = \sum_{i=1}^{N} (C_i(d_i) + P_i(d_i, d_{i-1}) + S_i(d_i))$$

   where $C_i(d_i)$ is the data cost, $P_i(d_i, d_{i-1})$ is the prior cost, and $S_i(d_i)$ is the smoothness cost. This optimization process produces the estimated disparity map.

4. Filtering of Disparity Map:

   The estimated disparity map is subjected to filtering techniques to improve its quality. First, the Weighted Least Squares (WLS) filter is applied to reduce noise and enhance the sharpness of disparity edges. The WLS filtering is formulated as:

$$\hat{\mathbf{D}} = \arg\min_{\mathbf{D}} \sum_{i=1}^{N} w_i \left(\mathbf{D}_i - \mathbf{I}_i\right)^2 + \lambda \left(\nabla \mathbf{D}\right)^2$$

   where $\hat{\mathbf{D}}$ is the filtered disparity map, $\mathbf{I}_i$ represents the intensity values, w_i is the weight for each pixel, and λ controls the smoothness regularization.

5. Disparity Completion using Morphological Closing Filter:

   To address occluded regions and fill in missing information, the disparity map undergoes disparity completion using a Morphological Closing filter. This filter employs a kernel to close gaps and smooth the disparities, ensuring a more complete and continuous depth estimation.

6. Conversion of Disparity Map to Depth Map:

   The filtered and completed disparity map, $\hat{\mathbf{D}}$, is converted to a depth map, denoted as $\mathbf{Z}$, using the focal length, f, and stereo baseline, b, of the camera system. The depth map is computed as:

$$\mathbf{Z} = \frac{f \cdot b}{\hat{\mathbf{D}}}$$

   providing a per-pixel estimate of the scene's depth or distance from the camera.

7. Conversion of Depth Map to 3D Point Cloud:

   The depth map, $\mathbf{Z}$, is further transformed into a 3D point cloud, denoted as $\mathbf{P}$. This conversion involves using the known camera parameters and geometry to map the depth values to their corresponding 3D coordinates. Each point in the point cloud is represented as $\mathbf{p}_i = (x_i, y_i, z_i)$, where x_i, y_i, and z_i are the coordinates in the 3D space.

8. Pose Estimation and Loop Closure using Teaser++:

   The consecutive estimated 3D point clouds, $\mathbf{P}_{t-1}$ and $\mathbf{P}_t$, are fed into the Teaser++ algorithm for pose estimation and loop closure. Teaser++ is a robust form of the traditional Iterative Closest Point (ICP) method (refer to Figure 6.2 for more insights). It performs point cloud registration and alignment to estimate the relative pose between consecutive frames, enabling accurate trajectory estimation. The registration process can be formulated as:

$$\mathbf{T} = \arg\min_{\mathbf{T}} \sum_{i=1}^{N} w_i \left\| \mathbf{R}\mathbf{P}_{t-1,i} + \mathbf{t} - \mathbf{P}_{t,i} \right\|^2$$

where $\mathbf{T} = (\mathbf{R}, \mathbf{t})$ represents the relative transformation between the point clouds, $\mathbf{P}_{t-1,i}$ and $\mathbf{P}_{t,i}$ are corresponding points in the point clouds, and w_i is the weight for each correspondence. The output of Teaser++ provides the relative pose $\mathbf{T}$ between consecutive frames, enabling accurate motion estimation and loop closure detection.

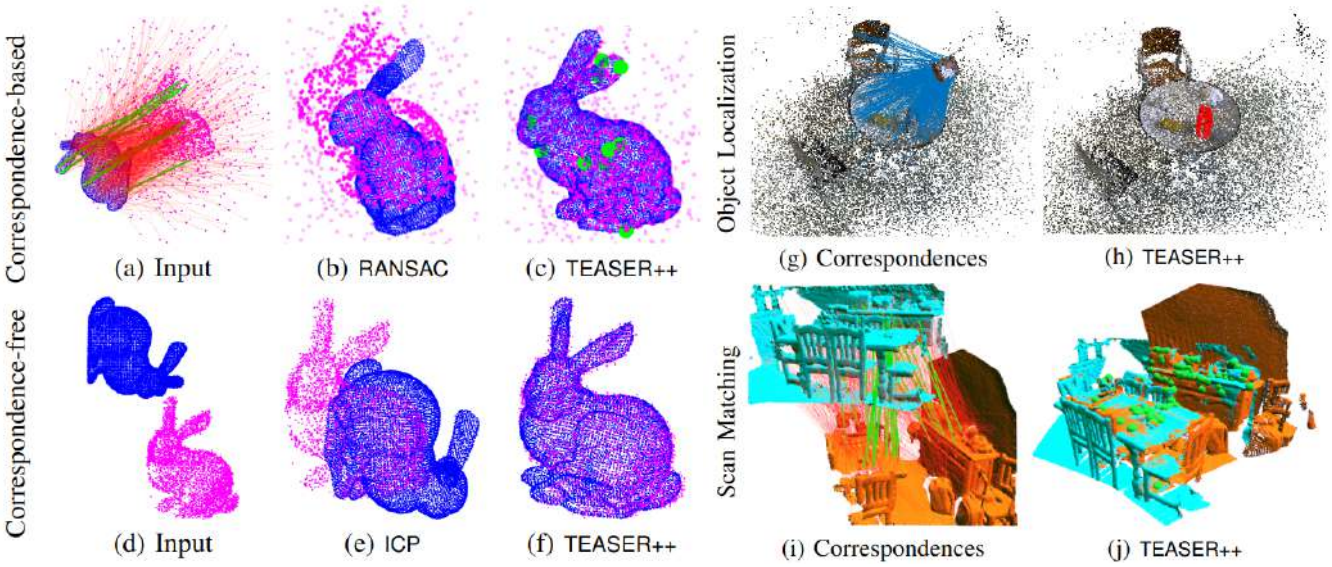


Figure 6.2: Comparison of 3D registration methods in the presence of outliers. (a) The Bunny dataset with 95% outliers (red lines) and 5% inliers (green lines). Existing algorithms like RANSAC (b) struggle to produce accurate estimates even after 10,000 iterations. The certifiable algorithm, TEASER, outperforms state-of-the-art in robustness and accuracy. The fast implementation, TEASER++ (c), computes precise millisecond estimates, even with extreme outlier rates, identifying the small inliers (green dots). TEASER++ excels in correspondence-free registration (d), where ICP (e) fails without a good initial guess, while TEASER++ (f) succeeds without requiring one. Tests on challenging RGB-D datasets for object localization (g-h) and scan matching (i-j), using traditional features (FPFH) and deep-learned features (3DSmoothNet), demonstrate the superior performance of TEASER++. The figure is courtesy of [58].

By following these steps, the proposed event-based stereo-dense mapping pipeline enables the reconstruction of a dense 3D representation of the environment while incorporating pose estimation and loop closure to improve the accuracy and robustness of the system.

6.3 A Proof-of-Concept Evaluation

6.3.1 Datasets Insights

We conducted evaluations of our stereo methods using sequences from five publicly available datasets. The DSEC dataset consisted of recordings with event cameras mounted on a car driving through Zurich's surroundings [13].

The TUM-VIE dataset [37], on the other hand, captured indoor and outdoor scenes using a sensor rig mounted on a helmet. For datasets that provided ground truth poses from a motion-capture system, we utilized those poses as input for all tested methods. In cases where camera poses were unavailable, such as in the TUM-VIE dataset, we computed them using data from the sensor rig, employing a visual-inertial odometry algorithm.

To enable quantitative assessment of the 3D reconstruction methods, certain datasets, including DSEC, included ground truth depth information. Depth measurements were obtained using a LiDAR operating at 10-20 Hz. However, it should be noted that event camera pixels corresponding to points outside the LiDAR's field of view or points in close proximity to the sensor rig might lack a LiDAR depth value.

Table 6.1 summarizes the main geometric parameters of the event cameras employed in the aforementioned datasets. The DAVIS camera configuration comprises frame- and event-based sensors sharing the same pixel array. Intrinsic and extrinsic calibration is performed using the intensity frames and subsequently applied to the event data. For datasets where the cameras exclusively output events (DSEC and TUM-VIE), calibration is achieved by converting events to frames and calibrating the frames using methods such as [54]. It is important to note that all methods employed in our study operate on undistorted coordinates.

Table 6.1: Experimental setups involved the utilization of stereo event-camera configurations, with the corresponding camera parameters being adjusted accordingly.

Dataset	Cameras	Resolution [pix]	Baseline [cm]	FOV [°]
DSEC	Prophesee Gen3	640 × 480	60	60.1
TUM-VIE	Prophesee Gen4	1280 × 720	11.84	90

6.3.2 Evaluation Metrics Insights

The performance evaluation of the proposed method is conducted using a comprehensive set of standard metrics on datasets that provide ground truth depth information, specifically the DSEC dataset. This evaluation encompasses various quantitative measures to assess the accuracy and robustness of the proposed method.

Firstly, the mean error (ME) and median error (MdE) between the estimated depth and the ground truth depth are calculated. The median error is favored in this evaluation due to its resilience to outliers. These errors are computed as follows:

$$\text{ME} = \frac{1}{N}\sum_{i=1}^{N}|D_i - \hat{D}_i|$$

$$\text{MdE} = \text{median}(|D_i - \hat{D}_i|)$$

where D_i represents the ground truth depth and $\hat{D}_i$ denotes the estimated depth for the i-th point.

Additionally, several other metrics are employed to analyze the proposed method's performance comprehen-

sively. These metrics include:

1. Number of Reconstructed Points (RP): This metric quantifies the total number of points successfully reconstructed by the proposed method.

2. Number of Outliers (NO): The number of outliers is determined using the bad-pix measure, which identifies points that deviate significantly from the ground truth depth.

3. Scale-Invariant Depth Error (SILog Err): This metric assesses the similarity between the estimated and ground truth depths, accounting for scale differences. It is calculated as:

$$\text{SILog Err} = \frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{\max(D_i,\hat{D}_i)}{\min(D_i,\hat{D}_i)}\right)^2$$

4. Sum of Absolute Relative Differences in Depth (AErrR): AErrR measures the relative difference between the estimated and ground truth depths. It is computed as:

$$\text{AErrR} = \frac{1}{N}\sum_{i=1}^{N}\frac{|D_i-\hat{D}_i|}{D_i}$$

5. δ-Accuracy Values: These values represent the percentage of points whose depth ratios with respect to the ground truth depth fall within a specified threshold. The depth ratio r_i for each point is calculated as $r_i = \frac{\hat{D}_i}{D_i}$, and the δ-accuracy is determined as the percentage of r_i values within the threshold.

The methodology presented in Section 6.2 is specifically tailored for events within a time window. To apply the method to an entire sequence, the sequence is divided into non-overlapping time windows, and the method is applied to each window individually. This approach comprehensively evaluates the proposed method's performance across the sequence, ensuring accurate and robust results.

6.3.3 Quantitative Analysis on DSEC Dataset

The proposed methods undergo a quantitative evaluation using the DSEC driving dataset, specifically focusing on the zurich04a sequence. The dataset provides maximum ground truth depth information up to 50 meters. The evaluation is conducted on a 35-second segment of stereo data, which comprises a substantial amount of information, including 635 million events and 350 ground truth depth maps. Notably, each depth map is computed using approximately 0.2 seconds of event data, equivalent to around 3.5 million events. For the evaluation, the ESVO method is executed by fusing two depth maps generated at a rate of 10 Hz, aligning with the frequency of the LiDAR data. In other words, each depth map is generated using 0.2 seconds of event data. The main observation from Table 6.2 is that our proposed methods variants (with and without the Morphological Closing FIlter) produce the densest point clouds at a very small cost on the overall depth estimation accuracy.

Table 6.2: Quantitative evaluation using the DSEC driving dataset

Algorithm	ME [m]	MdE [m]	NO [%]	SILog [×100]	AErrR [%]	log RMSE [×100]	δ <1.25 [%]	δ <1.25^2 [%]	δ <1.25^3 [%]	#RP [million]
EMVS (mono) [48]	5.64	2.52	13.68	13.23	25.52	36.49	72.56	87.12	93.56	1.31
ESVO [46]	3.88	1.56	12.08	9.23	18.89	30.80	**84.53**	**92.57**	**95.63**	3.40
MEC Depth [170]	**3.27**	**0.90**	10.75	**8.19**	**17.48**	**28.73**	83.30	91.56	95.62	1.25
MEC Depth + MF [170]	3.51	0.96	11.81	8.89	18.84	29.99	81.72	90.68	95.07	3.83
Proposed	4.79	3.23	6.87	32.90	27.84	35.51	49.29	84.07	94.44	57.62
Proposed + MF	4.73	3.20	**5.64**	31.95	28.01	35.31	47.44	85.64	94.08	**62.13**

MF: denotes the closing Morphological Filter.

6.3.4 Qualitative Analysis on TUM-VIE Dataset

The TUM-VIE dataset allowed us to conduct experiments using high-resolution event cameras (1Mpix) and evaluate our method's robustness to camera pose errors, as shown in Figure 6.3. Throughout our experiments, including those on DSEC and TUM-VIE, we consistently demonstrated the advantages of stereo over monocular methods, which include higher accuracy, outlier rejection, and faster convergence due to additional parallax information. We also investigated the sensitivity of our method to the camera's spatial resolution and contrast threshold, observing that higher resolution and lower threshold values result in improved accuracy at the cost of increased computational burden due to a larger number of input events. Our method does not require event simultaneity and can effectively fuse E3CTs even when constructed from temporally separated events. The best results were achieved when fusing E3CTs derived from identical time intervals.

6.4 Conclusion and Future Work

In conclusion, this thesis chapter explored dense Simultaneous Localization and Mapping (SLAM) using Event cameras. We presented a pioneering end-to-end approach for a dense event-based SLAM system, utilizing the Event 3-Channel Tensor (E3CT) and advanced techniques such as Semi-global matching (SGM), WLS filter, and Morphological Closing filter to estimate disparity maps and convert them into depth maps. The proposed method showcased practical feasibility and effectiveness by evaluating real-world benchmarks like DSEC and TUM-VIE, demonstrating its potential and value in the field.

Looking ahead, there are several promising avenues for future work. One interesting direction is the hybridization of depth maps estimated from the stereo E3CTs with depth maps obtained from other sensors such as stereo RGB cameras or LiDAR. This hybridization can leverage the complementary strengths of different sensing modalities and enhance the accuracy and robustness of the reconstructed 3D environment.

To achieve this hybridization, filtering or optimization techniques can be employed. Filtering methods like Kalman or particle filtering can fuse the depth maps from multiple sources and refine the final depth estimates. These methods effectively handle noise, uncertainties, and outliers in individual depth maps. Optimization-based approaches, such as bundle adjustment or graph optimization, can be employed to jointly optimize the parameters of the depth

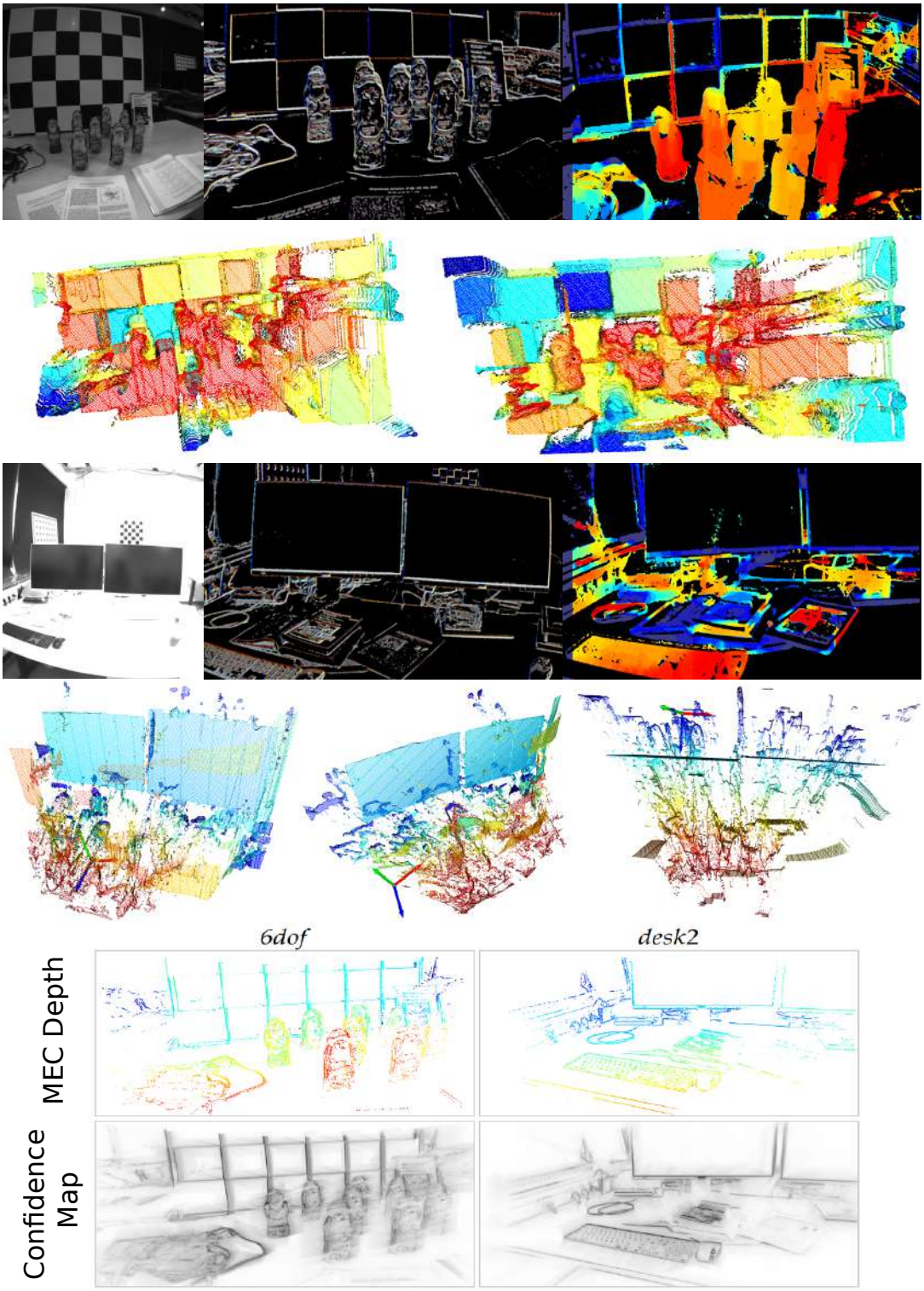


Figure 6.3: Top (2 rows): snippets of experiments on mocap-6dof from the TUM-VIE dataset that show from left to right grey-scale frame, E3CT frame, and the semi-dense depth map. The 2^{nd} row shows the 3D scene reconstruction. Middle (2 rows): snippets of experiments on mocap-desk2 from the TUM-VIE dataset that show from left to right grey-scale frame, E3CT frame, and the semi-dense depth map. The 2^{nd} row shows the 3D scene reconstruction. Bottom (2 rows): The MEC Depth method [170] (with the confidence map) is compared qualitatively to our proposed method output for the two TUM-VIE sequences (mocap-6dof and mocap-desk2).

maps from different sensors, ensuring consistency and improving the overall reconstruction quality.

