WANTING XU

NOVEL SOLUTIONS TO CHALLENGING RELATIVE POSE PROBLEMS FOR BOTH FRAME-BASED AND NEUROMORPHIC CAMERAS

DISS. SHANGHAITECH

NOVEL SOLUTIONS TO CHALLENGING RELATIVE POSE PROBLEMS FOR BOTH FRAME-BASED AND NEUROMORPHIC CAMERAS

A dissertation submitted to attain the degree of

DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE AND TECHNOLOGY of SHANGHAITECH UNIVERSITY
(Dr. Phil. ShanghaiTech University)

presented by

WANTING XU

born on 10 July 1997
citizen of China

accepted on the recommendation of

Prof. Dr. Sören Schwertfeger
Prof. Dr. Javier Civera
Prof. Dr. Zuzana Kukelova
Prof. Dr. Yujiao Shi

2024

To my family

ABSTRACT

This thesis reviews the latest research progress and challenges in vision-based localization, with a special focus on novel solutions to the challenging problem of relative pose estimation for frame-based and neuromorphic cameras. Despite the clear advantages of vision-based localization in terms of cost-effectiveness, miniaturization, and adaptability, it still faces challenges in terms of processing well under low-light conditions, processing blurred images, and improving real-time capabilities and robustness. The thesis discusses key technological trends in pose estimation such as the integration with deep learning, applications of event cameras, multi-sensor fusion, as well as research in visual-inertial odometry, velocity estimation, and line-based SLAM. It also looks forward to future research directions for relative pose estimation. These relative pose problems are mostly analyzed in the context of the SLAM problem. The thesis concludes with a discussion of possible remaining issues to be addressed in relative pose estimation, both with a view onto the SLAM problem and within a wider context.

The thesis first introduces the latest advancements in generalized relative pose estimation, proposing for the first time a fast and certifiably globally optimal solution through convex optimization. The main contributions include a novel formulation for estimating the generalized essential matrix and its relationship with existing eigenvalue-based methods, the globally optimal solution of the original formulation achieved through Semidefinite Relaxation (SDR) techniques, and the sufficient and necessary conditions for recovering the optimal Generalized Essential Matrix (GEM) from the relaxed problem. Furthermore, for the generalized camera relative pose and scale problems, the thesis presents a computationally efficient, accurate, and robust solution. This includes the first closed-form solution using 26 point correspondences, a new solver that requires only 9 correspondences, the simplification of correspondence requirements using affine covariant feature detectors, and a minimal solver based on 2 affine correspondences (AC) suitable for computations in scenarios with given directional correspondence between view graphs.

Due to the complexity of the generalized camera relative pose and scale problem, although we have proposed linear solutions and minimal solutions given directional correspondence between view graphs, finding minimal solutions under full degrees of freedom remains challenging. Generally,

minimal solutions require solving a system of polynomial equations, necessitating the use of Gröbner basis methods from algebraic geometry. Therefore, this thesis explored how to improve the numerical stability of Gröbner basis solvers through simple permutation selection. First, it was proven that in common problem categories within geometric vision, simple variable reordering can lead to accuracy differences in the application of the same elimination templates. Secondly, it was demonstrated how to train classifiers using original coefficient information to select a more accurate permutation online at a lower computational cost.

The entire first part of the thesis is devoted to research on pose estimation methods for normal cameras. In particular, it develops novel algorithms for generalized camera relative pose estimation, an important topic in many areas including SLAM, self-driving cars and intelligence augmentation devices. While important progress has been made, normal cameras however still face challenges in highly dynamic or difficult illumination scenarios. The second part of the thesis feeds into this line of research, and proposes novel geometric pose estimation methods for event cameras. The thesis introduces a new solver in the context of autonomous driving, and proposes to use a continuous-time model based on the nonholonomic Ackermann motion model, in combination with a single event camera for monocular motion estimation on a ground vehicle. The reduced motion model plus a few manipulations elegantly transform the problem into a univariate optimization problem. In addition, event cameras were fused with inertial sensors, innovatively developing a line-based visual-inertial 3D velocity estimator. The thesis introduce a novel two-layer RANSAC strategy for processing event data, perform geometric and event-based velocity initialization, and proposed a complete solution that includes initialization and a sliding window backend optimizer.

ACKNOWLEDGEMENTS

Reflecting on my PhD journey, it's true that I encountered numerous challenges and difficult moments. However, thanks to my perseverance and the warm companionship of those around me, I was able to overcome these obstacles and grow step by step, leading me to where I am today.

First and foremost, I must mention my supervisor, Laurent Kneip, to whom I wish to extend my sincerest gratitude. It was he who provided me the opportunity to study within one of the leading groups in computer vision and robotics, and to transition from a student with a background in mathematics to a PhD in computer vision SLAM. I am grateful for his warmth, patience and sincerity. Every time I encountered difficulties in my research, he would encourage me without hesitation, enabling me to persevere and offering incredibly valuable and insightful guidance. It is my good fortune to have such an excellent supervisor.

To my collaborators, I am deeply grateful to Xin Peng for introducing me to the field of event camera research. I thank her for her continuous support, which laid the groundwork for my subsequent work. It is an honor to have such an outstanding collaborator and friend like you. I appreciate Li Cui, as conversations and outings with you, whether about research or life, are always a joy. Thanks to Yifan (Haikun) Zuo, who is so considerate of others, being around you is always refreshing, and I am grateful for the times we have shared through the lows. Thank you to Xinyue Zhang, although our time together was short, it was memorable. Deep conversations beyond research made me believe that you are a person morally upright. Within the realm of research, your effort, diligence, and proactiveness have left a lasting impression on me. Thanks to Si'ao Zhang for your trust and tolerance, allowing me to guide your undergraduate thesis, and to Jiahang Wu, for the journey we've shared in collaboration. Working with such talented and positive students like you has been a great honor for me.

Additionally, I'd like to thank Lan Hu and Ji Zhao for the guidance they provided me at the beginning of my graduate studies. I also want to express my gratitude to my colleagues at the Mobile Perception Lab (MPL), including Zhanpeng Ouyang, Kun Huang, Yifu Wang, Yuchen Cao, Peng Wu, Ling Gao, Jiaxin Wei, Runze Yuan, Hang Su, Jiaqi Yang, Tao Liu, and Zijia Dai. I am thankful for their sharing and the time we spent together.

During my PhD, I am grateful for the support of my advisor, Laurent Kneip, which enabled me to have the opportunity to spend a year at ETH Zurich under the guidance of Professor Marc Pollefeys and postdoctoral researcher Daniel Barath at CVG for an exchange visit. This year greatly broadened my horizons, both academically and in life. I am thankful to Marc for his approachability and excellent academic taste, which left a deep impression on me. Thanks to Daniel for his patience, friendliness, and precise guidance, which were incredibly beneficial to me. I also want to thank the group of intelligent, hardworking, and outstanding individuals I met at CVG.

Special thanks to Yidan Gao, Zihan Zhu, and Yiming Zhao. With you, my life in the D-lab basement became so much more enjoyable, and my year in Zurich was no longer lonely. Thank you for your warmth and understanding. The deep conversations in the D-lab, the "too good to go" meals from Royal Panda by Lake Zurich, and every meal in chopsticks and picnic barbecue we had will all become cherished memories.

Finally, I must express my deepest gratitude to my good friend, Yingling. Throughout my PhD years, you have been by my side, always providing great comfort whenever I faced setbacks or disappointments. I am so fortunate to have you. With you, joy is doubled and pain dissipates. I also want to thank my family, whose love has been the driving force that keeps me going. Thank you for your kindness, your constant understanding, and your respect and support for my choices, which have shaped who I am today.

CONTENTS

1

INTRODUCTION

1.1 MOTIVATION

Simultaneous Localization and Mapping (SLAM) technology occupies a central position in the fields of robotics and automation. SLAM aims to enable a robot to navigate in an unknown environment while simultaneously constructing a map of that environment. This technology can be divided into two main categories: laser SLAM and visual SLAM. Laser SLAM uses laser scanners (such as LiDAR) to measure the distances around the environment, using this information to create a two-dimensional or three-dimensional map of the environment. Laser SLAM, with its high accuracy and robustness, has been widely applied in fields such as autonomous driving vehicles and industrial automation. Visual SLAM, on the other hand, utilizes cameras as the primary sensors. It estimates the motion of the robot by analyzing a continuous sequence of images, while simultaneously constructing a three-dimensional model of the environment.

Visual SLAM, with its cost-effectiveness, provision of rich environmental information, compactness and portability, strong adaptability, and lower energy consumption, has become a powerful complement and alternative to laser SLAM. Visual SLAM is also the main subject of research in this paper. It primarily relies on cameras to gather environmental information, which not only reduces the system cost, making it easier for large-scale deployment, but also captures rich visual details including colors and textures, significantly enhancing the quality of the map and the depth of environmental understanding. Furthermore, the miniaturization of cameras allows for the easy integration of visual SLAM systems into small or mobile devices such as smartphones, drones, and wearable devices, and it exhibits outstanding adaptability in environments where LiDAR may not be as effective, such as indoors. The relatively lower energy consumption also makes visual SLAM more practical for use in battery-powered mobile devices, helping to extend the device's usage time. These features reveal the broad application potential of visual SLAM in numerous fields such as augmented/virtual reality (AR/VR), autonomous driving vehicles, drone navigation, robotic navigation and task execution, and the preservation of historical monuments and buildings.

1.2 RESEARCH CHALLENGES

Despite the numerous advantages of visual SLAM, it still faces some challenges, such as degraded performance in low-light conditions, dealing with blurred images caused by high-speed movement, and improving the real-time capabilities and robustness of the algorithms. Future research directions include enhancing the accuracy and efficiency of visual SLAM systems, extending their application in complex environments, and utilizing advanced technologies such as deep learning to address current challenges. Pose estimation and multi-view geometry are core technical issues, which are crucial for the system's accuracy, efficiency, and robustness. In response to these aspects, several key research directions and technological trends are emerging.

- **Solving Difficult Pose Estimation Problems**: When facing complex pose estimation challenges, such as with generalized camera models and the generalized camera and scale problem, traditional methods may encounter performance bottlenecks. This necessitates the development of new algorithms, such as by constructing convex functions to approximate the original problem, enabling the optimization problem to be solved efficiently and accurately. Moreover, utilizing more information from point correspondences and developing new solutions for polynomial equation systems can effectively enhance the accuracy and robustness of pose estimation.

- **Combining Deep Learning with Traditional Geometric Algorithms**: Integrating deep learning with traditional geometric algorithms, such as with algebraic geometry, aids in solving complex polynomial equation problems and enhancing stability. This provides new perspectives for solving pose estimation issues in visual SLAM.

- **Application of Event Cameras in Pose Estimation**: Event cameras offer a novel method of visual information acquisition, providing more accurate and timely visual feedback for fast-moving objects and environments with high dynamic range. Therefore, in pose estimation, event cameras can enhance the performance of SLAM systems under conditions of high-speed movement and complex lighting.

- **Multi-Sensor Fusion Enhances Multi-View Geometry and Pose Estimation**: Multi-sensor fusion technology integrates data from various sensors such as cameras, IMUs, GPS, and event cameras, providing

a more comprehensive environmental understanding for the visual SLAM system. This fusion not only improves the accuracy of pose estimation but also ensures the robustness of multi-view geometry processing when visual information is insufficient or of low quality, especially demonstrating its value in complex environments.

1.3 LITERATURE REVIEW

1.3.1 *Generalized Camera Pose Estimation*

Using a non-central camera rig has attracted much attention from both academic and industrial communities. The most common case is that of a set of cameras—often with non-overlapping views—attached to a headset, micro air vehicle (MAV) or ground vehicle. Which is particularly relevant for real-time visual localization [1, 2] and autonomous driving [3]. the thesis will separately introduce the related work based on Point Correspondence (PC) and that dependent on Affine Correspondence (AC).

Relative Pose of a Central Camera: Regarding PC, essential or fundamental matrix estimation by algebraic error minimization has been extensively studied in previous literature [4–7]. For both the essential and the fundamental matrix, pose estimation by algebraic error minimization can be formulated as a polynomial optimization problem [8]. A polynomial optimization problem can be reformulated as a quadratically constrained quadratic program (QCQP), which has numerous off-the-shelf solvers. In multiple view geometry, semidefinite relaxation (SDR) for polynomial optimization problems was first studied by Kahl and Henrion in [9]. Recent work [7, 10] has successfully applied it to globally optimal, non-minimal central relative pose computation. For AC, an AC provides 3 independent constraints on the relative pose [11–13]. Thus, two ACs are enough to determine the relative pose for a pair of views [12–14]. In [15], guidelines are proposed for the effective usage of ACs within a model estimation pipeline. Further customizations have been developed for known vertical direction considering known global scales [16], planar motion [16, 17], or known feature depths [18].

Relative Pose of a Generalized Camera: For PC, the minimal solver is based on algebraic geometry, and uses 6 correspondences in order to come up with 64 solutions [19]. However, its large elimination template leads to poor numerical stability. Kim *et al.* later proposed alternative approaches for relative displacement estimation with non-overlapping multi-camera

systems using second-order cone programming (SOCP) [20] or branch-and-bound over the space of all rotations [21]. [1] furthermore derived a 5+1 point algorithm, and [22] proposed the antipodal epipolar constraint. A minimal solution for the case of non-holonomic motion was proposed in [3]. Minimal solutions for motions with a common direction were proposed in [23, 24]. An eigenvalue-based formulation for GEM estimation together with efficient local optimization was proposed in [25]. Very recently, another local optimization method for GEM estimation was proposed using an alternating minimization method [26]. The use of ACs to estimate the relative pose of multi-camera systems has recently drawn significant attention. [27] proposes a linear solver to recover the 6-DoF relative pose using 6 ACs, generalizing the 17 PCs solver proposed in [28]. [29] uses a first-order approximation to the relative rotation to estimate the relative pose with 2 ACs, which generalizes the 6 PCs solver proposed in [30]. They assume that the relative rotation of the multi-camera systems is small. Furthermore, [31] estimates the 3-DoF relative pose under planar motion from a single AC, as well as the 4-DoF pose with known vertical direction from 2 ACs. [32] focuses on the full 6-DoF relative pose problem of multi-camera systems from the minimum number of 2 ACs and not using any pose priors.

Generalized Relative Pose and Scale: There is a further generalization of GEM estimation, the *generalized relative pose and scale* problem. It introduces a further unknown: a relative scale factor between the ray origins in both generalized camera frames. It has an important application in structure from motion with central cameras. The generalized relative pose and scale problem was first introduced and solved for known vertical direction in [33], where the authors propose a minimal solver based on 5 point correspondences (PC) after the vertical directions have been aligned. The assumption of known vertical direction is dropped in [34], which solves for a full similarity transformation in a non-minimal, optimization-based manner. Both methods solve the generalized relative pose and scale problem from point correspondences only. The above studies on the relative pose of central and generalized cameras based on ACs all show that the characteristics of ACs help to construct accurate and efficient algorithms, thus proving that the application of ACs in the generalized relative pose and scale problem is very promising.

Therefore, the thesis present the latest advancements in generalized relative pose solutions based on PC in Chapter 2, and the latest advancements in generalized relative pose and scale problem solutions based on AC in Chapter 3.

1.3.2 *Solving Polynomial Equations in Pose Estimation*

The most commonly used method for solving polynomial equation systems in pose estimation is the Gröbner basis method. The Gröbner basis theory largely relies on the original work of Bruno Buchberger [35], who introduced the well-known *Buchberger algorithm* for the computation of a Gröbner basis. Good descriptions of the material can be found in Cox et al.'s introductions to algebraic geometry [36, 37]. One of the pioneering works employing a Gröbner basis solver in computer vision is presented by Stewenius et al. [38], who apply the technique to derive a closed-form solution to the calibrated generalized relative pose problem. While the application of the Gröbner basis method originally involves a manual search for the elimination template, a major breakthrough has been achieved by Kukelova's work on automatic solver generation [39]. The method has since been used exhaustively to solve both absolute [40, 41] and relative camera pose estimation problems [38, 42]. The method has furthermore been employed to solve a large variety of more specialized solvers that for example consider the partially uncalibrated or planar case [43], directional correspondences [44], or even special geometric arrangements such as two intersecting lines [45]. Gröbner basis solvers are important as they utilize a minimal set of points, and thus benefit robust hypothesis and test schemes.

Recent years have shown a number of works aiming at an improvement of solver efficiency with respect to the original solvers generated by Kukelova et al.'s toolbox [39]. For example, Bujnak et al.'s main contribution in [46] consists of using a modification of *FGLM* [47] to transform a *grevlex* Gröbner basis into a lexicographical one. Solutions to the latter can notably be recovered by efficiently finding the roots of a univariate polynomial. Kukelova et al. [48] later present an improvement of the reduction of the actual elimination template by exploiting the fact that it often has a sparse, block-diagonal structure. Further improvements are possible in situations in which there is a p-fold symmetry in the variety [49] or in which the ideals are saturated [50]. The most recent advancements that directly address the automatic solver generation issue are presented by Larsson et al. They first present an improved automatic solver generator [51], and then explore Gröbner fans for a variety of basis choices or even a random sampling scheme of linearly independent monomials in the quotient ring [52]. Depending on the chosen basis monomials, the solver will notably employ different ideal generators from the elimination template, which potentially leads to improved accuracy or even computational efficiency. The latter

contribution is related to our work in that it acknowledges the existence of multiple possible elimination templates. However, similar to most prior art, the choice of the basis and the resulting elimination template is fixed offline at solver generation stage, which limits the flexibility of the approach. One notable exception that is highly related to our work is presented by Byröd et al. [53]. It introduces online strategies for an improved construction of the action matrix.

To the best of our knowledge, the work in Chapter 4 is the first to exploit variable permutations and permutation-invariant polynomial forms to change the behavior of an elimination template at online stage. Furthermore, while neural networks have been recently used to learn permutations (e.g. visual permutation learning [54]), to the best of our knowledge, the thesis are the first to combine deep learning and algebraic geometry, and devise an automatically trained classifier for efficient online selection of a suitable permutation in the context of polynomial solving.

1.3.3 *Pose Estimation for Event Cameras*

The past two decades have seen the development of a large body of visual SLAM solutions. Seminal sparse keypoint based methods have been introduced by Klein et al. [55], Mul-Artal et al. [56, 57], Campos et al. [58], and Qin et al. [59]. More recently, the community has also presented geometric solutions to the semi-dense [60] or dense case [61–63]. However, these methods all rely on traditional cameras, for which both the establishment of correspondences as well as the geometry are well-understood problems to which many learning and non-learning based solutions exist. Event-based vision is not at a similar level of maturity, and currently still lacks behind even in the development of fundamental geometric pose solvers. A thorough investigation of event-based vision is provided in the survey of Gallego et al. [64] or via online resources [65].