Furthermore, integrating additional sensor information, such as color or intensity data from RGB cameras, can provide valuable cues for depth estimation. Incorporating the RGB camera depth maps into the event-based SLAM pipeline makes it possible to exploit the rich texture and visual features present in the RGB images, leading to more accurate and detailed depth estimation.

Additionally, exploring the fusion of event-based depth maps with LiDAR data can offer significant benefits. LiDAR provides precise and dense 3D measurements, which can serve as ground truth or strong constraints for optimizing the depth maps estimated from the event cameras. By incorporating LiDAR data into the fusion process, the final depth maps can benefit from the high accuracy and completeness of the LiDAR measurements.

Overall, the hybridization of depth maps estimated from stereo E3CTs with other sensor modalities, such as RGB cameras or LiDAR, through filtering or optimization techniques holds great potential for advancing the field of dense event-based SLAM. Further research and development in this direction can lead to more accurate, robust, and comprehensive 3D mapping systems with broader applicability in real-world scenarios.

7

Conclusions and perspectives

Abstract

This chapter serves as a summary of the contributions and key findings presented in the preceding chapters of this Ph.D. thesis. The research conducted in this thesis puts forth several innovative solutions that contribute to the advancement of multi-modal heterogeneous sensor fusion in the context of autonomous systems' navigation within large-scale and dynamic environments.

"I am among those who think that science has great beauty. A scientist in his laboratory is not only a technician: he is also a child placed before natural phenomena which impress him like a fairy tale."

Marie Curie

7.1 Conclusions

This Ph.D. thesis has addressed the challenges of sensor fusion and Simultaneous Localization And Mapping (SLAM) for autonomous systems, specifically focusing on Autonomous Ground Vehicles (AGVs) and Micro Aerial Vehicles (MAVs) navigating large-scale and dynamic environments. Through the development of innovative solutions, the research has significantly advanced the field of visual odometry and SLAM.

In the first methodological chapter, we introduced IBISCape, a simulated benchmark for validating high-fidelity SLAM systems, including data synchronization and acquisition APIs for telemetry from heterogeneous sensors, ground truth scene segmentation, depth maps, and vehicle ego-motion. The chapter also proposed innovative calibration targets and a pre-processing layer for integrating DVS sensor events in any frame-based Visual-SLAM system.

In the second methodological chapter, we presented a novel approach for intrinsic and extrinsic calibration of an RGB-D-IMU visual-inertial setup using a GPS-aided optimizer bootstrapping algorithm. Our method delivers reliable initial estimates for the RGB camera intrinsics and trajectory while optimizing spatio-temporal parameters. Extensive experimental results on real-world and simulated sequences confirm the effectiveness and robustness of our method.

The third methodological chapter focused on developing an accurate and computationally inexpensive localization solution for MAVs in large-scale environments. We proposed a decoupled optimization- and filtering-based sensor fusion technique, achieving high estimation accuracy and minimum system complexity. The results from real-world indoor and outdoor settings demonstrated the method's reliability and performance compared to other techniques in the literature.

In the fourth methodological chapter, we introduced the DH-PTAM system for robust parallel tracking and mapping in dynamic environments using stereo images and event streams. By leveraging deep learning-based feature extraction and description, DH-PTAM outperforms state-of-the-art visual-inertial SLAM methods in challenging scenarios. The system is scalable and accurate, providing an effective solution for 3D reconstruction and pose estimation.

The fifth methodological chapter explored new frontiers in the field of dense SLAM using Event cameras. We presented a novel end-to-end approach for an event-based dense SLAM system, achieving spatio-temporal hybridization of the stereo events and point clouds using the efficient probabilistic approach Teaser++. The proof of concept evaluation was performed on DSEC, a real-world public benchmark.

Overall, the contributions of this thesis have significantly advanced research in multi-modal heterogeneous sensor fusion applied to autonomous systems navigating large-scale and dynamic environments. The proposed benchmarks, calibration targets, and pre-processing layers offer reliable validation of SLAM systems. The proposed calibration and SLAM algorithms enable more accurate and robust pose estimation, while the proposed sensor fusion

techniques achieve high-accuracy localization with minimal system delay.

7.2 Perspectives

Building on the significant contributions of this thesis, there are several promising avenues for future research:

- Extending the proposed algorithms to support multiple vision sensors (stereo RGB, for instance) and multiple IMUs would enhance the system's capabilities and adaptability to different configurations and applications.
- Investigating the application of the proposed SLAM methods to other autonomous systems, such as underwater vehicles and drones, to validate their performance and robustness in various contexts.
- Developing methods for efficient map management, map merging, and map updating in long-term SLAM applications, addressing challenges related to dynamic changes in the environment and the need for efficient data storage and retrieval.
- Enhancing the real-time performance of the proposed algorithms by optimizing their computational efficiency, possibly through hardware acceleration or parallelization techniques.
- Extending the algorithms' online calibration and pose estimation capability to include multiple IMUs with multiple vision sensors (RGB and depth), thereby generalizing the optimization problem and enabling more complex sensor configurations.
- Investigating the potential for incorporating semantic information into the proposed SLAM algorithms to enable richer scene understanding, better loop-closure detection, and improved robustness in highly dynamic environments.

In conclusion, this Ph.D. thesis has made significant contributions to the field of SLAM and sensor fusion for autonomous systems. The innovative solutions presented throughout the methodological chapters have shown great promise in addressing the challenges of large-scale and dynamic environments. The perspectives outlined above provide a road-map for future research, which will continue to advance the state-of-the-art in this exciting and rapidly evolving field.

A CARLA Synchronization Modes

We generate data by eight acquisition APIs with four sensor setups mentioned in Tab. 2.4 in two groups: 1. calibration and 2. SLAM. SLAM data acquisition APIs run on all CARLA maps with an autopilot for traffic-aligned navigation. On the other hand, calibration APIs run on our modified CARLA-map with manual vehicle control to apply desired motions to collect sequences with basic or complex motions. Both AprilGrid and Checkerboard targets are introduced during acquisition. Half of the calibration sequences are collected using the AprilGrid 6×6 and the other half using the Checkerboard 7×7.

In order to operate all sensors in the same acquisition API on multiple frequencies, we develop the following procedure: the core data acquisition concept is that the CARLA world clock ticks with the highest frequency sensor in the setup. After that, the system waits to listen to all sensors sending data at this tick, updates the weather conditions, and waits for a new world tick. This allows the acquisition of all sensors data with its occurrence timestamps. Then, one can apply any synchronization/calibration algorithms on the collected datasets as in [9, 11]. We apply this methodology (see Program A.1) to all sensor setups except the RGB-D setup, which requires time-synchronized and registered frames.

Program A.1: Normal Data Acquisition Mode.

```
data = []
sensor_list = Create_Sensors_List()
Create_Senors_Listener_handler(sensor_list)
Create_Weather_Control_class()
while (CARLA_world_tick()):
    Update_world_weather()
    for sensor in sensor_list:
        sensor.listen()
        if (RECORD_ON()):
```

```
            data.append(sensors.data())
```

On the contrary, the CARLA world ticks with the lowest frequency sensor in the LiDAR/RGB-D setup with CARLA `synchronous_mode` acquisition (see Program A.2). All the spawned sensors in the setup are stacked in a queue waiting for the world's tick to start listening to the data. Although all sensors operate with their frequencies, the API reads the measurements of all sensors simultaneously at the timestamp of that CARLA world tick.

Program A.2: Synchronous Data Acquisition Mode.

```
data = []
sensors = Create_Sensors_List()
Create_Senors_Synchronization_Queue_class()
Create_Weather_Control_class()
while (CARLA_world_tick()):
    Update_world_weather()
    Sensors_Queue.tick(sensors)
    Sensors_Queue.listen(sensors)
    if (RECORD_ON()):
        data.append(sensors.data())
```

The open source data acquisition APIs and all sequences can be accessed using the Github repository: `https://github.com/AbanobSoliman/IBISCape.git`

In the repository there is a complete manual on how to execute the APIs in all setups and options, including a library developed for IBISCape dataset files format to be processed using Robotic Operating System (ROS) based algorithms. Besides the Python based ROS tools, we attach the configuration files for all the assessed algorithms along with the Kalibr calibration results.

B

On Manifold IMU Online Calibration

This appendix presents a quantitative analysis of a sample undergoing experiments on manifold IMU online calibration. This analysis focuses on the smoothness of the 2D path plots generated using cumulative and non-cumulative B-splines, as well as the discrete-time path generated using ground truth. The analysis considers the C^1 and C^2-continuity conditions and evaluates the performance of the sample against these criteria. The results of this analysis provide valuable insights into the effectiveness of manifold IMU online calibration and contribute to the ongoing research in this field.

Table B.1: Sample size for the discrete-time and continuous-time poses, velocities & accelerations.

Data	Size
Control Poses (discrete-time)	434
Quadratic B-spline (continuous-time)	23760
Cubic B-spline (continuous-time)	23705
Eff. & BL. (Accel./Vel.) Estimations	23705
IMU readings (Accel. & Gyro.)	21741
IMU Estimated Biases (b_w, b_a)	21741

Table B.2: Positions on B-spline trajectory analysis

Degree	Type	Domain		Smoothness		Time to Generate (sec.)	
Quadratic	Cumulative	$\mathrm{R}(3)$	$\mathrm{SE}(3)$	Medium	Medium	0.0072	2.74785
	Non-Cumulative	$\mathrm{R}(3)$		Low		0.0046	
Cubic	Cumulative	$\mathrm{R}(3)$	$\mathrm{SE}(3)$	High	High	0.0179	3.7248
	Non-Cumulative	$\mathrm{R}(3)$		Medium		0.0077	

Table B.3: Cumulative Orientations on B-spline trajectory analysis

Degree	Domain	Smoothness	Time to Generate (sec.)
Quadratic	$\mathrm{SO}(3)$	Low	2.5640
Quadratic	$\mathrm{SE}(3)$	Medium	2.74785
Cubic	$\mathrm{SO}(3)$	Medium	3.3083
Cubic	$\mathrm{SE}(3)$	High	3.7248

The upcoming appendix subsections provide a detailed analysis of the performance of IMU bias estimation using time derivative B-spline models. Specifically, we compare the accuracy of biases estimated using these models on the ground truth values of the EuRoC non-linear estimator with those estimated using the Vicon system alone. Our results show that the biases estimated using the B-spline models are highly accurate, providing a valuable initial calibration step for a multi-modal framework of sensors.

Moreover, we demonstrate that our linear calibration framework has a low processing load and can generate highly accurate values for biases, which can be used as a reliable initial guess in any non-linear estimator. We also evaluate the performance of the B-spline time derivatives proposed by [93] from both qualitative (smoothness) and quantitative (time for generation) perspectives. Our analysis shows that these time derivatives are faster and more reliable than the baseline algorithm used in a multi-modal sensor calibration framework.

Overall, our findings demonstrate the effectiveness of the **Efficient** time derivative B-spline models by [93] in IMU bias estimation and their potential for use in multi-modal sensor calibration.

Table B.4: IMU Online Calibration using the Baseline and Efficient Models of the Generative B-spline in $\mathrm{SE}(3)$.

V101	b_{a_x}	b_{a_y}	b_{a_z}	b_{ω_x}	b_{ω_y}	b_{ω_z}	**V101**	b_{a_x}	b_{a_y}	b_{a_z}	b_{ω_x}	b_{ω_y}	b_{ω_z}
Baseline	-0.011469	0.198229	0.081414	-0.002119	0.023376	0.076494	**Baseline**	-0.006148	0.543458	0.068384	-0.002177	0.021956	0.076343
Efficient	-0.018737	0.200878	0.073887	-0.002171	0.021169	0.076489	**Efficient**	-0.013231	0.547694	0.063963	-0.002219	0.020867	0.076295
Vicon	-0.012492	0.547666	0.069073	-0.002229	0.020700	0.076350	**Optimizer**	-0.012492	0.547666	0.069073	-0.002229	0.020700	0.076350
V102	b_{a_x}	b_{a_y}	b_{a_z}	b_{ω_x}	b_{ω_y}	b_{ω_z}	**V102**	b_{a_x}	b_{a_y}	b_{a_z}	b_{ω_x}	b_{ω_y}	b_{ω_z}
Baseline	0.025622	0.140923	0.010189	-0.000306	0.030212	0.075139	**Baseline**	0.017615	0.068431	0.058637	-0.000842	0.023458	0.075741
Efficient	-0.009334	0.175685	0.030821	-0.002141	0.023146	0.075064	**Efficient**	-0.013839	0.106027	0.095274	-0.002014	0.020023	0.075965
Vicon	-0.013337	0.103464	0.093086	-0.002153	0.020744	0.075806	**Optimizer**	-0.013337	0.103464	0.093086	-0.002153	0.020744	0.075806
V103	b_{a_x}	b_{a_y}	b_{a_z}	b_{ω_x}	b_{ω_y}	b_{ω_z}	**V103**	b_{a_x}	b_{a_y}	b_{a_z}	b_{ω_x}	b_{ω_y}	b_{ω_z}
Baseline	0.066031	0.186584	-0.031513	-0.001386	0.027252	0.074564	**Baseline**	0.040577	0.168604	0.014779	-0.001504	0.024185	0.075915
Efficient	-0.000540	0.200116	0.019334	-0.002320	0.022748	0.076470	**Efficient**	-0.023387	0.182120	0.081518	-0.002402	0.021809	0.076622
Vicon	-0.022808	0.177689	0.090354	-0.002341	0.021815	0.076602	**Optimizer**	-0.022808	0.177689	0.090354	-0.002341	0.021815	0.076602

B.1 Calibration results using EuRoC IMU and Vicon as ground truth

B.1.1 EuRoC Dataset: Vicon room 1 “easy”