Various solvers for 2D point correspondence-based motion estimation have been proposed based on the epipolar constraint, such as the eight-point solver [66], the seven-point solver [67], the six-point solver [68], or—in the minimal case—the five-point solver [69, 70]. Various specialized solvers for more constrained scenarios have been presented as well, such as solvers for a known directional correspondence [71], or a solver for planar motion [72]. In the case of non-holonomic planar motion (i.e. motion adhering to the Ackermann steering model), the problem reduces to only two degrees of freedom. As presented by Scaramuzza et al. [73–75], a single

correspondence is enough to recover the solution in this case. In [73], a scale-invariant solution is presented. In [74, 75], a known displacement away from the non-steering axis is used to additionally recover scale. The non-holonomic Ackermann motion model has also been explored for multi-camera systems [76], or even articulated multi-perspective cameras [77]. Perhaps most closely related to the present work is the method by Huang et al. [78], who present a planar tri-focal tensor based [72, 79] n-linear [67] solution able to utilize n measurements of a single line or point captured during a constant velocity arc of a circle in order to accurately determine the non-holonomic motion of the camera. Huang et al. [80] furthermore introduce a continuous non-holonomic trajectory model for use in back-end optimization. The methods listed here are limited to a traditional constant-framerate sensor operating under mild conditions, extension for event cameras is urgently needed.

However, the fundamentally different, asynchronous nature of event streams makes it difficult to directly use traditional geometric constraints from multiple-view geometry. Instead of full 6-Dof SLAM systems, most original works on event-based motion estimation therefore focus on simpler scenarios. Weikersdorfer et. al. [81] originally propose a 2D-SLAM system with a dynamic vision sensor by employing a particle filter. The same group also proposes an event-based 3D SLAM framework by fusing events with a classical frame-based RGB-D camera [82]. Other event-based visual odometry systems make use of known depth or 3D structure [83–87], or are simply limited to the pure rotation scenario [88]. Contrast maximization [88, 89] is proposed as a unifying framework applicable to several event-based vision tasks. Although it has garnered significant attention from researchers in the field [90–92], its applicability is currently restricted to homographic warping scenarios. Full 6-DoF estimation is solved by Kim et al. [93] using a filtering approach, and Rebecq et al. [94] use an alternating tracking and mapping framework. Zhu et al. [95] and Rebecq et al. [96] furthermore propose more reliable frameworks by fusing the measurements with an IMU. Furthermore, Mueggler et. al. [97] leverage continuous-time representations and spline-based trajectory optimization to perform visual-inertial odometry with an event camera.

More reliable 6-DoF odometry and SLAM solutions keep being obtained by fusion with other sensors. Kueng et al. [90] combine the event camera with a standard camera to track features and build a probabilistic map. A similar sensor combination is used in Ultimate-SLAM [98], which improves robustness and accuracy by a combined minimization of both vision and

event-based residual errors. Zhou et al. [99] propose the first event-based stereo odometry system. Zuo et al. [100] propose the use of a hybrid stereo setup of an event and a depth camera to realize DEVO, a semi-dense edge-tracking method inspired by Canny-VO [101], and Hidalgo-Carrió et al. [102] introduce EDS, a 6-DOF monocular direct visual odometry which combines events and frames.

Event-based motion estimation can be divided into optimization-based [94, 97, 103], filter-based [81, 104] and learning-based [105, 106] solutions. However, there is a lack of research on how fundamental geometry can be applied to event-based vision.

Therefore, Chapter 5, first include a ground vehicle motion model into single event camera motion tracking and thereby achieve robustness and computational efficiency superior to the existing, more general 6 DoF monocular solutions. Chapter 6 leverage our previous result [107] on applying trifocal tensor geometry [67] to explain the relationship between the events generated by a 3D line feature observation and the ego-motion of an event camera. This work extend our previous result by a more reliable direct solution of the camera velocity, and furthermore propose a novel visual-inertial fusion back-end to achieve reliable, velocity estimation. The back-end is similar to the line-based visual-inertial odometry framework IDOL [103], except that this work directly perform dynamics estimation in camera-centric coordinates, a more intuitive and fail-safe approach that does not depend on stable global map tracking.

1.3.4 *Visual-Inertial Odometry and Velometer*

The thesis provide an overview of the most important state-of-the-art contributions on visual-inertial odometry with standard cameras as well as direct visual-inertial speed estimation. Note that the field is very broad, and that the present overview mostly introduces recent, state-of-the-art contributions employing traditional geometric concepts. For a more complete overview including modern data-driven methods, the reader is kindly referred to [108].

The state-of-the-art solution to monocular real-time motion and structure estimation consists of fusing the visual measurements with an inertial measurement unit (IMU). The latter is composed of an accelerometer and a gyroscope to measure body accelerations and angular velocities. IMUs provide highly complementary information to visual readings, and thus have contributed substantially to the current robustness of visual-inertial

localization and mapping [108]. Most importantly, IMUs add metric scale to the otherwise scale invariant results from monocular SLAM. The preferred fusion strategy is tightly-coupled, and was initially demonstrated in seminal works by Sterlow and Singh [109], Dong-Si and Mourikis [110], and Mourikis et al. [111]. The first two works propose the first optimization-based visual-inertial fusion techniques (the latter one including fixed-lag smoothing). The third work proposes MSCKF, a popular EKF-based visual-inertial odometry system which updates IMU error states alongside camera poses. Based on the findings of Strasdat et al. [112], optimization-based methods have become the go-to strategy for visual-inertial fusion. Furthermore, the works of Lupton and Sukkarieh [113] and Forster et al. [114] have introduced IMU pre-integration terms, thus avoiding repeated integrations during optimization. These findings have nowadays led to the state-of-the-art optimization-based visual-inertial fusion frameworks ORB-SLAM3 [115], VINS-Mono [116], and OKVIS [117]. More recently, the community has also introduced dense optical flow [61] and direct, photometric fusion techniques [118]. The above-mentioned frameworks all employ world-centric coordinates, and as such the stability of the estimation is highly dependent on reliable local map tracking. Furthermore, the use of normal cameras and their tendency of producing blurry images naturally puts limitations to the tolerable motion dynamics.

The ego-velocity can be obtained as an implicitly estimated sub-state estimated in position-based visual-inertial fusion, by conducting temporal differentiation of positions, or by direct estimation from sensor measurements. Most velocity estimation algorithms rely on the relationship between pixel velocities (optical flow) and metric velocities [119–123]. PX4FLOW [121] is a popular optical flow sensor that consists of a camera, a gyroscope and an ultrasonic range sensor. The velocity estimation of PX4FLOW requires stable depth readings and highly depends on the planarity of the observed scene. Song et. al. [119] compute the 2D velocity of a downward-facing camera from optical flow measurements using a known scene depth assumption. Weiss et. al. [122] propose an inertial-optical flow framework for metric 3D speed estimation of a self-calibrating camera-IMU setup. The camera is regarded as a speed sensor, and the algorithm makes use of the continuous 8-point algorithm [124] for scale-invariant velocity samples. The combined use of optical flow and IMU measurements achieves complete dynamic vehicle state estimation [123]. More recently, Deng et. al. [125] propose a multicopter metric velocity estimation algorithm which also combines a low-cost IMU and a monocular camera. Outliers in the point

correspondences are removed by the Mean Shift algorithm, and the internal estimator is given by a Linear Kalman Filter. Moreover, Gao et al. [126] rely on optical flow extracted from a forward-looking stereo camera to estimate the translational velocity of as MAV.

Different from the aforementioned velocity estimation approaches, the method in Chapter 6 estimates speed by fusing an event camera and an IMU, thus achieving better performance in challenging dynamic scenarios.

1.3.5 *Line-based Visual SLAM*

Besides an abundance of works on sparse point-based structure-from-motion, visual odometry and SLAM, the community has invested significant efforts into the development of higher-level feature-based implementations. Most commonly, frameworks include lines and planes to represent larger segments of the environment. They are registered against straight line measurements or uniformly colored image segments such as super-pixels, and generally increase the algorithm's robustness and accuracy in otherwise feature-deprived scenarios. Weng et al. [127] propose a closed-form solution for pose estimation with line correspondences. As demonstrated, at least three views are needed to do projective reconstruction from line correspondences. Hartley further discusses the trifocal tensor, an algebraic object that helps to link the motion in three views to point or line-feature observations [128]. All details about trifocal tensor geometry with lines can be found in the book by Hartley and Zisserman [67]. Line detection and description can be done using the state-of-the-art LSD line detector [129]. Typically, high-level features are utilized in conjunction with points. PL-SLAM [130] is proposed to handle low-texture scenes by merging line features into ORB-SLAM. Tightly-coupled monocular visual–inertial odometry with points and lines is further proposed by He et al. [131], who minimize both IMU integration errors and reprojection errors over points and lines.

Event cameras generate events for changing brightness levels at every pixel. As such, events are mostly generated by moving high-gradient regions in the image, i.e. they are mostly sensitive to moving appearance edges. Given the abundance of straight lines in man-made environments and the compactness of line representations, this work utilize them in our event-inertial fusion framework in Chapter 6 to model event-generating segments of the environment in 3D.

1.4 CONTRIBUTIONS

This thesis mainly has the following contributions:

- For the generalized camera model, this work propose a new global optimization method for estimating the Generalized Essential Matrix (GEM). Unlike existing algorithms, the new algorithm does not rely on local optimization or relinearization techniques, enhancing its applicability and robustness. It also yields a provably global optimal solution within polynomial time. Furthermore, building on the generalized camera model, this work tackle the challenging problem of solving for unknown scale. To advance the solution of this problem, this work introduce the first closed-form solution for the general case, utilizing either 26 point correspondences (PC) or 9 affine correspondences (AC). In cases where the direction of gravity is known, this work also propose a solver that requires only 2 ACs.

- In various geometric vision problems, including solving for the generalized camera model, solving systems of polynomial equations is required. This work propose the first approach to enhance the stability of polynomial solving using deep learning. Specifically, in addressing common problem categories in geometric vision, altering the order of variables effectively changes the arrangement of columns in the initial coefficient matrix. This discovery indicates that the same elimination template can be applied in different ways, each potentially yielding different levels of solution accuracy. Secondly, this work demonstrated that the original set of coefficients contains enough information to train a classifier capable of online selection of an efficient solver with minimal computational cost.

- However, in environments of fast motion and insufficient lighting, traditional cameras often fail to operate effectively. To address pose estimation challenges under such conditions, the thesis consider developing algorithms based on event cameras. Unlike traditional cameras, event cameras generate data only when there is a change in pixel-level brightness, reducing data volume and energy consumption while avoiding motion blur and exposure issues. The thesis demonstrated the reliability of event-based pure visual odometry on planar ground vehicles by applying the nonholonomic motion model constraints of the Ackermann steering platform. This work extended the single-feature n-linear to quasi-time-continuous event trajectories

and realized a polynomial form through variable-order Taylor expansion. Robust averaging of multiple event trajectories was achieved through histogram voting. Furthermore, the thesis proposed a novel tight visual-inertial coupling scheme, which achieves tight coupling directly at the first-order kinematics level and establishes a direct relationship dependent on event and camera velocity using trifocal tensor geometry, effectively obtaining velocity estimates in high-dynamic situations. Noise and outliers are handled through a nested dual-layer RANSAC scheme and tightly coupled with pre-integrated inertial signals, using a sliding window optimizer to obtain smooth velocity signals. These contributions showcase the superiority of event-based pure visual odometry and tight event-inertial coupling methods under challenging conditions.

1.5 STRUCTURE

This thesis is divided into the following chapters:

- **Chapter 1** introduces the background of the research, elucidating its necessity and significance. Initially, it discusses the state of visual SLAM technology and the existing issues, then delves into specific analyses from several key research directions, including generalized cameras, solving systems of polynomial equations, and event camera velocity estimation, providing a detailed overview of the current research status in these areas. On this basis, the contributions of this research to these key fields are articulated. Finally, the structure of the entire thesis is outlined.

- **Chapter 2** presents the latest advancements in solving for the generalized relative pose. It specifically utilizes convex optimization techniques for estimating the Generalized Essential Matrix (GEM), minimizing the sum of squared residuals through a Quadratically Constrained Quadratic Program (QCQP), overcoming the limitations of traditional methods in numerical stability and achieving high precision. This method is capable of obtaining a provably global optimal solution within polynomial time, and its superior performance has been validated through experiments in both synthetic and real-world scenarios.

- **Chapter 3** presents the latest advancements in solving for the generalized relative pose and scale problem. It introduces the first closed-form

solution for aligning externally calibrated view sets. By utilizing point correspondences or affine correspondences, combined with information on the direction of gravity, this method is capable of achieving fast and accurate estimates. Its testing on synthetic data and real-world datasets has demonstrated excellent performance, proving its suitability for real-time applications.

- **Chapter 4** discusses advancements in using deep learning to improve the solving of systems of polynomial equations. It specifically explores the impact of variable reordering in solving geometric vision problems and proposes a method based on training classifiers with the original set of coefficients to online select the optimal solver, aiming to enhance solving accuracy and efficiency.

- **Chapter 5** showcases the application of event-based visual odometry on planar ground vehicles, by transforming event streams into quasi-time-continuous trajectories and applying Ackermann steering constraints, achieving accurate and robust estimation of the vehicle's instantaneous rotational speed. Under challenging lighting conditions, this method outperforms traditional techniques.

- **Chapter 6** introduces a novel approach based on tight event-inertial coupling, utilizing dynamic visual sensors and trifocal tensor geometry to estimate velocity directly at the first-order kinematics level, achieving reliable velocity estimation independent of absolute coordinates in high-dynamic scenes. This method demonstrated performance superior to traditional approaches on both simulated and real data.

- **Chapter 7** concludes with a summary of the research findings and an outlook on future work.

2

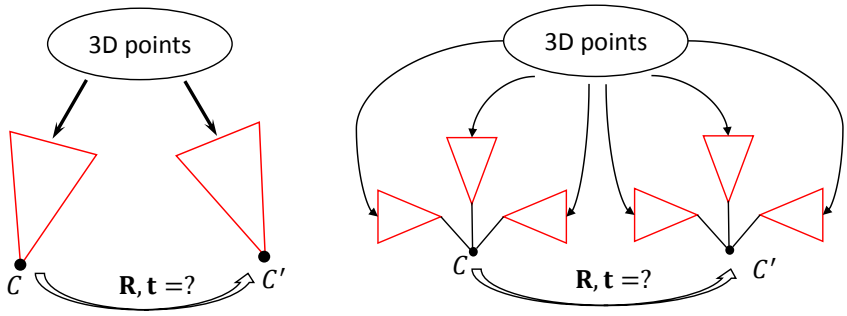
PROGRESS ON GENERALIZED RELATIVE POSE

Relative pose estimation from images plays an important role in many geometric vision tasks, such as structure-from-motion (SfM) and simultaneous localization and mapping (SLAM). While *central cameras* can be modeled by the pin-hole or perspective camera model [67], more general *non-central cameras* such as multi-camera arrays are modelled by the generalized camera model [132]. This chapter presents a new method to estimate the generalized essential matrix (GEM) or relative pose for non-central cameras.

The essential matrix encodes the relative pose for pin-hole cameras and is well understood [7, 67, 69]. GEM estimation is more involved. A generalized camera is formed by abstracting landmark observations into spatial rays that are no longer required to originate from a common point (i.e. the focal point). Figure 2.1 demonstrates the difference between central and non-central cameras. As illustrated, the generalized camera model allows us to describe the measurements of a number of interesting camera systems, such as a multi-camera rig of rigidly attached cameras.

From a more abstract and geometric point of view, a generalized camera consists of a Euclidean reference frame in which measurements are represented by rays in space, described by a suitable parameterization such as Plücker line vectors. In contrast to the standard essential matrix for which there exists an unobservability in the norm of the translation, the translation extracted from a GEM is generally unique. The down-side of this scale observability is that the minimal solution of the GEM requires at least 6 instead of only 5 correspondences across the two views (i.e. one correspondence per degree of freedom in the problem).

There are both linear [28] and non-linear solutions [19, 25, 26] to GEM estimation. The linear solver—also known as the 17-point algorithm—takes 17 correspondences to derive the relative pose of the generalized camera. This method can be easily applied to an arbitrarily large number of points. However, its solution is not globally optimal, as the linearization ignores side-constraints on the GEM and the contained essential and rotation matrices. The most closely related works to ours are the non-minimal solvers by Kneip and Li [25] and Campos *et al.* [26], which use many correspondences to calculate a potentially accurate relative pose, but rely



(a) central camera (b) non-central camera

FIGURE 2.1: Relative pose estimation. Red triangles represent perspective cameras. Solid arrows pointing from 3D points to cameras depict the imaging process. In the non-central camera scenario, three rigidly attached central cameras constitute a non-central camera.

on local optimization methods and therefore may depend on a sufficiently accurate initial guess. They do not guarantee global optimality.

By contrast, the present chapter leverages convex optimization to—for the first time—come up with a fast and certifiably globally optimal solution to the non-minimal generalized relative pose problem, whose optimality may be certified *a-posteriori*. In summary, the contribution of this chapter is two-fold:

- **Formulation**. This work propose a novel formulation for GEM estimation of generalized cameras, and discuss its relation to the previously proposed eigenvalue-based formulation in [25].
- **Optimization**. This work provide a certifiably globally optimal solution by semidefinite relaxation (SDR) of the original formulation. This work also provide a sufficient and necessary condition to recover the optimal GEM from the relaxed problem.

As demonstrated in Section 2.3, our method sets a new state-of-the-art in terms of both accuracy and robustness while at the same time remaining computationally efficient.

2.1 NON-MINIMAL SOLVER FOR GEM ESTIMATION

Relative pose consists of a translation $\mathbf{t}$ — expressed in the first frame and denoting the position of the second frame w.r.t. the first one — and a rotation $\mathbf{R}$ — transforming vectors from the second into the first frame[1]. The translation $\mathbf{t} = [t_1, t_2, t_3]^\top$ is thus identical with a point in $\mathbb{R}^3$. The 3D rotation $\mathbf{R}$ is a 3×3 orthogonal matrix with determinant 1 and belonging to the Special Orthogonal group $\mathrm{SO}(3)$, i.e.,

$$\mathrm{SO}(3) \triangleq \{\mathbf{R} \in \mathbb{R}^{3\times3} | \mathbf{R}^\top\mathbf{R} = \mathbf{I}_3, \det(\mathbf{R}) = 1\}, \tag{2.1}$$

where $\mathbf{I}_3$ is the 3×3 identity matrix.