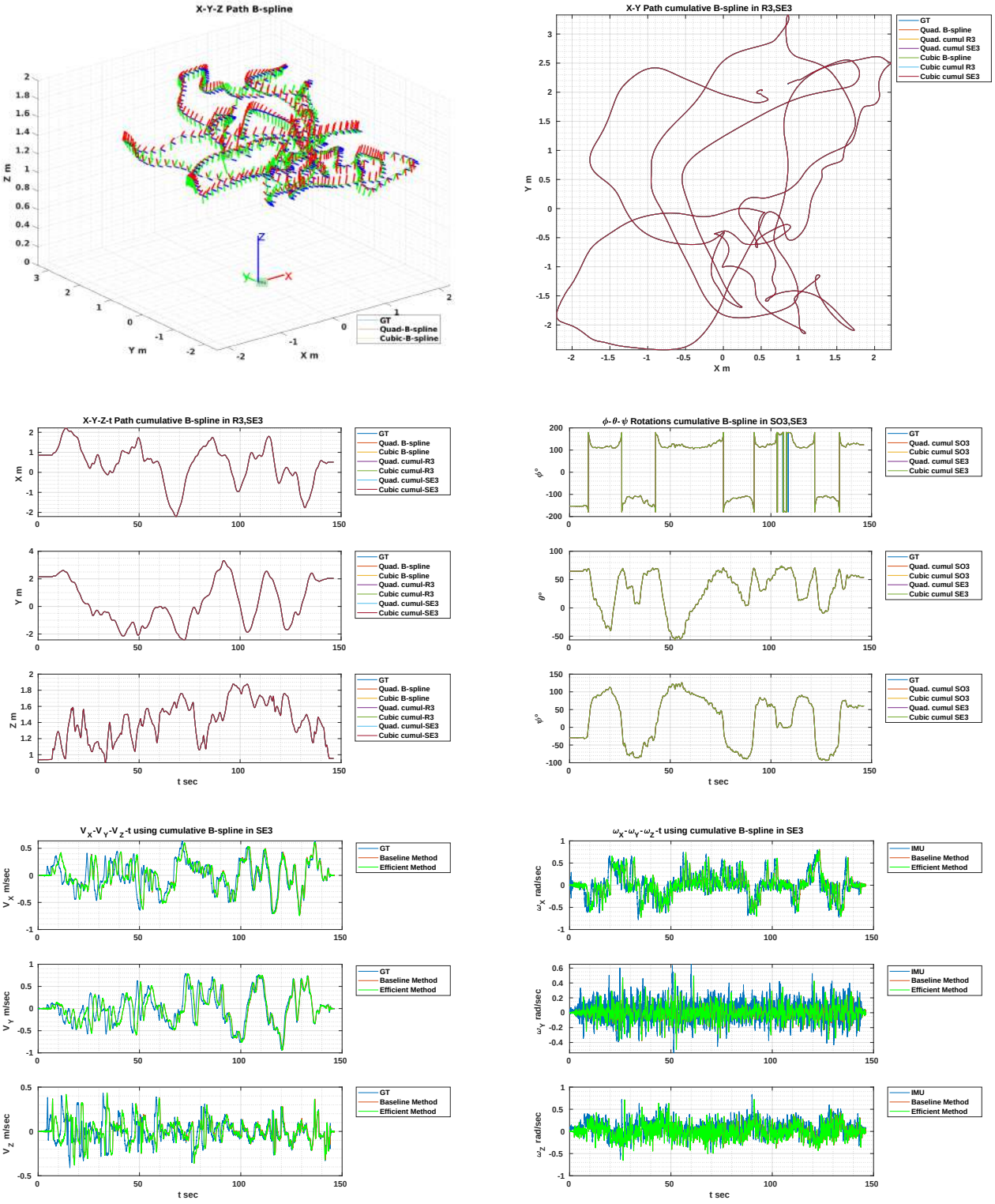


Figure B.1: Vicon room 1 Easy: B-spline comparison in $\mathrm{R}(3), \mathrm{SE}(3)$

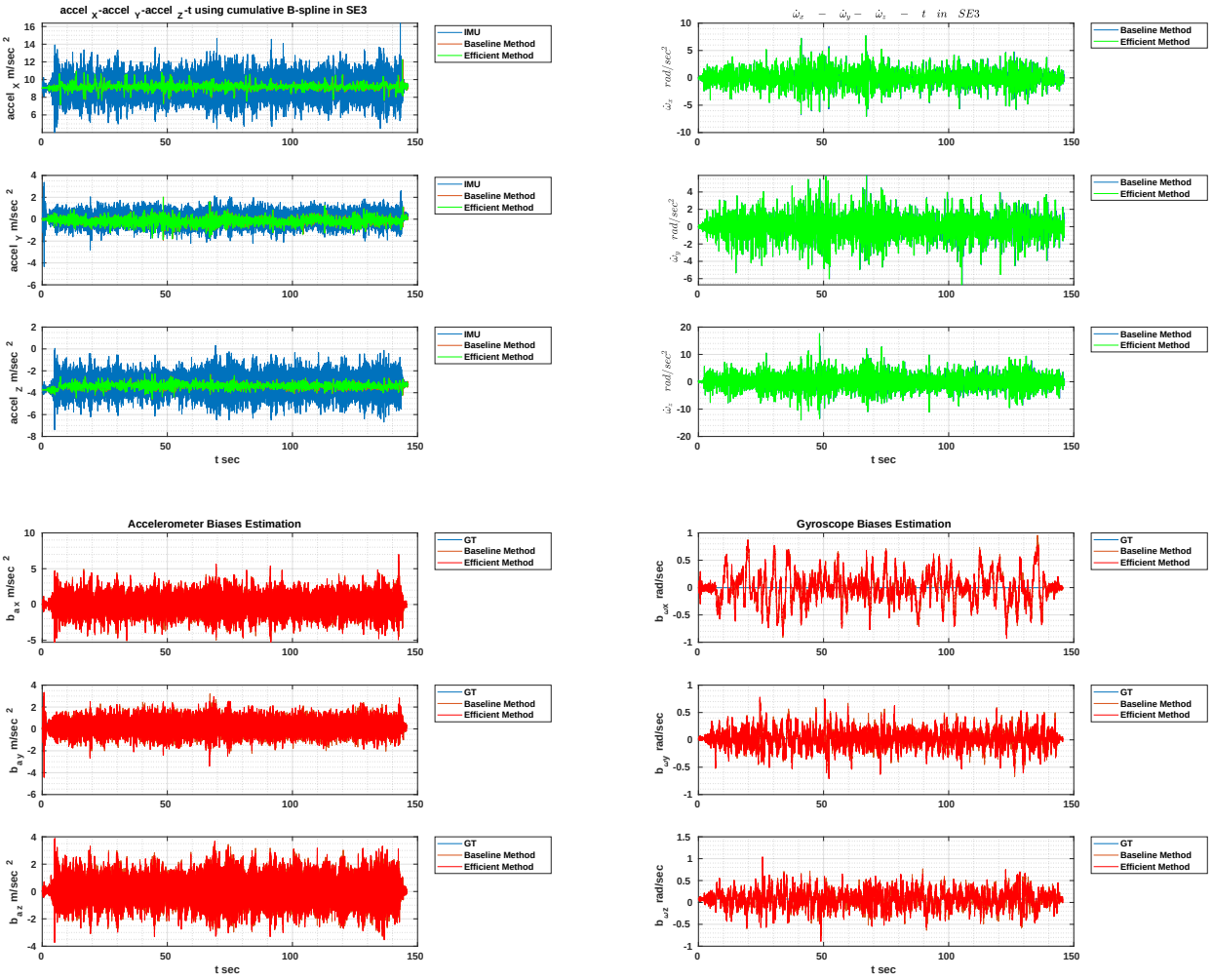


Figure B.2: Vicon room 1 Easy: Baseline/Efficient/GT comparison

The IMU online calibration experiment on the V101-Easy sequence shows high-precision accelerometer and gyroscope biases estimation based on the **Efficient** B-spline model compared to the baseline model using the Vicon measured trajectory.

B.1.2 EuRoC Dataset: Vicon room 1 “medium”

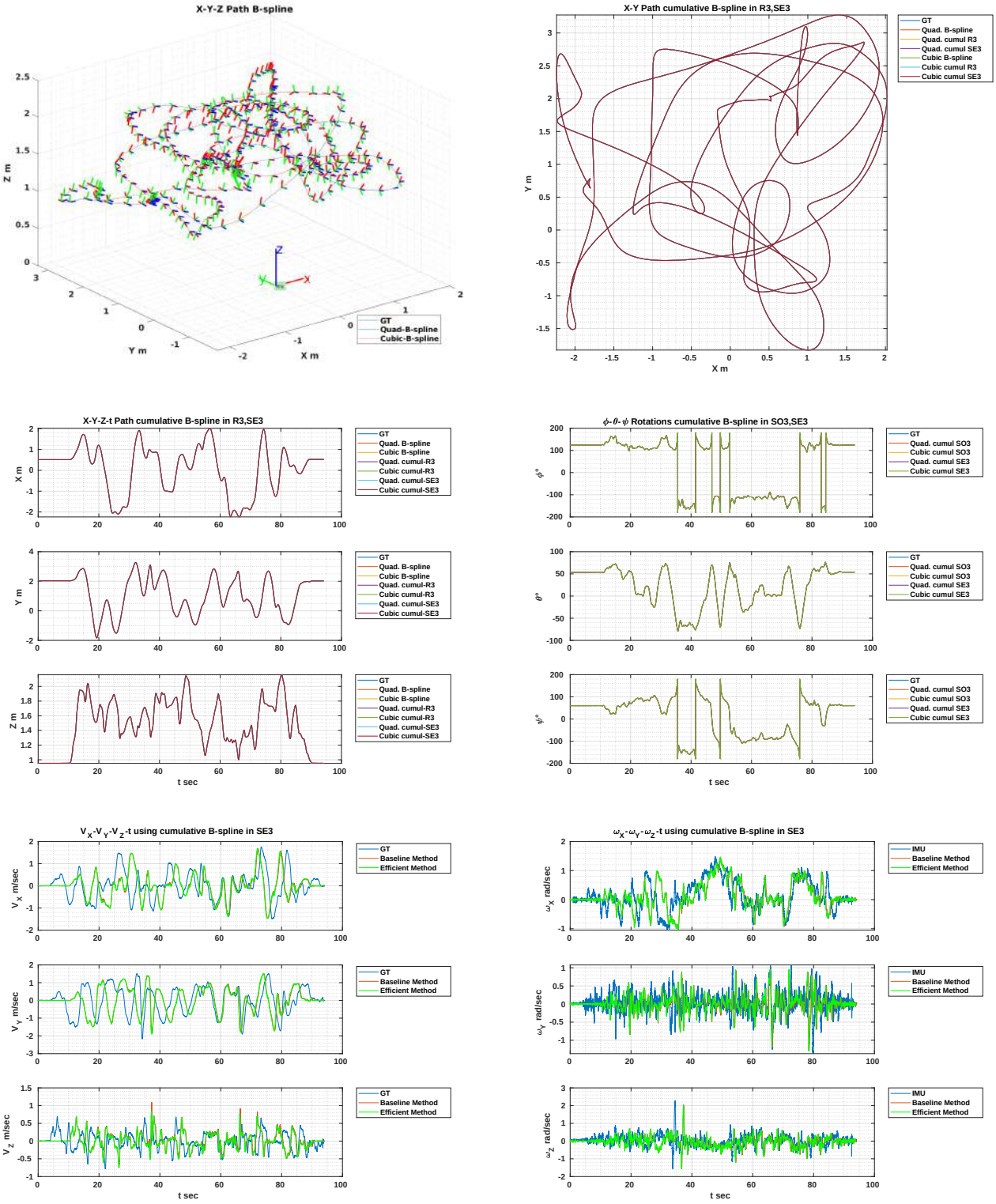


Figure B.3: Vicon room 1 Medium: B-spline comparison in $\mathrm{R}(3), \mathrm{SE}(3)$

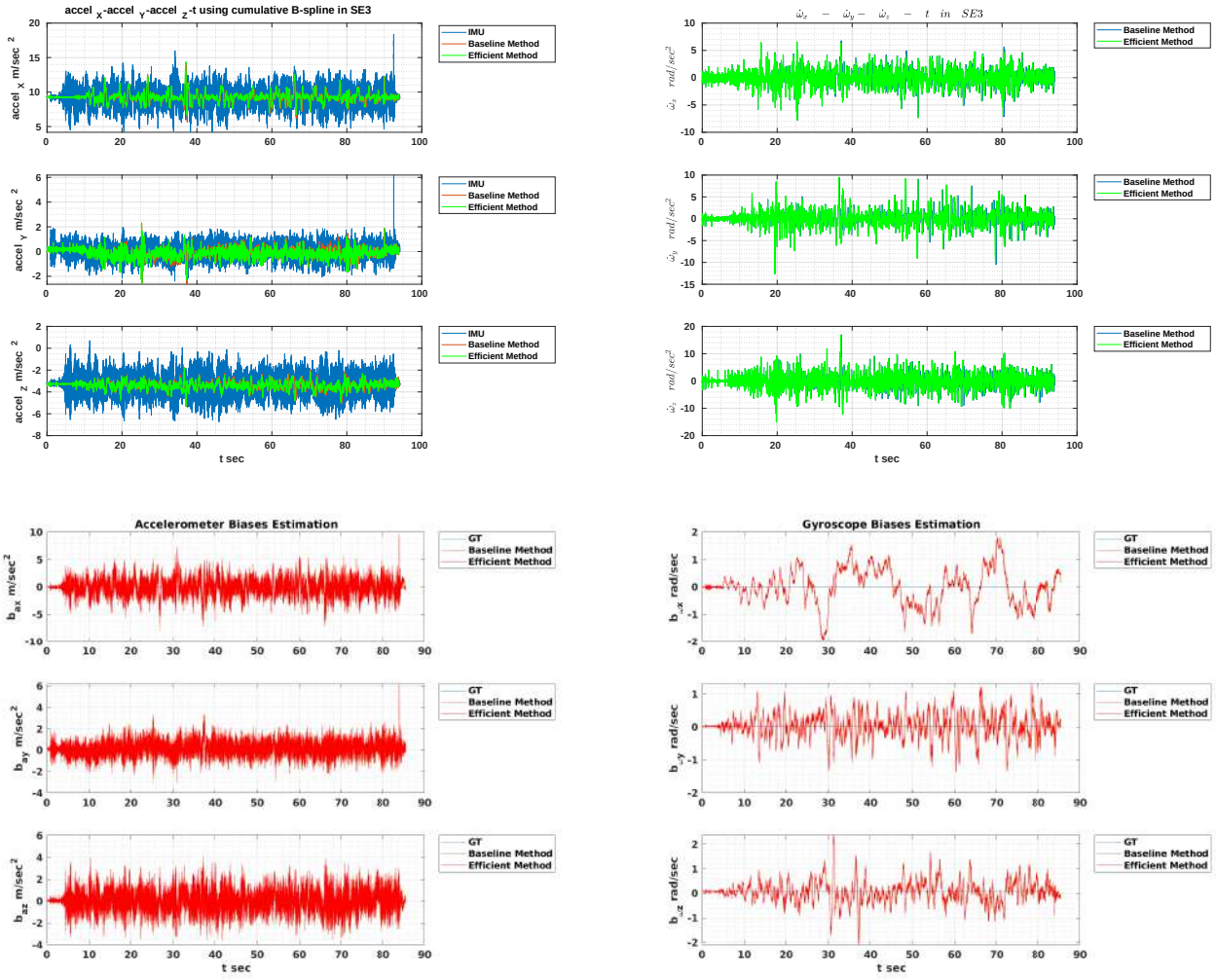


Figure B.4: Vicon room 1 Medium: Baseline/Efficient/GT comparison

The IMU online calibration experiment on the V102-Medium sequence shows high-precision accelerometer and gyroscope biases estimation based on the **Efficient** B-spline model compared to the baseline model using the Vicon measured trajectory.

B.1.3 EuRoC Dataset: Vicon room 1 “difficult”

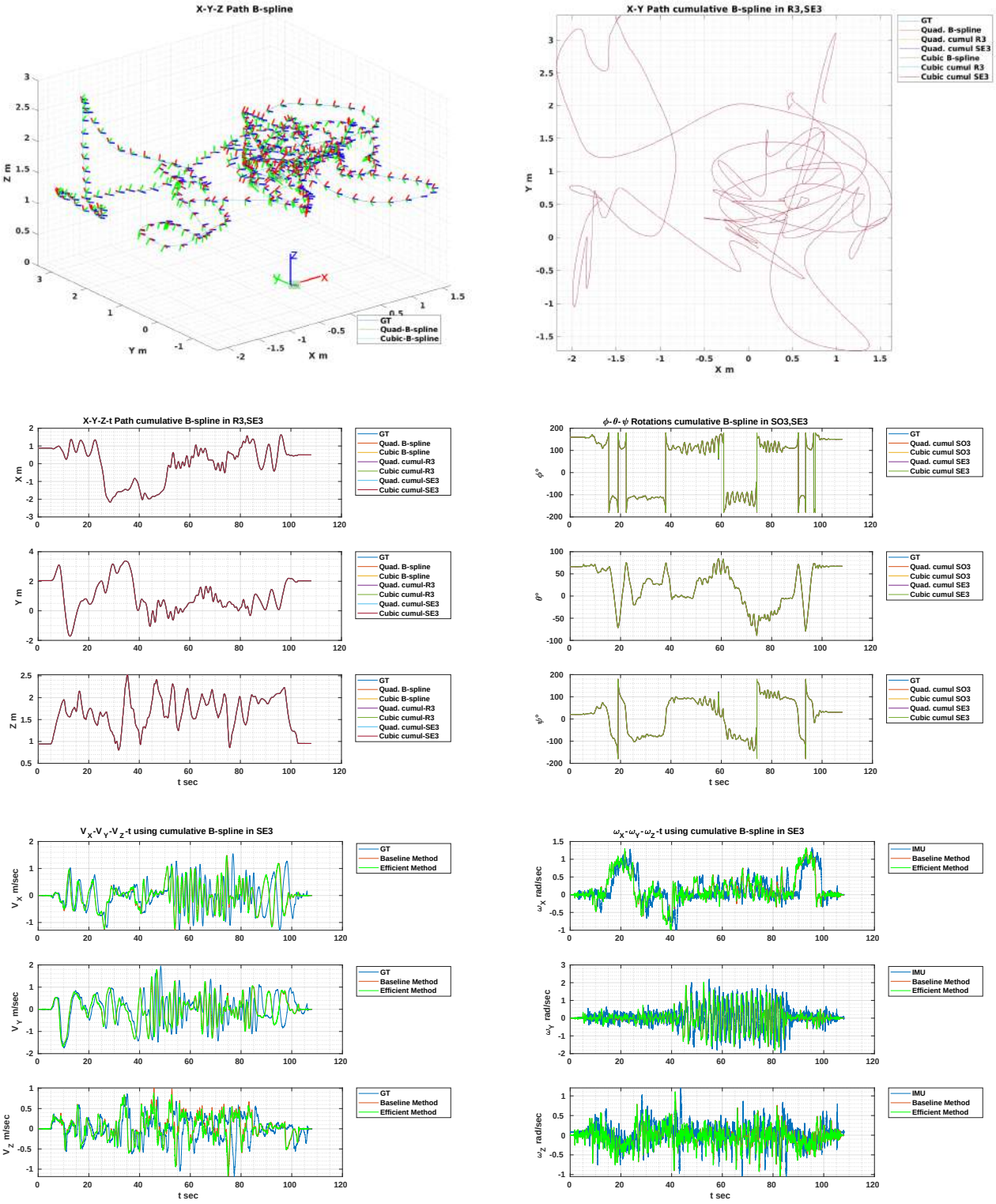


Figure B.5: Vicon room 1 Difficult: B-spline comparison in $\mathrm{R}(3), \mathrm{SE}(3)$