The essential matrix $\mathbf{E}$ is defined as

$$\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}, \tag{2.2}$$

where $[\cdot]_\times$ constructs the corresponding skew-symmetric matrix of a 3-dimensional vector [67]. The elements of the essential matrix $\mathbf{E}$ and the rotation matrix $\mathbf{R}$ are denoted by e_{ij} and r_{ij}, respectively, where i represents the row index and j the column index. This work furthermore define the vectors

$$\mathbf{e} \triangleq \mathrm{vec}(\mathbf{E}) = [e_{11}\ e_{21}\ \ldots\ e_{33}]^T, \text{ and} \tag{2.3}$$

$$\mathbf{r} \triangleq \mathrm{vec}(\mathbf{R}) = [r_{11}\ r_{21}\ \ldots\ r_{33}]^T, \tag{2.4}$$

where $\mathrm{vec}(\cdot)$ stacks matrix entries by column-first order.

This work define the essential matrix set as

$$\mathcal{M}_\mathbf{E} \triangleq \{\mathbf{E} \mid \mathbf{E} = [\mathbf{t}]_\times \mathbf{R}, \exists\ \mathbf{R} \in \mathrm{SO}(3)\}. \tag{2.5}$$

This essential matrix set is called the *essential matrix manifold* [133]. It is worth mentioning that scale-ambiguity does not exist in GEM estimation, which is why $\mathcal{M}_\mathbf{E}$ does not contain any constraints on $\mathbf{t}$. By contrast, there is a scale-ambiguity for standard relative pose estimation, and the translation $\mathbf{t}$ is typically restricted to length 1.

2.1.1 *Generalized Essential Matrix*

Now review the GEM describing the relative pose geometry for generalized cameras [25, 28, 132]. As outlined in [132], the transformation rule and the

1 Bold capital letters denote matrices (e.g., $\mathbf{E}$ and $\mathbf{R}$); bold lower-case letters denote column vectors (e.g., $\mathbf{e}$, $\mathbf{r}$, and $\mathbf{t}$); non-bold lower-case letters represent scalars (e.g., e and r). $\mathbf{X}_{[a:b,c:d]}$ stands for the submatrix of $\mathbf{X}$ constructed by rows $a \sim b$ and columns $c \sim d$; $\mathbf{x}_{[a:b]}$ stands for the entries of vector $\mathbf{x}$ indexed from a to b.

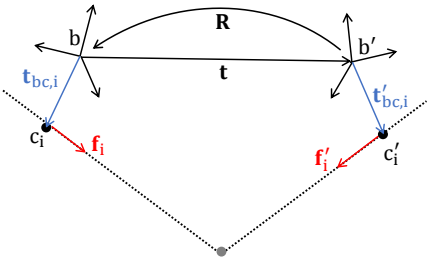


FIGURE 2.2: Geometry of the generalized relative pose problem for multi-camera systems.

intersection-constraint of Plücker line-vectors easily leads to the epipolar constraint

$$\mathbf{l}_i^\top \begin{bmatrix} \mathbf{E} & \mathbf{R} \\ \mathbf{R} & \mathbf{0} \end{bmatrix} \mathbf{l}'_i = 0, \tag{2.6}$$

where $(\mathbf{l}_i, \mathbf{l}'_i)$ denotes a pair of corresponding Plücker line-vectors pointing at the i-th 3D point from two different generalized cameras.

Figure 6.4 illustrates a multi-camera system, which is a common special case of a generalized camera. A point on each Plücker-line is easily given by the capturing camera's center c_i, seen from the origin of the multi-camera system b. If denoting this displacement by $\mathbf{t}_{bc,i}$, could obtain

$$\mathbf{l}_i = \begin{bmatrix} \mathbf{f}_i \\ \mathbf{t}_{bc,i} \times \mathbf{f}_i \end{bmatrix}. \tag{2.7}$$

Note that this work assume that—without loss of generality—c and b have identical orientation. The generalized epipolar constraint thus becomes

$$\mathbf{f}_i^\top \mathbf{E} \mathbf{f}'_i + \mathbf{f}_i^\top \mathbf{R} \mathbf{h}'_i + \mathbf{h}_i^\top \mathbf{R} \mathbf{f}'_i = 0, \tag{2.8}$$

where

$$\mathbf{h}_i \triangleq \mathbf{t}_{bc,i} \times \mathbf{f}_i; \quad \mathbf{h}'_i \triangleq \mathbf{t}'_{bc,i} \times \mathbf{f}'_i.$$

2.1.2 *Optimizing the GEM by Minimizing the Algebraic Error*

Due to the existence of measurement noise, the generalized epipolar constraint will not be strictly satisfied. Denoting the residual for i-th correspondence as

$$\varepsilon_i = \mathbf{f}_i^\top \mathbf{E}\mathbf{f}_i' + \mathbf{f}_i^\top \mathbf{R}\mathbf{h}_i' + \mathbf{h}_i^\top \mathbf{R}\mathbf{f}_i', \tag{2.9}$$

the summation of squared residuals for N correspondences $\{(\mathbf{l}_i, \mathbf{l}_i')\}_{i=1}^N$ becomes a quadratic function in $\mathbf{e}$ and $\mathbf{r}$

$$\varepsilon \triangleq \sum_{i=1}^N \varepsilon_i^2 = [\mathbf{e}^\top, \mathbf{r}^\top]\mathbf{C}\begin{bmatrix}\mathbf{e}\\ \mathbf{r}\end{bmatrix}. \tag{2.10}$$

$\mathbf{C}$ can be expressed explicitly by

$$\mathbf{C} = \begin{bmatrix}\mathbf{C}_1 & \mathbf{C}_4 + \mathbf{C}_5\\ \mathbf{C}_4^\top + \mathbf{C}_5^\top & \mathbf{C}_2 + \mathbf{C}_3 + \mathbf{C}_6 + \mathbf{C}_6^\top\end{bmatrix}, \tag{2.11}$$

where

$$\begin{cases}\mathbf{C}_1 = \sum_{i=1}^N \left(\mathbf{f}_i' \otimes \mathbf{f}_i\right)\left(\mathbf{f}_i' \otimes \mathbf{f}_i\right)^\top\\ \mathbf{C}_2 = \sum_{i=1}^N \left(\mathbf{h}_i' \otimes \mathbf{f}_i\right)\left(\mathbf{h}_i' \otimes \mathbf{f}_i\right)^\top\\ \mathbf{C}_3 = \sum_{i=1}^N \left(\mathbf{f}_i' \otimes \mathbf{h}_i\right)\left(\mathbf{f}_i' \otimes \mathbf{h}_i\right)^\top\\ \mathbf{C}_4 = \sum_{i=1}^N \left(\mathbf{f}_i' \otimes \mathbf{f}_i\right)\left(\mathbf{h}_i' \otimes \mathbf{f}_i\right)^\top\\ \mathbf{C}_5 = \sum_{i=1}^N \left(\mathbf{f}_i' \otimes \mathbf{f}_i\right)\left(\mathbf{f}_i' \otimes \mathbf{h}_i\right)^\top\\ \mathbf{C}_6 = \sum_{i=1}^N \left(\mathbf{h}_i' \otimes \mathbf{f}_i\right)\left(\mathbf{f}_i' \otimes \mathbf{h}_i\right)^\top.\end{cases}$$

Note that $\{\mathbf{C}_j\}_{j=1}^6$ are Gram matrices, so they are positive semidefinite (PSD) and symmetric (and so does $\mathbf{C}$). In practice, $\mathbf{C}$ is positive definite for non-minimal GEM estimation scenario.

2.1.3 *A QCQP Formulation*

The problem of minimizing the algebraic error on the manifold $\mathcal{M}_\mathbf{E}$ can be formulated as

$$\begin{aligned}\min_{\mathbf{E},\mathbf{R},\mathbf{t}} \quad & [\mathbf{e}^\top, \mathbf{r}^\top]\mathbf{C}\begin{bmatrix}\mathbf{e}\\ \mathbf{r}\end{bmatrix}\\ \text{s.t.} \quad & \mathbf{E} = [\mathbf{t}]_\times \mathbf{R}, \quad \mathbf{R} \in \mathrm{SO}(3).\end{aligned} \tag{2.12}$$

$$\min_{\mathbf{E},\mathbf{R},\mathbf{t}} \ [\mathbf{e}^\top, \mathbf{r}^\top]\mathbf{C}\begin{bmatrix}\mathbf{e}\\ \mathbf{r}\end{bmatrix} \ \text{s.t.} \ \ \mathbf{E} = [\mathbf{t}]_\times \mathbf{R}, \ \ \mathbf{R} \in \mathrm{SO}(3). \tag{2.13}$$

This problem is a QCQP: The objective is a sum of squares, which are PSD quadratic polynomials; the largest set of independent quadratic constraints to define SO(3) is 20 [134, 135]; and, lastly, the constraint between $\mathbf{E}$, $\mathbf{R}$ and $\mathbf{t}$, meaning $\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}$, is also quadratic. The problem has 21 variables and 29 constraints.