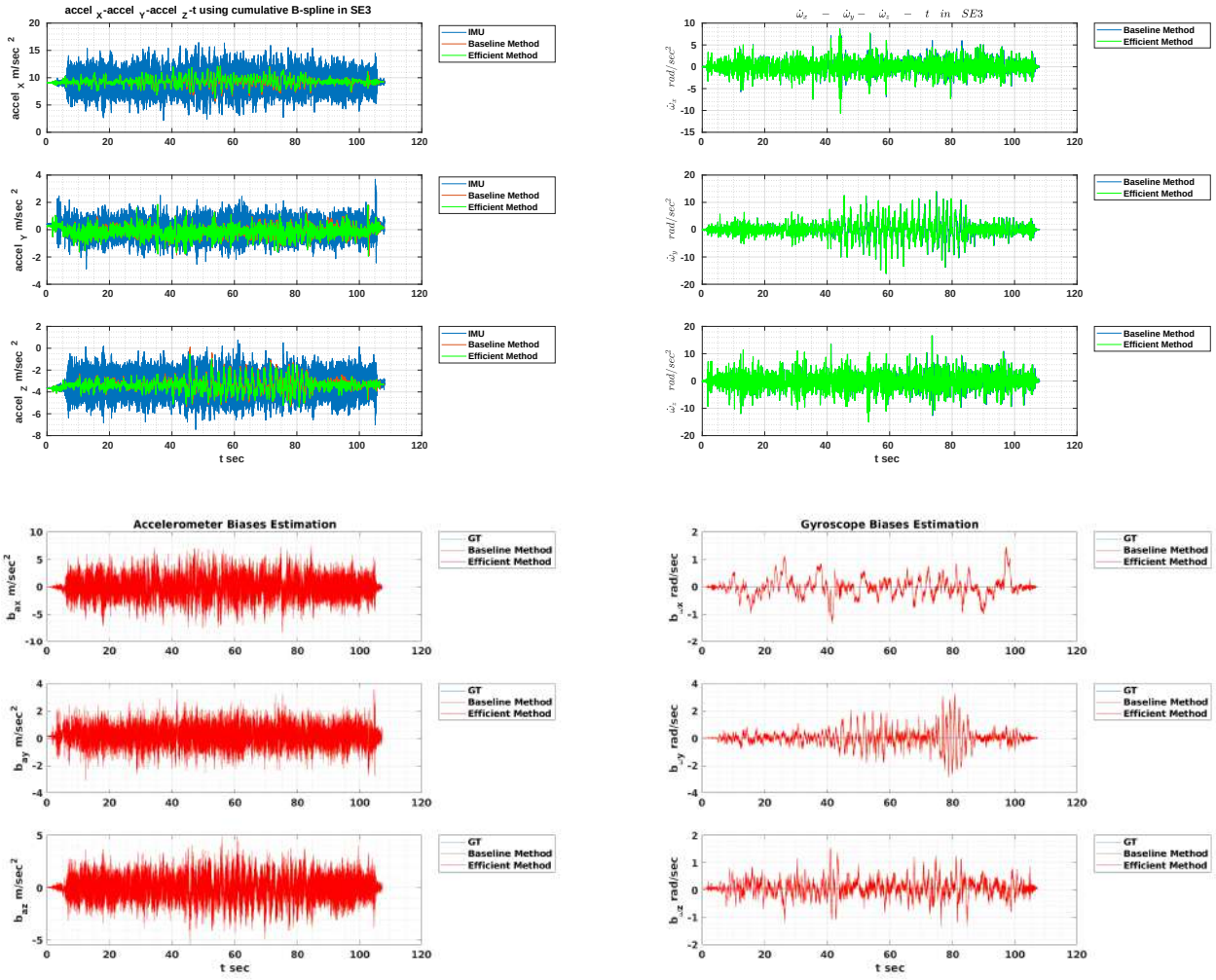


Figure B.6: Vicon room 1 Difficult: Baseline/Efficient/GT comparison

The IMU online calibration experiment on the V103-Difficult sequence shows high-precision accelerometer and gyroscope biases estimation based on the **Efficient** B-spline model compared to the baseline model using the Vicon measured trajectory.

B.2 Calibration results using EuRoC IMU and Optimizer as ground truth

B.2.1 EuRoC Dataset: Vicon room 1 “easy”

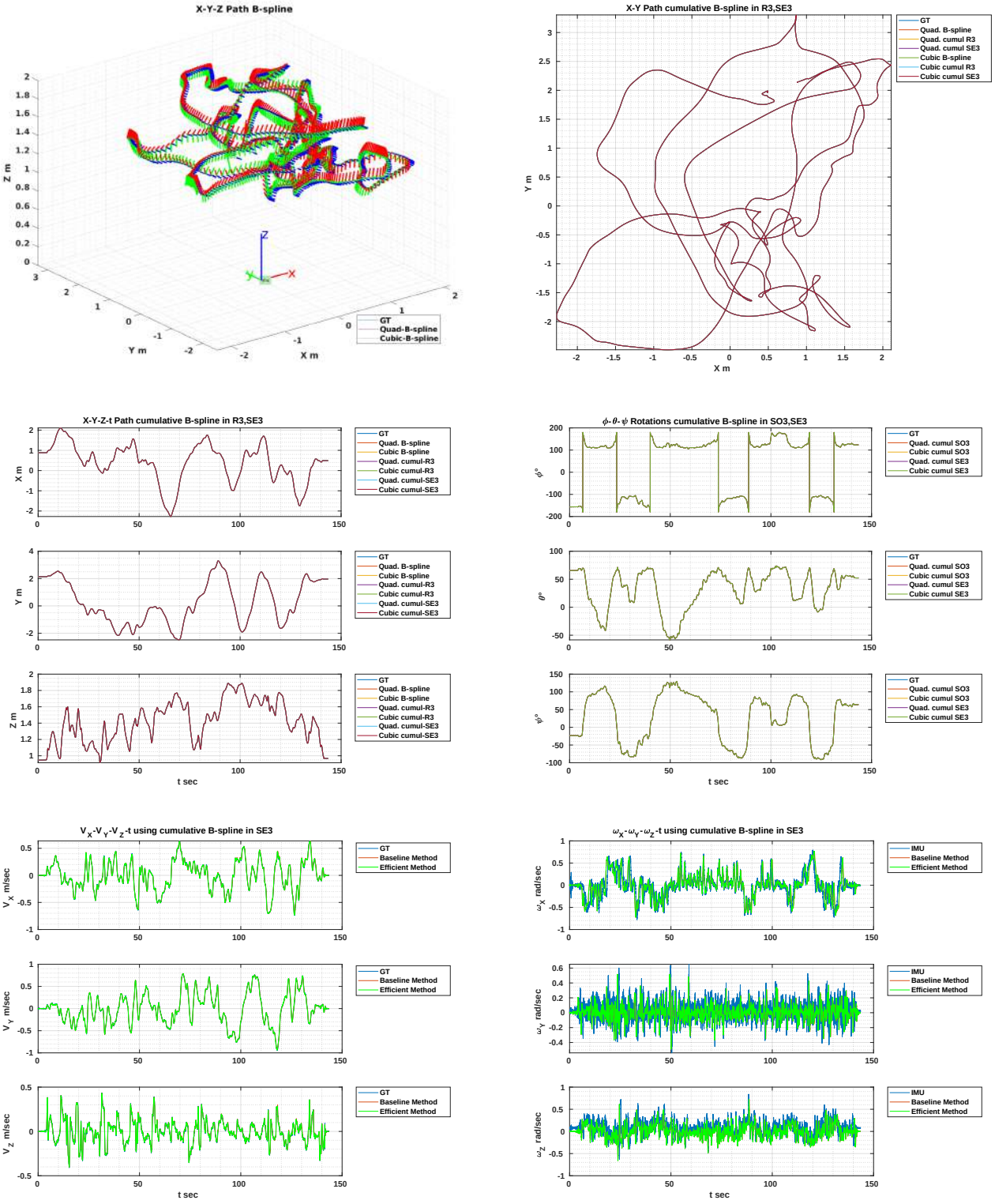


Figure B.7: Vicon room 1 Easy: B-spline comparison in $\mathrm{R}(3), \mathrm{SE}(3)$

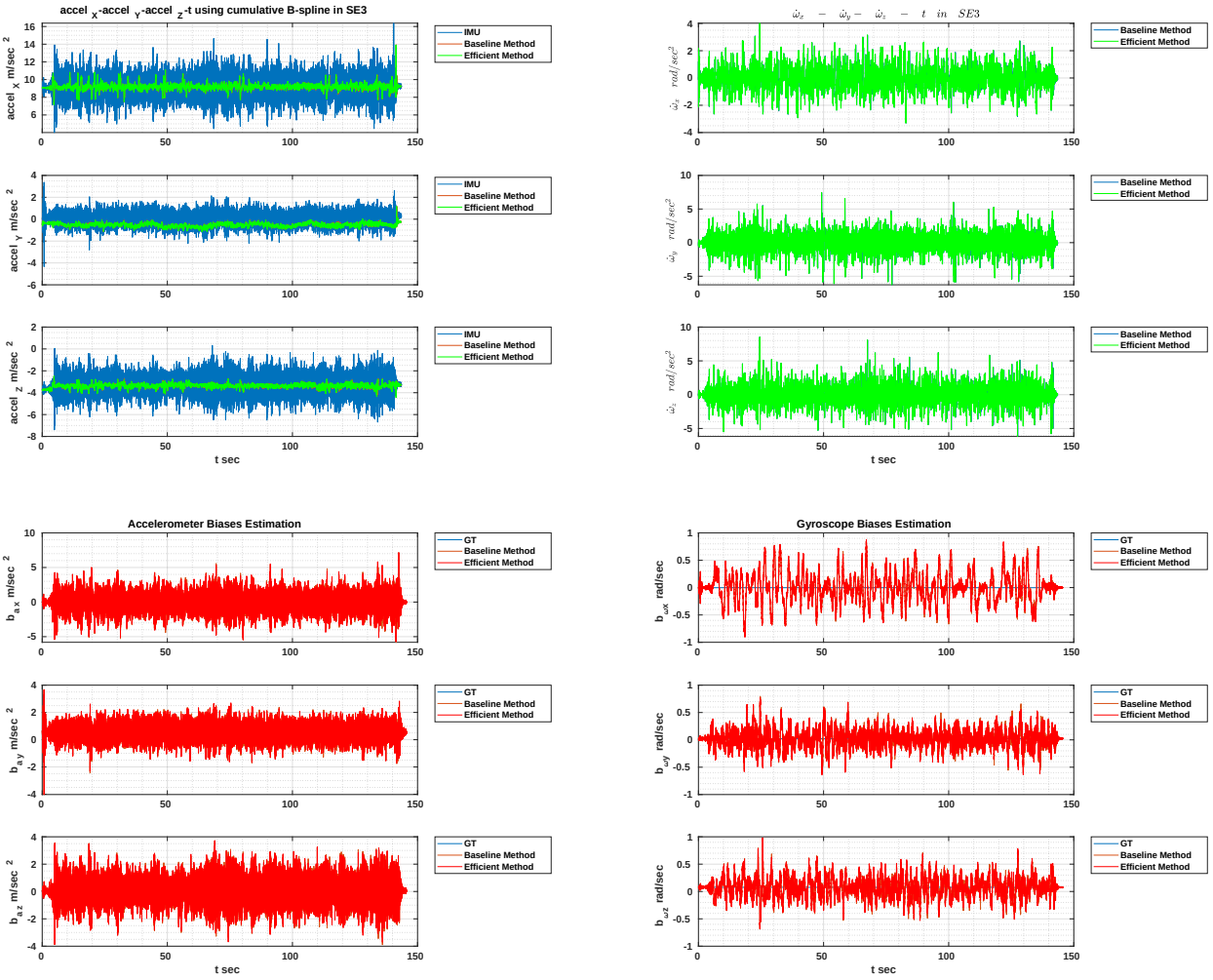


Figure B.8: Vicon room 1 Easy: Baseline/Efficient/GT comparison

The IMU online calibration experiment on the V101-Easy sequence shows high-precision accelerometer and gyroscope biases estimation based on the **Efficient** B-spline model compared to the baseline model using the EuRoC non-linearly optimized trajectory.

B.2.2 EuRoC Dataset: Vicon room 1 “medium”

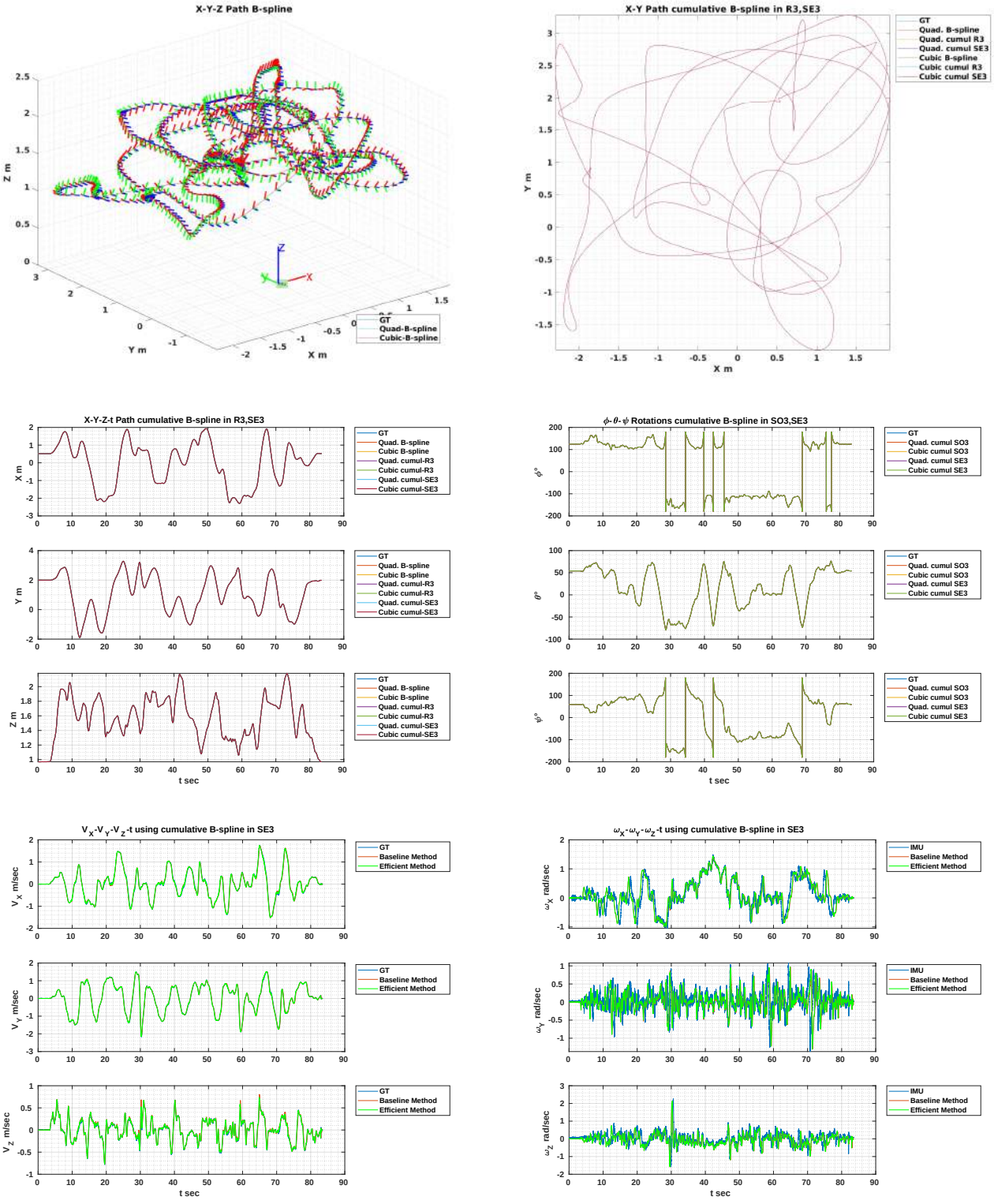


Figure B.9: Vicon room 1 Medium: B-spline comparison in $\mathrm{R}(3), \mathrm{SE}(3)$

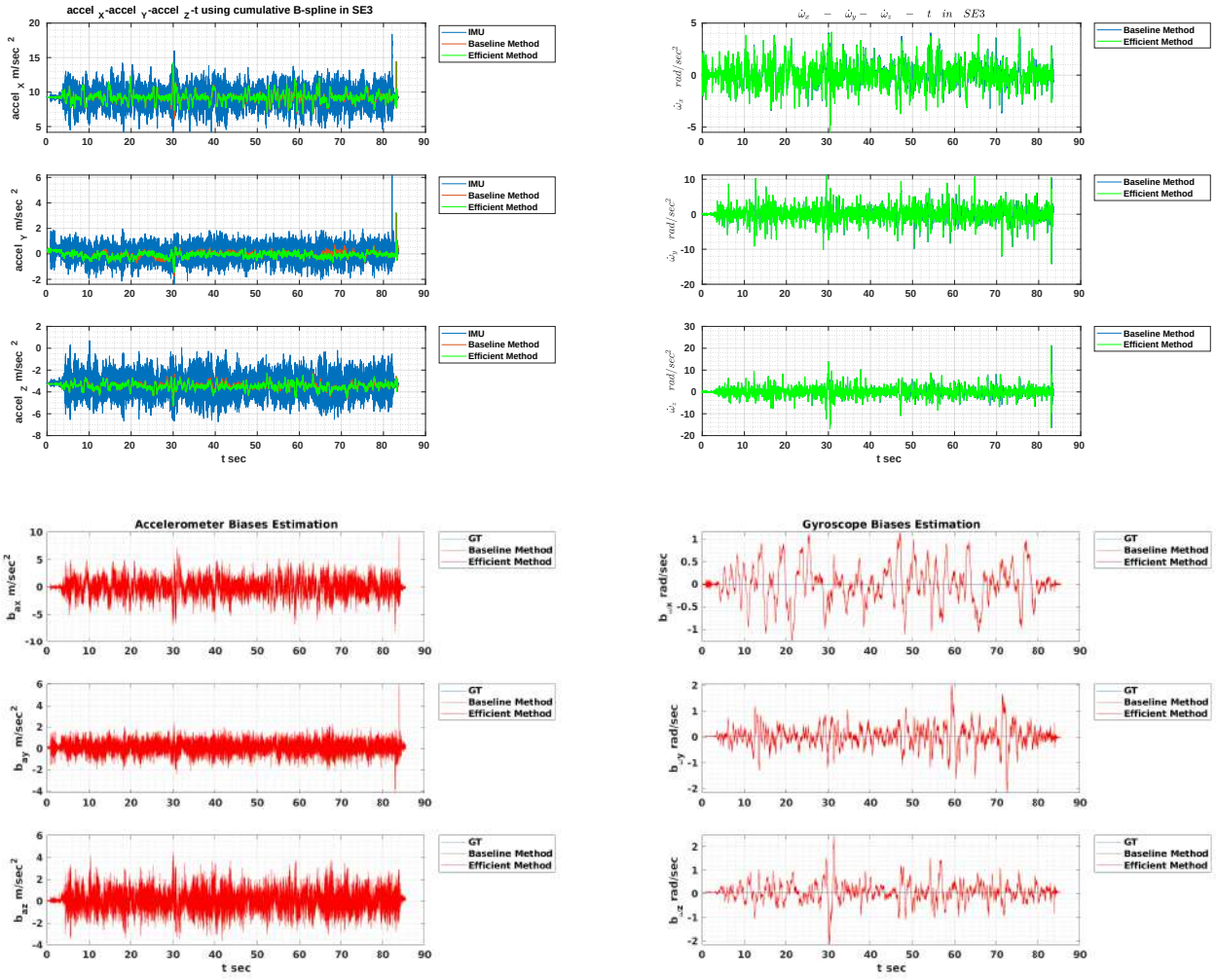


Figure B.10: Vicon room 1 Medium: Baseline/Efficient/GT comparison

The IMU online calibration experiment on the V102-Medium sequence shows high-precision accelerometer and gyroscope biases estimation based on the **Efficient** B-spline model compared to the baseline model using the EuRoC non-linearly optimized trajectory.

B.2.3 EuRoC Dataset: Vicon room 1 “difficult”

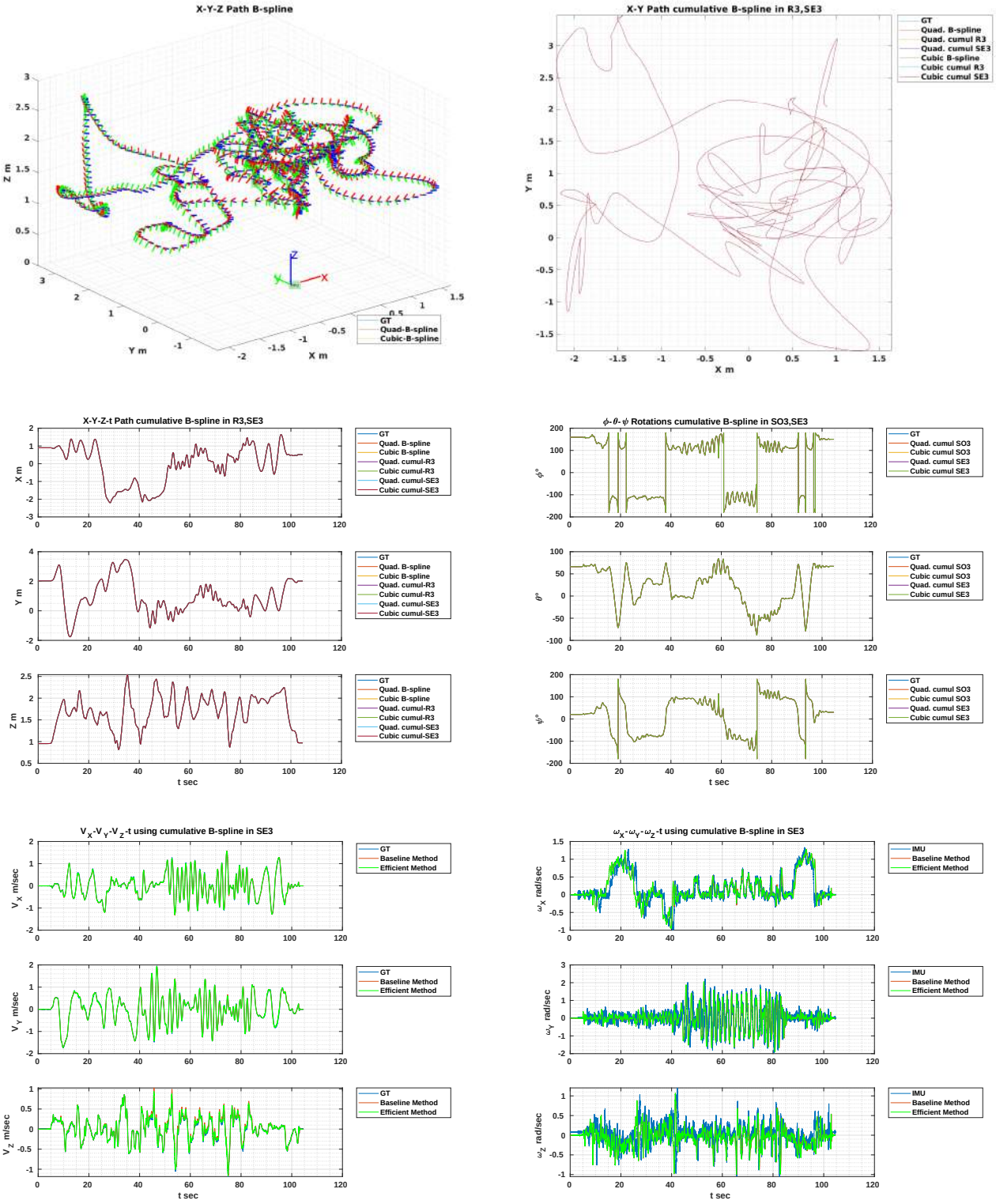


Figure B.11: Vicon room 1 Difficult: B-spline comparison in $\mathrm{R}(3), \mathrm{SE}(3)$

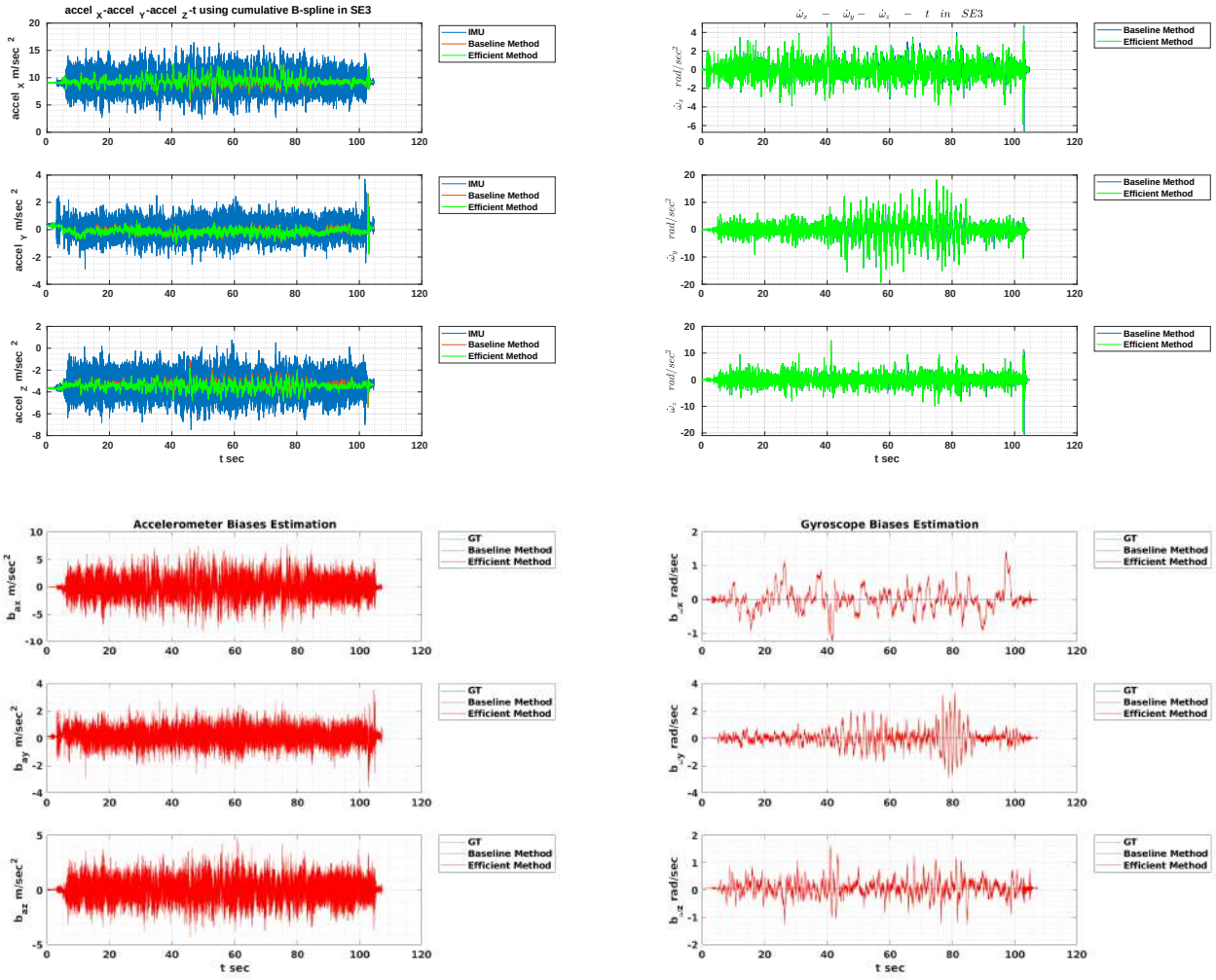


Figure B.12: Vicon room 1 Difficult: Baseline/Efficient/GT comparison

The IMU online calibration experiment on the V103-Difficult sequence shows high-precision accelerometer and gyroscope biases estimation based on the **Efficient** B-spline model compared to the baseline model using the EuRoC non-linearly optimized trajectory.

C

Q_d Derivation Equations

The Q_d in Equation (4.11) can be obtained after the consecutive matrix multiplications are performed using the following formulas. For simplicity, let $t = \Delta t$,$\sigma = d\sigma$,$\beta = -R_{(\hat{q_w^i})}$:

$$Q_{d_{11}} = \int_{\Delta t} (\sigma_{n_a}^2 . t^2 . \beta . \beta^\top + \sigma_{n_\omega}^2 . A . A^\top + \sigma_{n_{b_\omega}}^2 . B . B^\top + \sigma_{n_{b_a}}^2 . \frac{t^4}{4} . \beta . \beta^\top),$$

$$\begin{aligned} Q_{d_{11}} = \sigma_{n_a}^2 . \frac{t^3}{3} . \beta . \beta^\top + \sigma_{n_\omega}^2 [(\beta \lfloor \hat{a} \rfloor_\times)(I_{d_3} . \frac{t^5}{20} + \frac{t^7}{504} \lfloor \hat{\omega} \rfloor_\times^2)(\beta \lfloor \hat{a} \rfloor_\times)^\top] + \dots \\ + \sigma_{n_{b_\omega}}^2 [(\beta \lfloor \hat{a} \rfloor_\times)(I_{d_3} . \frac{t^7}{252} + \frac{t^9}{8640} \lfloor \hat{\omega} \rfloor_\times^2)(\beta \lfloor \hat{a} \rfloor_\times)^\top] + \sigma_{n_{b_a}}^2 . \frac{t^5}{20} . \beta . \beta^\top, \end{aligned}$$

$$Q_{d_{12}} = \int_{\Delta t} (\sigma_{n_a}^2 . t . \beta . \beta^\top + \sigma_{n_\omega}^2 . A . C^\top + \sigma_{n_{b_\omega}}^2 . B . D^\top + \sigma_{n_{b_a}}^2 . \frac{t^3}{2} . \beta . \beta^\top),$$

$$\begin{aligned} Q_{d_{12}} = \sigma_{n_a}^2 . \frac{t^2}{2} . \beta . \beta^\top + \sigma_{n_\omega}^2 [(\beta \lfloor \hat{a} \rfloor_\times)(I_{d_3} . \frac{t^4}{8} + \frac{t^5}{60} \lfloor \hat{\omega} \rfloor_\times + \frac{t^6}{144} \lfloor \hat{\omega} \rfloor_\times^2)(\beta \lfloor \hat{a} \rfloor_\times)^\top] + \dots \\ + \sigma_{n_{b_\omega}}^2 [(\beta \lfloor \hat{a} \rfloor_\times)(I_{d_3} . \frac{t^6}{72} + \frac{t^7}{1008} \lfloor \hat{\omega} \rfloor_\times + \frac{t^8}{1920} \lfloor \hat{\omega} \rfloor_\times^2)(\beta \lfloor \hat{a} \rfloor_\times)^\top] + \sigma_{n_{b_a}}^2 . \frac{t^4}{8} . \beta . \beta^\top, \end{aligned}$$

$$Q_{d_{13}} = \int_{\Delta t} (\sigma_{n_\omega}^2 . A . E^\top + \sigma_{n_{b_\omega}}^2 . B . F^\top),$$

$$\begin{aligned} Q_{d_{13}} = \sigma_{n_\omega}^2 [(\beta \lfloor \hat{a} \rfloor_\times)(I_{d_3} . \frac{t^3}{6} + \frac{t^4}{12} \lfloor \hat{\omega} \rfloor_\times + \frac{t^5}{40} \lfloor \hat{\omega} \rfloor_\times^2)] + \dots \\ + \sigma_{n_{b_\omega}}^2 [(\beta \lfloor \hat{a} \rfloor_\times)(I_{d_3} . \frac{t^5}{30} + \frac{t^6}{144} \lfloor \hat{\omega} \rfloor_\times + \frac{11 . t^7}{5040} \lfloor \hat{\omega} \rfloor_\times^2)], \end{aligned}$$

$$Q_{d_{14}} = \int_{\Delta t} (\sigma_{n_{b_\omega}}^2 . B) = \sigma_{n_{b_\omega}}^2 [(\beta \lfloor \hat{a} \rfloor_\times)(-I_{d_3} . \frac{t^4}{24} + \frac{t^5}{120} \lfloor \hat{\omega} \rfloor_\times - \frac{t^6}{720} \lfloor \hat{\omega} \rfloor_\times^2)],$$

$$Q_{d_{15}} = \int_{\Delta t} (\sigma_{n_{b_a}}^2 . \frac{t^2}{2} . \beta) = \sigma_{n_{b_a}}^2 . \frac{t^3}{6} . \beta,$$

$$Q_{d_{16}} = \int_{\Delta t}(0_{3\times 13}) = 0_{3\times 13},$$

$$Q_{d_{21}} = \int_{\Delta t}(\sigma^2_{n_a}.t.\beta.\beta^\top + \sigma^2_{n_\omega}.C.A^\top + \sigma^2_{n_{b_\omega}}.D.B^\top + \sigma^2_{n_{b_a}}.\frac{t^3}{2}.\beta.\beta^\top),$$