There are some interesting examples in the literature on how the introduction of linearly independent redundant constraints into a QCQP formulation may significantly improve the tightness of the subsequent semidefinite relaxation [7, 135–137]. For the 20 quadratic constraints considered for SO(3), more than half of them are also redundant and added only for the sake of better tightness [135]. Inspired by this idea, this work introduce redundant constraints for problem (2.13). The below equalities are easily verified:

$$\begin{cases} \mathbf{t}^\top \mathbf{E} = \mathbf{t}^\top([\mathbf{t}]_\times \mathbf{R}) = \mathbf{0} \\ \mathbf{E}\mathbf{E}^\top = ([\mathbf{t}]_\times \mathbf{R})([\mathbf{t}]_\times \mathbf{R})^\top = [\mathbf{t}]_\times [\mathbf{t}]_\times^\top \\ \mathbf{E}\mathbf{R}^\top = ([\mathbf{t}]_\times \mathbf{R})\mathbf{R}^\top = [\mathbf{t}]_\times \end{cases} \tag{2.14}$$

These 3 equalities introduce 3, 6 and 9 additional constraints, respectively.

2.1.4 *Relations between Algebraic Error Based and Eigenvalue Based Formulations*

In [25], an eigenvalue-based formulation was proposed. Here demonstrate the close relation between the algebraic-error-based and the eigenvalue-based formulation. By substituting (3.9) into (2.8) and applying the permutation rule for triple scalar products, could obtain

$$-(\mathbf{f}_i \times \mathbf{R}\mathbf{f}_i')^\top \mathbf{t} + \left(\mathbf{f}_i^\top \mathbf{R}\mathbf{h}_i' + \mathbf{h}_i^\top \mathbf{R}\mathbf{f}_i'\right) = 0, \tag{2.15}$$

which can obviously be rewritten as

$$\begin{aligned} &\mathbf{g}_i^\top \tilde{\mathbf{t}} = 0, \ \ \text{with} \\ &\mathbf{g}_i = \begin{bmatrix} \mathbf{f}_i \times \mathbf{R}\mathbf{f}_i' \\ \mathbf{f}_i^\top \mathbf{R}\mathbf{h}_i' + \mathbf{h}_i^\top \mathbf{R}\mathbf{f}_i' \end{bmatrix} \ \text{and} \ \tilde{\mathbf{t}} = \begin{bmatrix} -w\mathbf{t} \\ w \end{bmatrix}. \end{aligned} \tag{2.16}$$

$\mathbf{g}_i$ here is called a *generalized epipolar plane normal vector*, and $\tilde{\mathbf{t}}$ the *homogeneous translation vector*, which has arbitrary scale [25]. This work set w as 1 without loss of generality. Denote

$$\mathbf{G} = [\mathbf{g}_1, \cdots, \mathbf{g}_k], \tag{2.17}$$

$$\mathbf{H} = \mathbf{G}\mathbf{G}^\top = \sum_{i=1}^{N} \mathbf{g}_i \mathbf{g}_i^\top. \tag{2.18}$$

Then one can express the summation of residuals by this new parameterization

$$\varepsilon = \sum_{i=1}^{N} \varepsilon_i^2 = \sum_{i=1}^{N} (\mathbf{g}_i^\top \tilde{\mathbf{t}})^2 = \left\| \mathbf{G}^\top \tilde{\mathbf{t}} \right\|_2^2. \tag{2.19}$$

Thus the algebraic-error-based formulation (2.13) is equivalent to the following problem

$$\min_{\mathbf{R},\tilde{\mathbf{t}}} \ \left\| \mathbf{G}^\top \tilde{\mathbf{t}} \right\|_2^2 \quad \text{s.t. } \mathbf{R} \in \mathrm{SO}(3), \ \ \tilde{\mathbf{t}}_{[4]} = 1. \tag{2.20}$$

This problem can be further reformulated as

$$\min_{\mathbf{R}} \ J(\mathbf{R}) \quad \text{s.t. } \mathbf{R} \in \mathrm{SO}(3), \tag{2.21}$$

where

$$J(\mathbf{R}) = \min_{\tilde{\mathbf{t}}} \ \left\| \mathbf{G}^\top \tilde{\mathbf{t}} \right\| \quad \text{s.t. } \tilde{\mathbf{t}}_{[4]} = 1. \tag{2.22}$$

If replace the constraint in problem (2.22) by $\|\tilde{\mathbf{t}}\| = 1$, $J(\mathbf{R})$ can be viewed as finding the optimal $\tilde{\mathbf{t}}$ to minimize $\|\mathbf{G}\tilde{\mathbf{t}}\|$ subject to the condition $\|\tilde{\mathbf{t}}\| = 1$. The solution is the unit eigenvector corresponding to the smallest eigenvalue of the matrix $\mathbf{H} = \mathbf{G}^\top \mathbf{G}$. Let $\sigma_{\mathbf{H},\min}$ denote the smallest eigenvalue of $\mathbf{H}$, thus the optimization problem becomes

$$\min_{\mathbf{R}} \ \sigma_{\mathbf{H},\min} \quad \text{s.t. } \mathbf{R} \in \mathrm{SO}(3). \tag{2.23}$$

which is exactly the eigenvalue-based formulation that was proposed in [25].

From the previous analysis, it can be seen that the algebraic error formulation and the eigenvalue-based formulation differ only by the domain of the translation vector. The algebraic error method implicitly assumes that the optimal translation is never infinite, as otherwise one can not assume that the homogeneous coordinate of $\tilde{\mathbf{t}}$ is 1. Fortunately, infinite translations in relative pose estimation are not a practical concern.

2.2 SEMIDEFINITE RELAXATION AND OPTIMIZATION

This work use semidefinite relaxation (SDR) to solve QCQP problem (2.13). Let us rewrite it in more general form as

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \ & \mathbf{x}^\top \mathbf{C}_0 \mathbf{x} \\ \text{s.t.} \ & \mathbf{x}^\top \mathbf{A}_i \mathbf{x} = 0, \quad i = 1, \cdots, m \\ & \mathbf{x}^\top \mathbf{L} \mathbf{x} = 1, \end{aligned} \tag{2.24}$$

where

$$\mathbf{x} = [\mathrm{vec}(\mathbf{E}); \mathrm{vec}(\mathbf{R}); \mathbf{t}; y], \tag{2.25}$$

is a vector stacking all variables. Note that here add an additional variable y that makes the objective and constraints purely quadratic (i.e., no linear or constant term in the objective and no linear term in the equality constraints). This trick is called *homogenization* [135, 138], and introduces the constraint $\mathbf{x}_{[n]}^2 = 1$. By introducing a matrix $\mathbf{L} = \mathrm{diag}([0, \cdots, 0, 1])$, this constraint can be reformulated as $\mathbf{x}^\top \mathbf{L} \mathbf{x} = 1$. Matrices $\mathbf{C}_0, \mathbf{A}_1, \cdots, \mathbf{A}_m \in \mathbb{S}^n$ are determined by the original problem (2.13), where $\mathbb{S}^n$ denotes the set of all real symmetric $n \times n$ matrices.

In our problem, $n = 22$; $\mathbf{C}_0 = \begin{bmatrix} \mathbf{C} & \mathbf{0}_{18\times 4} \\ \mathbf{0}_{4\times 18} & \mathbf{0}_{4\times 4} \end{bmatrix}$. A crucial first step in deriving an SDR of problem (2.24) is to observe that

$$\mathbf{x}^\top \mathbf{C}_0 \mathbf{x} = \mathrm{trace}(\mathbf{x}^\top \mathbf{C}_0 \mathbf{x}) = \mathrm{trace}(\mathbf{C}_0 \mathbf{x}\mathbf{x}^\top), \tag{2.26}$$

$$\mathbf{x}^\top \mathbf{A}_i \mathbf{x} = \mathrm{trace}(\mathbf{x}^\top \mathbf{A}_i \mathbf{x}) = \mathrm{trace}(\mathbf{A}_i \mathbf{x}\mathbf{x}^\top). \tag{2.27}$$

In particular, both the objective function and constraints in problem (2.24) are linear in the matrix $\mathbf{x}\mathbf{x}^\top$. Thus, by introducing a new variable $\mathbf{X} = \mathbf{x}\mathbf{x}^\top$ and noting that $\mathbf{X} = \mathbf{x}\mathbf{x}^\top$ is equivalent to $\mathbf{X}$ being a rank one symmetric PSD matrix, can obtain the following equivalent form of problem (2.24):

$$\begin{aligned} \min_{\mathbf{X}\in\mathbb{S}^n} \ & \mathrm{trace}(\mathbf{C}_0 \mathbf{X}) \\ \text{s.t.} \ & \mathrm{trace}(\mathbf{A}_i \mathbf{X}) = 0, \quad i = 1, \cdots, m, \\ & \mathrm{trace}(\mathbf{L}\mathbf{X}) = 1, \quad \mathbf{X} \succeq \mathbf{0}, \quad \mathrm{rank}(\mathbf{X}) = 1. \end{aligned} \tag{2.28}$$

Here, $\mathbf{X} \succeq \mathbf{0}$ means that $\mathbf{X}$ is PSD. Solving rank constrained semidefinite programs is NP-hard [139]. SDR drops the rank constraint $\text{rank}(\mathbf{X}) = 1$ to obtain the following relaxed version of problem (2.28)

$$\begin{aligned} \min_{\mathbf{X} \in \mathbb{S}^n} \quad & \text{trace}(\mathbf{C}_0 \mathbf{X}) \\ \text{s.t.} \quad & \text{trace}(\mathbf{A}_i \mathbf{X}) = 0, \quad i = 1, \cdots, m, \\ & \text{trace}(\mathbf{L}\mathbf{X}) = 1, \quad \mathbf{X} \succeq \mathbf{0}. \end{aligned} \tag{2.29}$$

Problem (2.29) turns out to be an instance of a semidefinite program (SDP) [138, 139], which may be solved using convex optimization. Modern solvers for SDP are based on primal-dual interior point methods. Its dual problem is

$$\begin{aligned} \max_{\boldsymbol{\lambda}, \rho} \quad & \rho \\ \text{s.t.} \quad & \mathbf{Q}(\boldsymbol{\lambda}, \rho) = \mathbf{C}_0 - \sum_{i=1}^{m} \lambda_i \mathbf{A}_i - \rho \mathbf{L} \succeq 0, \end{aligned} \tag{2.30}$$

where $\boldsymbol{\lambda} = [\lambda_1, \cdots, \lambda_m]^\top \in \mathbb{R}^m$. Problem (2.30) is called the *Lagrangian dual problem* of problem (2.24), and $\mathbf{Q}(\boldsymbol{\lambda}, \rho)$ is the Hessian of the Lagrangian. In summary, the relations between the main formulations are demonstrated in Fig. 2.3.

Now prove that there is no duality gap between (2.29) and (2.30). Thus the problem can be readily solved using off-the-shelf primal-dual interior point methods [140].

Theorem 2.2.1 *For QCQP problem* (2.13)*, there is no duality gap between the primal SDP problem* (2.29) *and its dual problem* (2.30)*.*

Proof 2.2.1 *Denote the optimal value for problem* (2.29) *and its dual problem* (2.30) *as f_{primal} and f_{dual}. The inequality $f_{primal} \geq f_{dual}$ follows from weak duality. Equality, and the existence of $\mathbf{X}^\star$ and $\boldsymbol{\lambda}^\star$ which attain the optimal values follow if this work can show that the feasible regions of both the primal and dual problems have nonempty interiors [][]Theorem 3.1]vandenberghe1996semidefinite (also known as Slater's constraint qualification [141].)*

For the primal problem (2.29)*, let $\mathbf{E}_0$ be an arbitrary point on the essential matrix manifold $\mathcal{M}_{\mathbf{E}}$: $\mathbf{E}_0 = [\mathbf{t}_0]_\times \mathbf{R}_0$. Denote $\mathbf{x}_0 = [\text{vec}(\mathbf{E}_0); \text{vec}(\mathbf{R}_0); \mathbf{t_0}; 1]$. It can be*

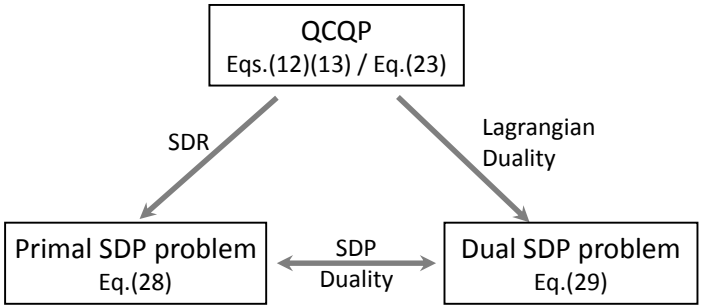


FIGURE 2.3: Relations between the main formulations in this work.

verified that $\mathbf{X}_0 = \mathbf{x}_0\mathbf{x}_0^\top$ *is an interior in the feasible domain of the primal problem. For the dual problem* (2.30)*, this work first list part of the constraints as follows*

$$\begin{cases} h_1 : e_{11}^2 + e_{12}^2 + e_{13}^2 - (t_2^2 + t_3^2) = 0, & \text{(2.31a)} \\ h_2 : e_{21}^2 + e_{22}^2 + e_{23}^2 - (t_1^2 + t_3^2) = 0, & \text{(2.31b)} \\ h_3 : e_{31}^2 + e_{32}^2 + e_{33}^2 - (t_1^2 + t_2^2) = 0, & \text{(2.31c)} \\ h_4 : r_{11}^2 + r_{12}^2 + r_{13}^2 - y^2 = 0, & \text{(2.31d)} \\ h_5 : r_{21}^2 + r_{22}^2 + r_{23}^2 - y^2 = 0, & \text{(2.31e)} \\ h_6 : r_{31}^2 + r_{32}^2 + r_{33}^2 - y^2 = 0, & \text{(2.31f)} \\ h_7 : r_{11}^2 + r_{21}^2 + r_{31}^2 - y^2 = 0, & \text{(2.31g)} \\ h_8 : r_{12}^2 + r_{22}^2 + r_{32}^2 - y^2 = 0, & \text{(2.31h)} \\ h_9 : r_{13}^2 + r_{23}^2 + r_{33}^2 - y^2 = 0, & \text{(2.31i)} \end{cases}$$

where $h_1 \sim h_3$ *follows from the constraint* $\mathbf{E}\mathbf{E}^\top = [\mathbf{t}]_\times [\mathbf{t}]_\times^\top$*, and* $h_4 \sim h_9$ *originates from the constraints* $\mathbf{R}\mathbf{R}^\top = \mathbf{R}^\top\mathbf{R} = \mathbf{I}_3$*. Recall that* $\mathbf{C} \succ 0$*, thus its minimal eigenvector* σ_{min} *is positive. Let* $\lambda_1 \sim \lambda_9$ *correspond to the Lagrangian of* $h_1 \sim h_9$ *respectively. Let the first* 9 *entries in* $\boldsymbol{\lambda}_0$ *satisfy* $\boldsymbol{\lambda}_{0[1:9]} = -\epsilon[1,1,1,1,1,1,1,1,1]^\top$*, and other entries in* $\boldsymbol{\lambda}_0$ *and* ρ_0 *be zero. It can be verified that* $\mathbf{Q}(\boldsymbol{\lambda}_0, \rho_0) = \begin{bmatrix} \mathbf{C} - \epsilon\mathbf{I}_{18} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 2\epsilon\mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 6\epsilon \end{bmatrix} \succ 0$*,* $\forall \epsilon \in (0, \sigma_{min})$*. That means* $\{\boldsymbol{\lambda}_0, \rho_0\}$ *is an interior point in the feasible domain of the dual problem.*

2.2.1 *Further Redundant Constraints*

To improve tightness of the SDR, this work add further redundant constraints on our SDP. The redundant constraint is taken from the SO(3) orbitope.

Definition 2.2.1 (Orbitope [142]) *An orbitope is the convex hull of an orbit of a compact algebraic group that acts linearly on a real vector space. The orbit has the structure of a real algebraic variety, and the orbitope is a convex semi-algebraic set.*

Theorem 2.2.2 (SO(3) Orbitope, Proposition 4.1 in [142]) *The tautological orbitope $conv(SO(3))$ is a spectrahedron whose boundary is a quartic hypersurface. In fact, a 3×3 matrix $\mathbf{R}$ lies in $conv(SO(3))$ if and only if*

$$\mathcal{L}(\mathbf{R}) + \mathbf{I}_4 \succeq 0, \tag{2.32}$$

where $\mathcal{L}(\mathbf{R}) =$

$$\begin{bmatrix} r_{11}+r_{22}+r_{33} & r_{32}-r_{23} & r_{13}-r_{31} & r_{21}-r_{12} \\ r_{32}-r_{23} & r_{11}-r_{22}-r_{33} & r_{21}+r_{12} & r_{13}+r_{31} \\ r_{13}-r_{31} & r_{21}+r_{12} & r_{22}-r_{11}-r_{33} & r_{32}+r_{23} \\ r_{21}-r_{12} & r_{13}+r_{31} & r_{32}+r_{23} & r_{33}-r_{11}-r_{22} \end{bmatrix}.$$

Inequality (2.32) provides an additional linear matrix inequality for our optimization problem. Note that $\{r_{ij}\}_{i,j=1}^{3}$ in $\mathbf{R}$ are also entries in $\mathbf{X}$ since $\mathbf{X} = \mathbf{x}\mathbf{x}^\top$ and $\mathbf{x} = [\mathrm{vec}(\mathbf{E}); \mathrm{vec}(\mathbf{R}); \mathbf{t}; 1]$. Therefore (2.32) can be reformulated in terms of $\mathbf{X}$.

2.2.2 *Recovery of Essential Matrix and Relative Pose*

Once the optimal $\mathbf{X}^\star$ of the SDP primal problem (2.29) has been calculated by an SDP solver, this work are left with the task to recover the optimal essential matrix $\mathbf{E}^\star$. Let us denote $\mathbf{X}_e^\star = \mathbf{X}^\star_{[1:9,1:9]}$, $\mathbf{X}_r^\star = \mathbf{X}^\star_{[10:18,10:18]}$ and $\mathbf{X}_t^\star = \mathbf{X}^\star_{[19:21,19:21]}$. Empirically, this work found that $\mathrm{rank}(\mathbf{X}_e^\star) = 1$. Denoting the eigenvector that corresponds to the nonzero eigenvalue of $\mathbf{X}_e^\star$ as $\mathbf{e}^\star$, the optimal essential matrix is

$$\mathbf{E}^\star = \mathrm{mat}(\mathbf{e}^\star, [3,3]), \tag{2.33}$$

where $\mathrm{mat}(\mathbf{e}, [r,c])$ reshapes the vector $\mathbf{e}$ to an $r \times c$ matrix by column-first order.