$$Q_{d_{21}} = \sigma^2_{n_a}.\frac{t^2}{2}.\beta.\beta^\top + \sigma^2_{n_\omega}[(\beta\lfloor\hat{a}\rfloor_\times)(I_{d_3}.\frac{t^4}{8} - \frac{t^5}{60}\lfloor\hat{\omega}\rfloor_\times + \frac{t^6}{144}\lfloor\hat{\omega}\rfloor^2_\times)(\beta\lfloor\hat{a}\rfloor_\times)^\top] + \dots$$
$$+\sigma^2_{n_{b_\omega}}[(\beta\lfloor\hat{a}\rfloor_\times)(I_{d_3}.\frac{t^6}{72} - \frac{t^7}{1008}\lfloor\hat{\omega}\rfloor_\times + \frac{t^8}{1920}\lfloor\hat{\omega}\rfloor^2_\times)(\beta\lfloor\hat{a}\rfloor_\times)^\top] + \sigma^2_{n_{b_a}}.\frac{t^4}{8}.\beta.\beta^\top,$$

$$Q_{d_{22}} = \int_{\Delta t}(\sigma^2_{n_a}.\beta.\beta^\top + \sigma^2_{n_\omega}.C.C^\top + \sigma^2_{n_{b_\omega}}.D.D^\top + \sigma^2_{n_{b_a}}.t^2.\beta.\beta^\top),$$

$$Q_{d_{22}} = \sigma^2_{n_a}.t.\beta.\beta^\top + \sigma^2_{n_\omega}[(\beta\lfloor\hat{a}\rfloor_\times)(I_{d_3}.\frac{t^3}{3} + \frac{t^5}{60}\lfloor\hat{\omega}\rfloor^2_\times)(\beta\lfloor\hat{a}\rfloor_\times)^\top] + \dots$$
$$+\sigma^2_{n_{b_\omega}}[(\beta\lfloor\hat{a}\rfloor_\times)(I_{d_3}.\frac{t^5}{20} + \frac{t^7}{504}\lfloor\hat{\omega}\rfloor^2_\times)(\beta\lfloor\hat{a}\rfloor_\times)^\top] + \sigma^2_{n_{b_a}}.\frac{t^3}{3}.\beta.\beta^\top,$$

$$Q_{d_{23}} = \int_{\Delta t}(\sigma^2_{n_\omega}.C.E^\top + \sigma^2_{n_{b_\omega}}.D.F^\top),$$

$$Q_{d_{23}} = \sigma^2_{n_\omega}[(\beta\lfloor\hat{a}\rfloor_\times)(I_{d_3}.\frac{t^2}{2} + \frac{t^3}{6}\lfloor\hat{\omega}\rfloor_\times + \frac{t^4}{24}\lfloor\hat{\omega}\rfloor^2_\times)] + \dots$$
$$+\sigma^2_{n_{b_\omega}}[(\beta\lfloor\hat{a}\rfloor_\times)(I_{d_3}.\frac{t^4}{8} + \frac{t^5}{60}\lfloor\hat{\omega}\rfloor_\times + \frac{t^6}{144}\lfloor\hat{\omega}\rfloor^2_\times)],$$

$$Q_{d_{24}} = \int_{\Delta t}(\sigma^2_{n_{b_\omega}}.D) = \sigma^2_{n_{b_\omega}}[-(\beta\lfloor\hat{a}\rfloor_\times)(I_{d_3}.\frac{t^3}{6} - \frac{t^4}{24}\lfloor\hat{\omega}\rfloor_\times + \frac{t^5}{120}\lfloor\hat{\omega}\rfloor^2_\times)],$$

$$Q_{d_{25}} = \int_{\Delta t}(\sigma^2_{n_{b_a}}.t.\beta) = \sigma^2_{n_{b_a}}.\frac{t^2}{2}.\beta,$$

$$Q_{d_{26}} = \int_{\Delta t}(0_{3\times 13}) = 0_{3\times 13},$$

$$Q_{d_{31}} = \int_{\Delta t}(\sigma^2_{n_\omega}.E.A^\top + \sigma^2_{n_{b_\omega}}.F.B^\top),$$

$$Q_{d_{31}} = \sigma^2_{n_\omega}[(I_{d_3}.\frac{t^3}{6} - \frac{t^4}{12}\lfloor\hat{\omega}\rfloor_\times + \frac{t^5}{40}\lfloor\hat{\omega}\rfloor^2_\times)(\beta\lfloor\hat{a}\rfloor_\times)^\top] + \dots$$
$$+\sigma^2_{n_{b_\omega}}[(I_{d_3}.\frac{t^5}{30} - \frac{t^6}{144}\lfloor\hat{\omega}\rfloor_\times + \frac{11.t^7}{5040}\lfloor\hat{\omega}\rfloor^2_\times)(\beta\lfloor\hat{a}\rfloor_\times)^\top],$$

$$Q_{d_{32}} = \int_{\Delta t}(\sigma^2_{n_\omega}.E.C^\top + \sigma^2_{n_{b_\omega}}.F.D^\top),$$

$$Q_{d_{32}} = \sigma^2_{n_\omega}[(I_{d_3}.\frac{t^2}{2} - \frac{t^3}{6}\lfloor\hat{\omega}\rfloor_\times + \frac{t^4}{24}\lfloor\hat{\omega}\rfloor^2_\times)(\beta\lfloor\hat{a}\rfloor_\times)^\top] + \dots$$
$$+\sigma^2_{n_{b_\omega}}[(I_{d_3}.\frac{t^4}{8} - \frac{t^5}{60}\lfloor\hat{\omega}\rfloor_\times + \frac{t^6}{144}\lfloor\hat{\omega}\rfloor^2_\times)(\beta\lfloor\hat{a}\rfloor_\times)^\top],$$

$$Q_{d_{33}} = \int_{\Delta t}(\sigma^2_{n_\omega}.E.E^\top + \sigma^2_{n_{b_\omega}}.F.F^\top),$$

$$Q_{d_{33}} = \sigma^2_{n_\omega}[(I_{d_3}.t)] + \sigma^2_{n_{b_\omega}}[(I_{d_3}.\frac{t^3}{3} + \frac{t^5}{60}\lfloor\hat{\omega}\rfloor^2_\times)],$$

$$Q_{d_{34}} = \int_{\Delta t}(\sigma^2_{n_{b_\omega}}.F) = \sigma^2_{n_{b_\omega}}[(I_{d_3}.\frac{-t^2}{2} + \frac{t^3}{6}\lfloor\hat{\omega}\rfloor_\times - \frac{t^4}{24}\lfloor\hat{\omega}\rfloor^2_\times)],$$

$$Q_{d_{35}} = \int_{\Delta t}(0_{3\times3}) = 0_{3\times3},$$

$$Q_{d_{36}} = \int_{\Delta t}(0_{3\times13}) = 0_{3\times13},$$

$$Q_{d_{41}} = \int_{\Delta t}(\sigma^2_{n_{b_\omega}}.B^\top) = \sigma^2_{n_{b_\omega}}[(\beta\lfloor\hat{a}\rfloor_\times)(-I_{d_3}.\frac{t^4}{24} + \frac{t^5}{120}\lfloor\hat{\omega}\rfloor_\times - \frac{t^6}{720}\lfloor\hat{\omega}\rfloor^2_\times)]^\top,$$

$$Q_{d_{42}} = \int_{\Delta t}(\sigma^2_{n_{b_\omega}}.D^\top) = \sigma^2_{n_{b_\omega}}[-(\beta\lfloor\hat{a}\rfloor_\times)(I_{d_3}.\frac{t^3}{6} - \frac{t^4}{24}\lfloor\hat{\omega}\rfloor_\times + \frac{t^5}{120}\lfloor\hat{\omega}\rfloor^2_\times)]^\top,$$

$$Q_{d_{43}} = \int_{\Delta t}(\sigma^2_{n_{b_\omega}}.F^\top) = \sigma^2_{n_{b_\omega}}[(I_{d_3}.\frac{-t^2}{2} + \frac{t^3}{6}\lfloor\hat{\omega}\rfloor_\times - \frac{t^4}{24}\lfloor\hat{\omega}\rfloor^2_\times)]^\top,$$

$$Q_{d_{44}} = \int_{\Delta t}(\sigma^2_{n_{b_\omega}}) = \sigma^2_{n_{b_\omega}}.t,$$

$$Q_{d_{45}} = \int_{\Delta t}(0_{3\times3}) = 0_{3\times3},$$

$$Q_{d_{46}} = \int_{\Delta t}(0_{3\times13}) = 0_{3\times13},$$

$$Q_{d_{51}} = \int_{\Delta t}(\sigma^2_{n_{b_a}}.\beta^\top.\frac{t^2}{2}) = \sigma^2_{n_{b_\omega}}.\beta^\top.\frac{t^3}{6},$$

$$Q_{d_{52}} = \int_{\Delta t}(\sigma^2_{n_{b_a}}.\beta^\top.t) = \sigma^2_{n_{b_\omega}}.\beta^\top.\frac{t^2}{2},$$

$$Q_{d_{53}} = \int_{\Delta t}(0_{3\times3}) = 0_{3\times3},$$

$$Q_{d_{54}} = \int_{\Delta t}(0_{3\times3}) = 0_{3\times3},$$

$$Q_{d_{55}} = \int_{\Delta t}(\sigma^2_{n_{b_a}}) = \sigma^2_{n_{b_a}}.t,$$

$$Q_{d_{56}} = \int_{\Delta t}(0_{3\times13}) = 0_{3\times13},$$

$$Q_{d_{61\to65}} = \textstyle\int_{\Delta t}(0_{13\times3}) = 0_{13\times3}, \quad Q_{d_{66}} = \int_{\Delta t}(0_{13\times13}) = 0_{13\times13}.$$

Bibliography

[1] A. Soliman, F. Bonardi, D. Sidibé, and S. Bouchafa, "IBISCape: A simulated benchmark for multi-modal SLAM systems evaluation in large-scale dynamic environments," *Journal of Intelligent & Robotic Systems*, vol. 106, no. 3, p. 53, Oct 2022. [Online]. Available: https://doi.org/10.1007/s10846-022-01753-7

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

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

[4] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2017.

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

[6] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 2100–2106.

[7] P. Alliez, F. Bonardi, S. Bouchafa, J.-Y. Didier, H. Hadj-Abdelkader, F. I. I. Muñoz, V. Kachurka, B. Rault, M. Robin, and D. Roussel, "Real-time multi-slam system for agent localization and 3d mapping in dynamic scenarios," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4894–4900, 2020.

[8] F. Caron, E. Duflos, D. Pomorski, and P. Vanheeghe, "Gps/imu data fusion using multisensor kalman filtering: introduction of contextual aspects," *Information fusion*, vol. 7, no. 2, pp. 221–230, 2006.

[9] Y. Yang, W. Lee, P. Osteen, P. Geneva, X. Zuo, and G. Huang, "icalib: Inertial aided multi-sensor calibration," pp. 1–7, 2021, iCRA - VINS Workshop 2021, Xi'an, China.

[10] J. PerŠić, L. Petrović, I. Marković, and I. Petrović, "Spatiotemporal multisensor calibration via gaussian processes moving target tracking," *IEEE Transactions on Robotics*, pp. 1–15, 2021.

[11] W. Lee, Y. Yang, and G. Huang, "Efficient multi-sensor aided inertial navigation with online calibration," pp. 5706–5712, 2021, 2021 IEEE International Conference on Robotics and Automation (ICRA).

[12] D. Gehrig, M. Rüegg, M. Gehrig, J. Hidalgo-Carrió, and D. Scaramuzza, "Combining events and frames using recurrent asynchronous multimodal networks for monocular depth prediction," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2822–2829, 2021.

[13] M. Gehrig, W. Aarents, D. Gehrig, and D. Scaramuzza, "DSEC: a stereo event camera dataset for driving scenarios," *IEEE Robotics and Automation Letters*, 2021.

[14] Y. Li, R. Yunus, N. Brasch, N. Navab, and F. Tombari, "Rgb-d slam with structural regularities," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11 581–11 587, 2021.

[15] C. Debeunne and D. Vivet, "A review of visual-lidar fusion based simultaneous localization and mapping," *Sensors*, vol. 20, no. 7, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/7/2068

[16] K. Minoda, F. Schilling, V. Wüest, D. Floreano, and T. Yairi, "Viode: A simulated dataset to address the challenges of visual-inertial odometry in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1343–1350, 2021.

[17] J.-E. Deschaud, D. Duque, J. P. Richa, S. Velasco-Forero, B. Marcotegui, and F. Goulette, "Paris-carla-3d: A real and synthetic outdoor point cloud dataset for challenging tasks in 3d mapping," *Remote Sensing*, vol. 13, no. 22, 2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/22/4713

[18] J.-E. Deschaud, "KITTI-CARLA: a KITTI-like dataset generated by CARLA Simulator," *arXiv e-prints*, 2021.

[19] A. R. Sekkat, Y. Dupuis, V. R. Kumar, H. Rashed, S. Yogamani, P. Vasseur, and P. Honeine, "Synwoodscape: Synthetic surround-view fisheye camera dataset for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8502–8509, 2022.

[20] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[21] D. Gehrig, A. Loquercio, K. G. Derpanis, and D. Scaramuzza, "End-to-end learning of representations for asynchronous event-based data," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5633–5643, 2019.

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

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

[24] J.-L. Blanco-Claraco, F.-A. Moreno-Duenas, and J. González-Jiménez, "The málaga urban dataset: High-rate stereo and lidar in a realistic urban scenario," *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 207–214, 2014.

[25] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of michigan north campus long-term vision and lidar dataset," *The International Journal of Robotics Research*, vol. 35, no. 9, pp. 1023–1035, 2016.

[26] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016. [Online]. Available: http://ijr.sagepub.com/content/early/2016/01/21/0278364915620033.abstract

[27] A. L. Majdik, C. Till, and D. Scaramuzza, "The zurich urban micro aerial vehicle dataset," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 269–273, 2017.

[28] B. Pfrommer, N. Sanket, K. Daniilidis, and J. Cleveland, "Penncosyvio: A challenging visual inertial odometry benchmark," IEEE, pp. 3847–3854, 2017, 2017 IEEE International Conference on Robotics and Automation (ICRA).

[29] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stueckler, and D. Cremers, "The tum vi benchmark for evaluating visual-inertial odometry," in *International Conference on Intelligent Robots and Systems (IROS)*, October 2018.

[30] K. M. Judd and J. D. Gammell, "The oxford multimotion dataset: Multiple se (3) motions with ground truth," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 800–807, 2019.

[31] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex urban dataset with multi-level sensors from highly diverse urban environments," *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 642–657, 2019.

[32] M. Kasper, S. McGuire, and C. Heckman, "A benchmark for visual-inertial odometry systems employing onboard illumination," IEEE, pp. 5256–5263, 2019, 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

[33] J. Delmerico, T. Cieslewski, H. Rebecq, M. Faessler, and D. Scaramuzza, "Are we ready for autonomous drone racing? the uzh-fpv drone racing dataset," IEEE, pp. 6713–6719, 2019, 2019 International Conference on Robotics and Automation (ICRA).

[34] D. Zuñiga-Noël, A. Jaenal, R. Gomez-Ojeda, and J. Gonzalez-Jimenez, "The uma-vi dataset: Visual–inertial odometry in low-textured and dynamic illumination environments," *The International Journal of Robotics Research*, vol. 39, no. 9, pp. 1052–1060, 2020.

[35] A. Antonini, W. Guerra, V. Murali, T. Sayre-McCord, and S. Karaman, "The blackbird uav dataset," *The International Journal of Robotics Research*, vol. 39, no. 10-11, pp. 1346–1364, 2020.

[36] H. Zhang, L. Jin, and C. Ye, "The VCU-RVI Benchmark: Evaluating Visual Inertial Odometry for Indoor Navigation Applications with an RGB-D Camera," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 6209–6214.

[37] S. Klenk, J. Chui, N. Demmel, and D. Cremers, "Tum-vie: The tum stereo visual-inertial event dataset," *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8601–8608, 2021.

[38] C. Yuan, J. Lai, P. Lyu, P. Shi, W. Zhao, and K. Huang, "A novel fault-tolerant navigation and positioning method with stereo-camera/micro electro mechanical systems inertial measurement unit (mems-imu) in hostile environment," *Micromachines*, vol. 9, p. 626, 11 2018.

[39] M. Faessler, F. Fontana, C. Forster, E. Mueggler, M. Pizzoli, and D. Scaramuzza, "Autonomous, vision-based flight and live dense 3d mapping with a quadrotor micro aerial vehicle," *Journal of Field Robotics*, vol. 33, no. 4, pp. 431–450, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21581

[40] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to mav navigation," pp. 3923–3929, 2013, 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems.

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

[42] 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)*. IEEE, 2015, pp. 298–304.

[43] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "OrbSLAM3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021. [Online]. Available: http://dx.doi.org/10.1109/TRO.2021.3075644

[44] V. Usenko, N. Demmel, D. Schubert, J. Stueckler, and D. Cremers, "Visual-inertial mapping with non-linear factor recovery," *IEEE Robotics and Automation Letters (RA-L) & Int. Conference on Intelligent Robotics and Automation (ICRA)*, vol. 5, no. 2, pp. 422–429, 2020.

[45] J. Delmerico and D. Scaramuzza, "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots," pp. 2502–2509, 2018, 2018 IEEE International Conference on Robotics and Automation (ICRA).

[46] Y. Zhou, G. Gallego, and S. Shen, "Event-based stereo visual odometry," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1433–1450, 2021.

[47] D. Gehrig, M. Gehrig, J. Hidalgo-Carrio, and D. Scaramuzza, "Video to events: Recycling video datasets for event cameras," *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 3583–3592, June 2020.

[48] H. Rebecq, G. Gallego, E. Mueggler, and D. Scaramuzza, "EMVS: Event-based multi-view stereo—3D reconstruction with an event camera in real-time," *Int. J. Comput. Vis.*, vol. 126, pp. 1394–1414, Dec. 2018.

[49] A. Tomy, A. Paigwar, K. S. Mann, A. Renzaglia, and C. Laugier, "Fusing Event-based and RGB camera for Robust Object Detection in Adverse Conditions," *ICRA 2022 - IEEE International Conference on Robotics and Automation*, May 2022. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03591717

[50] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "Events-to-video: Bringing modern computer vision to event cameras," *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019.

[51] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," *Robotics: Science and Systems*, 2014.

[52] Y. Pan, P. Xiao, Y. He, Z. Shao, and Z. Li, "Mulls: Versatile lidar slam via multi-metric linear least square," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11 633–11 640, 2021.

[53] 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)*. IEEE, 2016, pp. 4304–4311.

[54] M. Muglikar, M. Gehrig, D. Gehrig, and D. Scaramuzza, "How to calibrate your event camera," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1403–1409, 2021.

[55] L. Galleani and P. Tavella, "The dynamic allan variance," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 56, no. 3, pp. 450–464, 2009.

[56] C. Tomasi and T. Kanade, "Detection and tracking of point," *Int J Comput Vis*, vol. 9, pp. 137–154, 1991.

[57] 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, 2022. [Online]. Available: https://www.mdpi.com/2072-4292/14/13/3010

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

[59] Y. Zhou, G. Gallego, H. Rebecq, L. Kneip, H. Li, and D. Scaramuzza, "Semi-dense 3d reconstruction with a stereo event camera," *Proceedings of the European conference on computer vision (ECCV)*, pp. 235–251, 2018.

[60] D. Hug, P. Banninger, I. Alzugaray, and M. Chli, "Continuous-time stereo-inertial odometry," *IEEE Robotics and Automation Letters*, pp. 1–1, 2022.

[61] M. Chghaf, S. Rodriguez, and A. E. Ouardi, "Camera, LiDAR and multi-modal SLAM systems for autonomous ground vehicles: a survey," *Journal of Intelligent & Robotic Systems*, vol. 105, no. 1, pp. 1–35, 2022.

[62] Z. Chang, Y. Meng, W. Liu, H. Zhu, and L. Wang, "WiCapose: multi-modal fusion based transparent authentication in mobile environments," *Journal of Information Security and Applications*, vol. 66, p. 103130, 2022.

[63] K. Jung, S. Shin, and H. Myung, "U-VIO: Tightly Coupled UWB Visual Inertial Odometry for Robust Localization," in *International Conference on Robot Intelligence Technology and Applications*. Springer, 2022, pp. 272–283.

[64] D. Nistér, "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, 2004.

[65] A. Heyden and M. Pollefeys, "Multiple view geometry," *Emerging topics in computer vision*, vol. 90, pp. 180–189, 2005.

[66] P. Antonante, V. Tzoumas, H. Yang, and L. Carlone, "Outlier-robust estimation: Hardness, minimally tuned algorithms, and applications," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 281–301, 2021.

[67] D. Barath, J. Noskova, M. Ivashechkin, and J. Matas, "MAGSAC++, a fast, reliable and accurate robust estimator," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[68] A. Das, J. Elfring, and G. Dubbelman, "Real-time vehicle positioning and mapping using graph optimization," *Sensors*, vol. 21, no. 8, p. 2815, 2021.

[69] V. Usenko, N. Demmel, D. Schubert, J. Stückler, and D. Cremers, "Visual-inertial mapping with non-linear factor recovery," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 422–429, 2019.

[70] X. Xiao, Y. Zhang, H. Li, H. Wang, and B. Li, "Camera-IMU Extrinsic Calibration Quality Monitoring for Autonomous Ground Vehicles," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4614–4621, 2022.

[71] J. Huai, Y. Zhuang, Y. Lin, G. Jozkow, Q. Yuan, and D. Chen, "Continuous-time spatiotemporal calibration of a rolling shutter camera-IMU system," *IEEE Sensors Journal*, vol. 22, no. 8, pp. 7920–7930, 2022.

[72] Y. Zhang, W. Liang, S. Zhang, X. Yuan, X. Xia, J. Tan, and Z. Pang, "High-precision Calibration of Camera and IMU on Manipulator for Bio-inspired Robotic System," *Journal of Bionic Engineering*, vol. 19, no. 2, pp. 299–313, 2022.

[73] J. Lee, D. Hanley, and T. Bretl, "Extrinsic calibration of multiple inertial sensors from arbitrary trajectories," *IEEE Robotics and Automation Letters*, 2022.

[74] Y. Zhou, D. Chen, J. Wu, M. Huang, and Y. Weng, "Calibration of RGB-D camera using depth correction model," *Journal of Physics: Conference Series*, vol. 2203, no. 1, p. 012032, 2022.

[75] F. Basso, E. Menegatti, and A. Pretto, "Robust intrinsic and extrinsic calibration of RGB-D cameras," *IEEE Transactions on Robotics*, vol. 34, no. 5, pp. 1315–1332, 2018.

[76] W. Darwish, S. Tang, W. Li, and W. Chen, "A new calibration method for commercial RGB-D sensors," *Sensors*, vol. 17, no. 6, p. 1204, 2017.

[77] H. Liu, D. Qu, F. Xu, F. Zou, J. Song, and K. Jia, "Approach for accurate calibration of RGB-D cameras using spheres," *Opt. Express*, vol. 28, no. 13, pp. 19 058–19 073, Jun 2020. [Online]. Available: http://opg.optica.org/oe/abstract.cfm?URI=oe-28-13-19058

[78] A. Staranowicz, G. R. Brown, F. Morbidi, and G. L. Mariottini, "Easy-to-Use and accurate calibration of RGB-D cameras from spheres," in *Image and Video Technology*, R. Klette, M. Rivera, and S. Satoh, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 265–278.

[79] C. Chu and S. Yang, "Keyframe-based RGB-D visual-inertial odometry and camera extrinsic calibration using Extended Kalman Filter," *IEEE Sensors Journal*, vol. 20, no. 11, pp. 6130–6138, 2020.

[80] C. X. Guo and S. I. Roumeliotis, "IMU-RGBD camera 3D pose estimation and extrinsic calibration: Observability analysis and consistency improvement," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 2935–2942.

[81] J. C. Chow, D. D. Lichti, J. D. Hol, G. Bellusci, and H. Luinge, "IMU and multiple RGB-D camera fusion for assisting indoor stop-and-go 3D terrestrial laser scanning," *Robotics*, vol. 3, no. 3, pp. 247–280, 2014.

[82] H. Ovrén, P.-E. Forssén, and D. Törnqvist, "Why would I want a gyroscope on my RGB-D sensor?" in *2013 IEEE Workshop on Robot Vision (WORV)*. IEEE, 2013, pp. 68–75.

[83] W. Chai, C. Chen, and E. Edwan, "Enhanced indoor navigation using fusion of IMU and RGB-D camera," in *International Conference on Computer Information Systems and Industrial Applications*. Atlantis Press, 2015, pp. 547–549.

[84] N. Brunetto, S. Salti, N. Fioraio, T. Cavallari, and L. Stefano, "Fusion of inertial and visual measurements for RGB-D slam on mobile devices," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 1–9.

[85] T. Laidlow, M. Bloesch, W. Li, and S. Leutenegger, "Dense RGB-D-inertial SLAM with map deformations," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 6741–6748.

[86] Z. Shan, R. Li, and S. Schwertfeger, "RGBD-inertial trajectory estimation and mapping for ground robots," *Sensors*, vol. 19, no. 10, p. 2251, 2019.

[87] Y. Ling, H. Liu, X. Zhu, J. Jiang, and B. Liang, "RGB-D inertial odometry for indoor robot via Keyframe-based nonlinear optimization," in *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE, 2018, pp. 973–979.

[88] H. Zhang and C. Ye, "DUI-VIO: Depth uncertainty incorporated visual inertial odometry based on an RGB-D camera," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5002–5008.

[89] X. Zuo, N. Merrill, W. Li, Y. Liu, M. Pollefeys, and G. P. Huang, "CodeVIO: visual-inertial odometry with learned optimizable dense depth," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14 382–14 388, 2021.

[90] J. Surber, L. Teixeira, and M. Chli, "Robust visual-inertial localization with weak GPS priors for repetitive UAV flights," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 6300–6306.

[91] A. Patron-Perez, S. Lovegrove, and G. Sibley, "A spline-based trajectory representation for sensor fusion and rolling shutter cameras," *International Journal of Computer Vision*, vol. 113, no. 3, pp. 208–219, 2015.

[92] J. Solà, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," *CoRR*, vol. abs/1812.01537, 2018. [Online]. Available: http://arxiv.org/abs/1812.01537

[93] C. Sommer, V. Usenko, D. Schubert, N. Demmel, and D. Cremers, "Efficient derivative computation for cumulative b-splines on lie groups," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. IEEE, 2020, pp. 11 145–11 153. [Online]. Available: https://doi.org/10.1109/CVPR42600.2020.01116

[94] K. Shoemake, "Animating rotation with quaternion curves," in *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, 1985, pp. 245–254.

[95] ——, "Quaternion calculus and fast animation," in *Siggraph 87 course # 10: Computer Animation: 3 D Motion specification and control*. Siggraph, 1987.

[96] M.-J. Kim, M.-S. Kim, and S. Y. Shin, "Ac/sup 2/-continuous b-spline quaternion curve interpolating a given sequence of solid orientations," in *Proceedings Computer Animation'95*. IEEE, 1995, pp. 72–81.

[97] E. B. Dam, M. Koch, and M. Lillholm, *Quaternions, interpolation and animation*. Citeseer, 1998, vol. 2.

[98] P. Crouch, G. Kun, and F. S. Leite, "The de casteljau algorithm on lie groups and spheres," *Journal of Dynamical and Control Systems*, vol. 5, no. 3, pp. 397–429, 1999.

[99] C. De Boor, "On calculating with b-splines," *Journal of Approximation theory*, vol. 6, no. 1, pp. 50–62, 1972.

[100] J. Shi and Tomasi, "Good features to track," in *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.

[101] W. Darwish, W. Li, S. Tang, and W. Chen, "Coarse to fine global RGB-D frames registration for precise indoor 3D model reconstruction," in *2017 International Conference on Localization and GNSS (ICL-GNSS)*. IEEE, 2017, pp. 1–5.

[102] Y. Wang and G. S. Chirikjian, "Nonparametric second-order theory of error propagation on motion groups," *The International journal of robotics research*, vol. 27, no. 11-12, pp. 1258–1273, 2008.

[103] A. Nurhakim, N. Ismail, H. M. Saputra, and S. Uyun, "Modified fourth-order runge-kutta method based on trapezoid approach," in *2018 4th International Conference on Wireless and Telematics (ICWT)*, 2018, pp. 1–5.

[104] M. Z. Khairallah, A. Soliman, F. Bonardi, D. Roussel, and S. Bouchafa, "Flow-based visual-inertial odometry for neuromorphic vision sensors using non-linear optimization with online calibration," in *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP 2023, Volume 5: VISAPP, Lisbon, Portugal, February 19-21, 2023*, P. Radeva, G. M. Farinella, and K. Bouatouch, Eds. SCITEPRESS, 2023, pp. 963–973. [Online]. Available: https://doi.org/10.5220/0011660400003417

[105] R. Voges and B. Wagner, "Timestamp offset calibration for an IMU-Camera system under interval uncertainty," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 377–384.

[106] P. J. Huber, "Robust estimation of a location parameter," in *Breakthroughs in statistics*. Springer, 1992, pp. 492–518.

[107] M. J. Black and P. Anandan, "The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields," *Computer Vision and Image Understanding*, vol. 63, no. 1, pp. 75–104, 1996. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314296900065

[108] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[109] S. Agarwal, K. Mierle, and T. C. S. Team, "Ceres Solver," 3 2022. [Online]. Available: https://github.com/ceres-solver/ceres-solver

[110] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015. [Online]. Available: https://doi.org/10.1177/0278364914554813

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

[112] T. Qin, J. Pan, S. Cao, and S. Shen, "A general optimization-based framework for local odometry estimation with multiple sensors," *arXiv preprint arXiv:1901.03638*, 2019.

[113] 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)*. IEEE, 2020, pp. 1689–1696.

[114] G. Cioffi, T. Cieslewski, and D. Scaramuzza, "Continuous-time vs. discrete-time vision-based slam: A comparative study," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2399–2406, 2022.

[115] M. Obst, S. Bauer, P. Reisdorf, and G. Wanielik, "Multipath detection with 3D digital maps for robust multi-constellation gnss/ins vehicle localization in urban areas," in *2012 IEEE Intelligent Vehicles Symposium*, 2012, pp. 184–190.

[116] B. Dong and K. Zhang, "A tightly coupled visual-inertial gnss state estimator based on point-line feature," *Sensors*, vol. 22, no. 9, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/9/3391

[117] N. Gu, F. Xing, and Z. You, "Gnss spoofing detection based on coupled visual/inertial/gnss navigation system," *Sensors*, vol. 21, no. 20, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/20/6769

[118] W. Huang, W. Wan, and H. Liu, "Optimization-based online initialization and calibration of monocular visual-inertial odometry considering spatial-temporal constraints," *Sensors*, vol. 21, no. 8, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/8/2673

[119] S. Ma, X. Bai, Y. Wang, and R. Fang, "Robust stereo visual-inertial odometry using nonlinear optimization," *Sensors*, vol. 19, no. 17, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/17/3747

[120] S. Zhang, W. Wang, H. Li, and S. Zhang, "Evtracker: An event-driven spatiotemporal method for dynamic object tracking," *Sensors*, vol. 22, no. 16, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/16/6090

[121] G. Ren, Y. Yu, H. Liu, and T. Stathaki, "Dynamic knowledge distillation with noise elimination for rgb-d salient object detection," *Sensors*, vol. 22, no. 16, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/16/6188

[122] F. Alonge, P. Cusumano, F. D'Ippolito, G. Garraffa, P. Livreri, and A. Sferlazza, "Localization in structured environments with uwb devices without acceleration measurements, and velocity estimation using a kalman-bucy filter," *Sensors*, vol. 22, no. 16, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/16/6308

[123] S. Cao, H. Gao, and J. You, "In-flight alignment of integrated sins/gps/polarization/geomagnetic navigation system based on federal ukf," *Sensors*, vol. 22, no. 16, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/16/5985

[124] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, "Robust stereo visual inertial odometry for fast autonomous flight," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 965–972, 2018.

[125] Y. Yu, W. Gao, C. Liu, S. Shen, and M. Liu, "A gps-aided omnidirectional visual-inertial state estimator in ubiquitous environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 7750–7755.

[126] R. Mascaro, L. Teixeira, T. Hinzmann, R. Siegwart, and M. Chli, "Gomsf: Graph-optimization based multi-sensor fusion for robust uav pose estimation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1421–1428.

[127] G. Cioffi and D. Scaramuzza, "Tightly-coupled fusion of global positional measurements in optimization-based visual-inertial odometry," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5089–5095.

[128] J. Dai, S. Liu, X. Hao, Z. Ren, and X. Yang, "Uav localization algorithm based on factor graph optimization in complex scenes," *Sensors*, vol. 22, no. 15, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/15/5862

[129] M. Brossard, S. Bonnabel, and A. Barrau, "Unscented kalman filter on lie groups for visual inertial odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 649–655.

[130] A. Brunello, A. Urgolo, F. Pittino, A. Montvay, and A. Montanari, "Virtual sensing and sensors selection for efficient temperature monitoring in indoor environments," *Sensors*, vol. 21, no. 8, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/8/2728

[131] M. Schimmack, B. Haus, and P. Mercorelli, "An extended kalman filter as an observer in a control structure for health monitoring of a metal–polymer hybrid soft actuator," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 3, pp. 1477–1487, 2018.

[132] P. Mercorelli, "A switching kalman filter for sensorless control of a hybrid hydraulic piezo actuator using mpc for camless internal combustion engines," in *2012 IEEE International Conference on Control Applications*, 2012, pp. 980–985.

[133] G. Huang, M. Kaess, and J. J. Leonard, "Towards consistent visual-inertial navigation," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 4926–4933.

[134] P. Huang, H. Meyr, M. Dörpinghaus, and G. Fettweis, "Observability analysis of flight state estimation for uavs and experimental validation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4659–4665.

[135] M. Lv, H. Wei, X. Fu, W. Wang, and D. Zhou, "A loosely coupled extended kalman filter algorithm for agricultural scene-based multi-sensor fusion," *Frontiers in Plant Science*, vol. 13, 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fpls.2022.849260

[136] J. Sola, "Quaternion kinematics for the error-state kalman filter," *arXiv preprint arXiv:1711.02508*, 2017.

[137] N. Trawny and S. I. Roumeliotis, "Indirect kalman filter for 3d attitude estimation," *Citeseer*, 2005.

[138] P. Moulon, P. Monasse, and R. Marlet, "Adaptive structure from motion with a contrario model estimation," in *Computer Vision – ACCV 2012*, K. M. Lee, Y. Matsushita, J. M. Rehg, and Z. Hu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 257–270.

[139] M. Li and A. I. Mourikis, "High-precision, consistent ekf-based visual-inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013. [Online]. Available: https://doi.org/10.1177/0278364913481251

[140] A. Soliman, H. Hadj-Abdelkader, F. Bonardi, S. Bouchafa, and D. Sidibé, "MAV localization in large-scale environments: A decoupled optimization/filtering approach," *Sensors*, vol. 23, no. 1, 2023.

[141] Z. Xu, Z. Rong, and Y. Wu, "A survey: which features are required for dynamic visual simultaneous localization and mapping?" *Visual Computing for Industry, Biomedicine, and Art*, vol. 4, no. 1, pp. 1–16, 2021.

[142] Y. Almalioglu, M. Turan, N. Trigoni, and A. Markham, "Deep learning-based robust positioning for all-weather autonomous driving," *Nature Machine Intelligence*, vol. 4, no. 9, pp. 749–760, 2022.

[143] J. Hidalgo-Carrió, G. Gallego, and D. Scaramuzza, "Event-aided direct sparse odometry," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5781–5790.

[144] S. Sun, G. Cioffi, C. De Visser, and D. Scaramuzza, "Autonomous quadrotor flight despite rotor failure with onboard vision sensors: Frames vs. events," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 580–587, 2021.

[145] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[146] J. Revaud, C. De Souza, M. Humenberger, and P. Weinzaepfel, "R2d2: Reliable and repeatable detector and descriptor," *Advances in neural information processing systems*, vol. 32, 2019.

[147] L. Gao, Y. Liang, J. Yang, S. Wu, C. Wang, J. Chen, and L. Kneip, "VECtor: A versatile event-centric benchmark for multi-sensor slam," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8217–8224, 2022.

[148] S. Klenk, J. Chui, N. Demmel, and D. Cremers, "Tum-vie: The tum stereo visual-inertial event dataset," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8601–8608.

[149] A. Merzlyakov and S. Macenski, "A comparison of modern general-purpose visual slam approaches," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 9190–9197.

[150] Z. Teed and J. Deng, "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras," *Advances in Neural Information Processing Systems*, vol. 34, pp. 16 558–16 569, 2021.

[151] F. Nie, W. Zhang, Z. Yao, Y. Shi, F. Li, and Q. Huang, "Lcpf: A particle filter lidar slam system with loop detection and correction," *IEEE Access*, vol. 8, pp. 20 401–20 412, 2020.

[152] R. Jurevičius, V. Marcinkevičius, and J. Šeibokas, "Robust gnss-denied localization for uav using particle filter and visual odometry," *Machine Vision and Applications*, vol. 30, no. 7, pp. 1181–1190, 2019.

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

[154] P. Koniusz, F. Yan, and K. Mikolajczyk, "Comparison of mid-level feature coding approaches and pooling strategies in visual concept detection," *Computer vision and image understanding*, vol. 117, no. 5, pp. 479–492, 2013.

[155] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, October 2012.

[156] Z. Ji, F. Wang, X. Gao, L. Xu, and X. Hu, "Ssnet: Learning mid-level image representation using salient superpixel network," *Applied Sciences*, vol. 10, no. 1, 2020.

[157] H. Kim, S. Leutenegger, and A. J. Davison, "Real-time 3d reconstruction and 6-dof tracking with an event camera," in *European conference on computer vision*. Springer, 2016, pp. 349–364.

[158] H. Rebecq, T. Horstschäfer, G. Gallego, and D. Scaramuzza, "EVO: A geometric approach to event-based 6-dof parallel tracking and mapping in real time," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 593–600, 2016.

[159] B. Kueng, E. Mueggler, G. Gallego, and D. Scaramuzza, "Low-latency visual odometry using event-based feature tracks," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 16–23.

[160] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high-speed scenarios," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 994–1001, 2018.

[161] H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization," in *British Machine Vision Conference*, 2017.

[162] A. I. Maqueda, A. Loquercio, G. Gallego, N. García, and D. Scaramuzza, "Event-based vision meets deep learning on steering prediction for self-driving cars," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5419–5427.

[163] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Unsupervised event-based learning of optical flow, depth, and egomotion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 989–997.

[164] A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce, and R. Benosman, "Hats: Histograms of averaged time surfaces for robust event-based object classification," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1731–1740, 2018.

[165] D. Gehrig, H. Rebecq, G. Gallego, and D. Scaramuzza, "Eklt: Asynchronous photometric feature tracking using events and frames," *International Journal of Computer Vision*, vol. 128, no. 3, pp. 601–618, 2020.

[166] P. R. G. Cadena, Y. Qian, C. Wang, and M. Yang, "Spade-e2vid: Spatially-adaptive denormalization for event-based video reconstruction," *IEEE Transactions on Image Processing*, vol. 30, pp. 2488–2500, 2021.

[167] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240$\times$ 180 130 db 3 μs latency global shutter spatiotemporal vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, p. 2333–2341, 2014.

[168] T. Pire, T. Fischer, G. Castro, P. De Cristóforis, J. Civera, and J. J. Berlles, "S-ptam: Stereo parallel tracking and mapping," *Robotics and Autonomous Systems*, vol. 93, pp. 27–42, 2017.

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

[170] S. Ghosh and G. Gallego, "Multi-event-camera depth estimation and outlier rejection by refocused events fusion," *Advanced Intelligent Systems*, vol. 4, no. 12, pp. 220–241, 2022.

[171] D. Weikersdorfer, R. Hoffmann, and J. Conradt, "Simultaneous localization and mapping for event-based vision systems," in *Computer Vision Systems: 9th International Conference, ICVS 2013, St. Petersburg, Russia, July 16-18, 2013. Proceedings 9*. Springer, 2013, pp. 133–142.

[172] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, 2008.

[173] W. Guan, W. Li, and Y. Ren, "Point cloud registration based on improved icp algorithm," in *2018 Chinese Control And Decision Conference (CCDC)*, 2018, pp. 1461–1465.

[174] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.

[175] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, 2004, pp. I–I.

[176] S. Ahrens, D. Levine, G. Andrews, and J. P. How, "Vision-based guidance and control of a hovering vehicle in unknown, gps-denied environments," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 2643–2648.

[177] H. Alismail, L. D. Baker, and B. Browning, "Continuous trajectory estimation for 3d slam from actuated lidar," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, p. 6096–6101.

[178] S. Anderson, F. Dellaert, and T. D. Barfoot, "A hierarchical wavelet decomposition for continuous-time slam," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, p. 373–380.

[179] P. Corke, J. Lobo, and J. Dias, "An introduction to inertial and visual sensing," *I. J. Robotic Res.*, vol. 26, pp. 519–535, 06 2007.

[180] C. Bibby and I. Reid, "A hybrid slam representation for dynamic marine environments," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, p. 257–264.

[181] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 2010, pp. 2559–2566.

[182] D. C. Brown, "Close-range camera calibration," *Photogrammetric Eng.*, vol. 37, no. 8, pp. 855–866, 1971.

[183] W. Bulten, A. C. Van Rossum, and W. F. G. Haselager, "Human slam, indoor localisation of devices and users," in *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*, 2016, pp. 211–222.

[184] Y. Cao, N. Wang, C. Xiao, D. Yang, J. Fang, R. Yang, Q. A. Chen, M. Liu, and B. Li, "Invisible for both camera and lidar: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks," in *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021, pp. 176–194.

[185] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA sensors documentation reference," 2017. [Online]. Available: https://carla.readthedocs.io/en/latest/ref_sensors/

[186] L. Chen, A. Yang, H. Hu, and W. Naeem, "Rbpf-msis: Toward rao-blackwellized particle filter slam for autonomous underwater vehicle with slow mechanical scanning imaging sonar," *IEEE Systems Journal*, vol. 14, no. 3, pp. 3301–3312, 2019.

[187] J. Civera, O. G. Grasa, A. J. Davison, and J. Montiel, "1-point ransac for extended kalman filtering: Application to real-time structure from motion and visual odometry," *Journal of field robotics*, vol. 27, no. 5, pp. 609–631, 2010.

[188] D. Claus and A. W. Fitzgibbon, "A rational function lens distortion model for general cameras," in *Proc. CVPR*, 2005, pp. 213–219.

[189] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," pp. 3213–3223, 2016, proceedings of the IEEE conference on computer vision and pattern recognition.

[190] S. Cosar and N. Bellotto, "Human re-identification with a robot thermal camera using entropy-based sampling," *Journal of Intelligent & Robotic Systems*, 2019.

[191] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[192] F. Devernay and O. Faugeras, "Straight lines have to be straight," *MVA*, vol. 13, pp. 14–24, 2001.

[193] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[194] A. W. Fitzgibbon, "Simultaneous linear estimation of multiple view geometry and lens distortion," in *Proc. CVPR*, 2001.

[195] P. Furgale, T. D. Barfoot, and G. Sibley, "Continuous-time batch estimation using temporal basis functions," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, p. 2088–2095.

[196] P. Furgale, H. Sommer, J. Maye, J. Rehder, T. Schneider, and L. Oth, "Kalibr: A unified camera/imu calibration toolbox," 2014.

[197] G. Gallego and A. Yezzi, "A compact formula for the derivative of a 3-d rotation in exponential coordinates," *Journal of Mathematical Imaging and Vision*, vol. 51, no. 3, p. 378–384, 2015.

[198] G. Gallego, J. Lund, E. Mueggler, H. Rebecq, T. Delbruck, and D. Scaramuzza, "Event-based, 6-dof camera tracking from photometric depth maps," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, p. 1–1, 11 2017.

[199] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.

[200] E. Hong and J. Lim, "Visual-inertial odometry with robust initialization and online scale estimation," *Sensors*, vol. 18, no. 12, p. 4287, 2018.

[201] S. Hrabar, G. S. Sukhatme, P. Corke, K. Usher, and J. Roberts, "Combined optic-flow and stereo-based navigation of urban canyons for a uav," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 3309–3316.

[202] Z. Huang, L. Sun, C. Zhao, S. Li, and S. Su, "Eventpoint: Self-supervised interest point detection and description for event-based camera," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 5396–5405.

[203] E. S. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *The International Journal of Robotics Research*, vol. 30, no. 4, pp. 407–430, 2011.

[204] N. Kaygusuz, O. Mendez, and R. Bowden, "Multi-camera sensor fusion for visual odometry using deep uncertainty estimation," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 2944–2949.

[205] I. A. Kazerouni, L. Fitzgerald, G. Dooly, and D. Toal, "A survey of state-of-the-art on visual slam," *Expert Systems with Applications*, p. 117734, 2022.

[206] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *The International Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, 2011.

[207] C. Kerl, J. Stuckler, and D. Cremers, "Dense continuous-time tracking and mapping with rolling shutter rgb-d cameras," in *Proceedings of the IEEE international conference on computer vision*, 2015, p. 2264–2272.

[208] G. Klein and D. Murray, "Parallel tracking and mapping on a camera phone," in *2009 8th IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 2009, pp. 83–86.

[209] H.-A. Le, T. Mensink, P. Das, S. Karaoglu, and T. Gevers, "Eden: Multimodal synthetic dataset of enclosed garden scenes," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2021, pp. 1579–1589.

[210] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An invitation to 3-d vision: from images to geometric models*. Springer Science & Business Media, 2012, vol. 26.

[211] A. Macario Barros, M. Michel, Y. Moline, G. Corre, and F. Carrel, "A comprehensive survey of visual slam algorithms," *Robotics*, vol. 11, no. 1, p. 24, 2022.

[212] P. S. Maybeck, *Stochastic models, estimation, and control*, ser. Mathematics in Science and Engineering. ACADEMIC PRESS, INC. (LONDON)LTD., 1979, vol. 141.

[213] P. Meer, D. Mintz, A. Rosenfeld, and D. Y. Kim, "Robust regression methods for computer vision: A review," *International journal of computer vision*, vol. 6, no. 1, pp. 59–70, 1991.

[214] N. D. Mermin, "What's wrong with these equations?" *Physics Today*, Oct. 1989, http://www.cvpr.org/doc/mermin.pdf.

[215] Y. Mo, Y. Wu, X. Yang, F. Liu, and Y. Liao, "Review the state-of-the-art technologies of semantic segmentation based on deep learning," *Neurocomputing*, vol. 493, pp. 626–646, 2022.

[216] S. Mohammadnejad, M. Nasiri, S. Roshani, and S. Roshani, "A novel fixed pattern noise reduction technique in image sensors for satellite applications," in *Electrical and Electronic Engineering, Vol. 2 No. 5, pp. 271-276. doi: 10.5923/j.eee.20120205.05*, 2012.

[217] A. I. Mourikis, N. Trawny, S. I. Roumeliotis, A. E. Johnson, A. Ansar, and L. Matthies, "Vision-aided inertial navigation for spacecraft entry, descent, and landing," *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 264–280, 2009.

[218] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam," *The International Journal of Robotics Research*, vol. 36, no. 2, p. 142–149, 2017.

[219] E. Mueggler, G. Gallego, H. Rebecq, and D. Scaramuzza, "Continuous-time visual-inertial odometry for event cameras," *IEEE Transactions on Robotics*, vol. 34, no. 6, p. 1425–1440, 2018.

[220] 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*. IEEE, 2011, pp. 2320–2327.

[221] V. V. Nieuwenhove, J. D. Beenhouwer, F. D. Carlo, L. Mancini, F. Marone, and J. Sijbers, "Dynamic intensity normalization using eigen flat fields in x-ray imaging," *Opt. Express*, vol. 23, no. 21, p. 27975–27989, Oct 2015. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-23-21-27975

[222] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[223] F. Ou, Y. Li, and Z. Miao, "Place recognition of large-scale unstructured orchards with attention score maps," *IEEE Robotics and Automation Letters*, 2023.

[224] B. Pain, T. Cunningham, B. Hancock, G. Yang, S. Seshadri, and M. Ortiz, "Reset noise suppression in two-dimensional cmos photodiode pixels through column-based feedback-reset," in *Digest. International Electron Devices Meeting,*. IEEE, 2002, p. 809–812.

[225] A. Proctor and E. Johnson, "Vision-only aircraft flight control methods and test results," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2004, p. 5351.

[226] P. L. Rosin, Y.-K. Lai, L. Shao, and Y. Liu, *RGB-D Image Analysis and Processing*. Springer, 2019.

[227] L. Schmarje, M. Santarossa, S.-M. Schröder, and R. Koch, "A survey on semi-, self-and unsupervised learning for image classification," *IEEE Access*, vol. 9, pp. 82 146–82 168, 2021.

[228] S. Shiba, Y. Aoki, and G. Gallego, "Secrets of event-based optical flow," *European Conference on Computer Vision (ECCV)*, 2022.

[229] I. R. Spremolla, M. Antunes, D. Aouada, and B. E. Ottersten, "Rgb-d and thermal sensor fusion-application in person tracking." in *VISIGRAPP (3: VISAPP)*, 2016, p. 612–619.

[230] R. Swaminathan and S. Nayar, "Nonmetric calibration of wide-angle lenses and polycameras," *IEEE T-PAMI*, vol. 22, no. 10, pp. 1172–1178, 2000.

[231] H. Taheri and Z. C. Xia, "Slam; definition and evolution," *Engineering Applications of Artificial Intelligence*, vol. 97, p. 104032, 2021.

[232] D. Talwar and S. Jung, "Particle filter-based localization of a mobile robot by using a single lidar sensor under slam in ros environment," in *2019 19th International Conference on Control, Automation and Systems (ICCAS)*. IEEE, 2019, pp. 1112–1115.

[233] T. Toczek, F. Hamdi, B. Heyrman, J. Dubois, J. Miteran, and D. Ginhac, "Scene-based non-uniformity correction: from algorithm to implementation on a smart camera," *Journal of Systems Architecture*, vol. 59, no. 10, p. 833–846, 2013.

[234] Y. R. Tsai, "An efficient and accurate camera calibration technique for 3D machine vision," in *Proc. CVPR*, 1986.

[235] S. Weiss and R. Siegwart, "Real-time metric state estimation for modular vision-inertial systems," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 4531–4537.

[236] S. M. Weiss, "Vision based navigation for micro helicopters," Ph.D. dissertation, ETH Zurich, 2012.

[237] C. Yuan, X. Liu, X. Hong, and F. Zhang, "Pixel-level extrinsic self calibration of high resolution lidar and camera in targetless environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7517–7524, 2021.

[238] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," *Digital Signal Processing*, p. 103514, 2022.

[239] Y. Zhang, A. Carballo, H. Yang, and K. Takeda, "Perception and sensing for autonomous vehicles under adverse weather conditions: A survey," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 196, pp. 146–177, 2023.

[240] Z. Zhang, "On the epipolar geometry between two images with lens distortion," in *Proc. ICPR*, 1996, pp. 407–411.

[241] L. Zheng and X. Zhang, "Chapter 8 - numerical methods," in *Modeling and Analysis of Modern Fluid Problems*, ser. Mathematics in Science and Engineering, L. Zheng and X. Zhang, Eds. Academic Press, 2017, pp. 361–455. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128117538000086

[242] S. Zingg, D. Scaramuzza, S. Weiss, and R. Siegwart, "Mav navigation through indoor corridors using optical flow," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 3361–3368.