Once the essential matrix has been obtained, one can recover rotation $\mathbf{R}^\star$ and translation $\mathbf{t}^\star$ by the standard textbook method [67]. However, $\mathbf{E}^\star$ and its derived translation $\mathbf{t}^\star$ do not have the proper scale. To recover the proper scale, this work denote the unknown scale factor as s and substitute $s\mathbf{E}^\star$ and $\mathbf{R}^\star$ into the generalized epipolar constraint (2.8). This work then calculate the scale s by solving the least squares problem

$$s = -\frac{\sum_{i=1}^{N}(\mathbf{f}_i^\top \mathbf{E}\mathbf{f}_i') \cdot (\mathbf{f}_i^\top \mathbf{R}\mathbf{h}_i' + \mathbf{h}_i^\top \mathbf{R}\mathbf{f}_i')}{\sum_{i=1}^{N}(\mathbf{f}_i^\top \mathbf{E}\mathbf{f}_i')^2}. \tag{2.34}$$

If the denominator in Eq. (2.34) is (near) zero, the problem is in a (near) degenerate configuration in which the scale is (nearly) unobservable. Known degenerate configurations correspond to a generalized camera moving along a straight line or—in some cases—a circular arc. In such a scenario, the real scale can not be recovered, while rotation and translation direction can still be found.

This work empirically verified that $\text{rank}(\mathbf{X}_e^\star)$ and $\text{rank}(\mathbf{X}_t^\star)$ remain 1, while $\text{rank}(\mathbf{X}_r^\star)$ may be 2. Since $\mathbf{X}_r^\star$ does not satisfy the rank-1 constraint, one can no longer recover the rotation from it. Fortunately, this work do not require $\mathbf{X}_r^\star$, and may recover the translation directly from $\mathbf{X}_t^\star$. Similarly, Section 2.2.3 introduces Theorem 2.2.3, a sufficient and necessary condition for global optimality, which again does not depend on $\text{rank}(\mathbf{X}_r^\star)$, which is why global optimality is not influenced by an eventual unobservability of scale. The outline of our method is shown in Algorithm 1.

2.2.3 *A Sufficient and Necessary Condition for Global Optimality*

Since SDR drops the rank-1 constraint, a sufficient condition for global optimality is that the optimal $\mathbf{X}^\star$ satisfies the rank-1 constraint. However, the rank-1 constraint of $\mathbf{X}^\star$ may not be necessary to guarantee global optimality. The following theorem provides a sufficient and necessary condition, which provides a theoretical foundation for the practical pose recovery method described in Section 2.2.2.

Theorem 2.2.3 *For QCQP problem* (2.13) *with constraints* (2.14)*, its SDR problem is tight if and only if: the optimal solution* $\mathbf{X}^\star$ *to its primal SDP problem* (2.29) *satisfies* $\text{rank}(\mathbf{X}_e^\star) = \text{rank}(\mathbf{X}_t^\star) = 1$.

Proof 2.2.2 *First, this work prove the* if *part. Note that* $\mathbf{X}_e^\star$ *and* $\mathbf{X}_t^\star$ *are real symmetric matrices because they are in the feasible region of the primal SDP.*

Algorithm 1: Generalized Essential Matrix Estimation by SDP Optimization

Input: observations $\{(\mathbf{l}_i, \mathbf{l}'_i)\}_{i=1}^{N}$

Output: Essential matrix $\mathbf{E}^\star$, rotation $\mathbf{R}^\star$, and translation $\mathbf{t}^\star$

Construct $\mathbf{C}$ by Eq. (2.11); $\mathbf{C}_0 = \begin{bmatrix} \mathbf{C} & \mathbf{0}_{18\times 4} \\ \mathbf{0}_{4\times 18} & \mathbf{0}_{4\times 4} \end{bmatrix}$;

Construct $\{\mathbf{A}_i\}_{i=1}^{m}$ and $\mathbf{L}$ in problem (2.24) which are independent of input;

Obtain $\mathbf{X}^\star$ by solving SDP problem (2.29) or its dual (2.30) with redundant constraints;

Assert that $\operatorname{rank}(\mathbf{X}_e^\star) = \operatorname{rank}(\mathbf{X}_t^\star) = 1$;

$\mathbf{E}^\star = \operatorname{mat}(\mathbf{e}^\star, [3,3])$, where $\mathbf{e}^\star$ is the eigenvector corresponding to the largest eigenvalue of $\mathbf{X}_e^\star$;

Decompose $\mathbf{E}^\star$ to obtain rotation $\mathbf{R}^\star$ and normalized translation $\mathbf{t}^\star$;

if $\sum_{i=1}^{N} (\mathbf{f}_i^\top \mathbf{E} \mathbf{f}'_i)^2$ *is larger than a threshold* **then**

  Calculate scale s by Eq. (2.34);

  $\mathbf{t}^\star \leftarrow s\mathbf{t}^\star$

else

  $\mathbf{t}^\star$ can only be determined up to scale.

end

Besides it is given that $\text{rank}(\mathbf{X}_e^\star) = \text{rank}(\mathbf{X}_t^\star) = 1$*, thus there exist two vectors* $\mathbf{e}^\star$ *and* $\mathbf{t}^\star$ *satisfying* $\mathbf{e}^\star(\mathbf{e}^\star)^\top = \mathbf{X}_e^\star$ *and* $\mathbf{t}^\star(\mathbf{t}^\star)^\top = \mathbf{X}_t^\star$*.*

According to Theorem 1 in [10], a real 3×3 *matrix* $\mathbf{E}$ *is an essential matrix if and only if there exists a vector* $\mathbf{t}$ *satisfying* $\mathbf{E}\mathbf{E}^\top = [\mathbf{t}]_\times[\mathbf{t}]_\times$*. Note that* $\mathbf{X}_e^\star = \mathbf{e}^\star(\mathbf{e}^\star)^\top$ *and* $\mathbf{X}_t^\star = \mathbf{t}^\star(\mathbf{t}^\star)^\top$ *satisfy the constraints in problem* (2.29) *since they are submatrices of a valid solution* $\mathbf{X}^\star$*. By algebraic derivation based on these constraints, it can be proven that* $\mathbf{E}^\star = mat(\mathbf{e}^\star, [3,3])$ *and* $\mathbf{t}^\star$ *satisfy* $\mathbf{E}^\star\mathbf{E}^{\star\top} = [\mathbf{t}^\star]_\times[\mathbf{t}^\star]_\times$*. Thus* $\mathbf{E}^\star$ *is a valid essential matrix.*

Next this work prove the only if *part. Since SDR is tight, it means this work can uniquely recover a valid relative pose from matrix* $\mathbf{X}^\star$*. According to Theorem 1 in [10], the minimal requirement to define a valid relative pose is constraints about* $\mathbf{E}$ *and* $\mathbf{t}$*. To ensure valid* $\mathbf{E}$ *and* $\mathbf{t}$ *can be recovered from* $\mathbf{X}^\star$*, it should satisfy that* $\text{rank}(\mathbf{X}_e^\star) \leq 1$ *and* $\text{rank}(\mathbf{X}_t^\star) \leq 1$*. Since* $\mathbf{X}_e^\star$ *and* $\mathbf{X}_t^\star$ *cannot be zero matrices (otherwise* $\mathbf{X}^\star$ *is not in the feasible region), the equalities should hold.*

Theorem 2.2.3 provides a sufficient and necessary global optimality condition to recover the optimal solution for the original QCQP. It also provides a method to verify global optimality. Empirically, the optimal $\mathbf{X}^\star$ obtained by the SDP problem always satisfies this condition. Finding the essential conditions to guarantee tightness however remains an open problem [143].

2.3 EXPERIMENTAL RESULTS

this work choose `SDPA` [144] as the interior point method (IPM) solver, and use the default parameters in all experiments. Our method is implemented in `MATLAB`, and all experiments are performed on an Intel Core i7 CPU with 1.7 Hz. To improve efficiency, this work use the results of the 17-point solver [28] for initialization when more than 17 inliers are available. In this chapter, only experiments for synthetic data use this initialization. To improve accuracy, this work follow the *suggest-and-improve* framework for general QCQPs [145]. This work furthermore use a local optimization method [26] to refine the results provided by `SDPA`. The complete method takes an average of only 15 ms.

This work compared our method against several state-of-the-art methods on both synthetic and real data. Specifically, this work compare our method against: (1) the minimal solver `6pt` [19]; (2) the linear solver `17pt` [28]; (3) the generalized eigenvalue solver `ge` [25]; and (4) an alternating minimization method (AMM), denoted `17pt-amm` [26]. Methods `ge` and `17pt-amm` are both initialized by `17pt`. Our own methods are referred to as `sdp` (without any refinement) and `sdp-amm` (with AMM refinement).

Among these methods, the implementation of `17pt-amm` was provided by the authors, and other comparison methods were taken from `OpenGV` [146]. Note furthermore that this work always ensure a balanced number of samples in each camera, independently of the experiment and number of cameras.

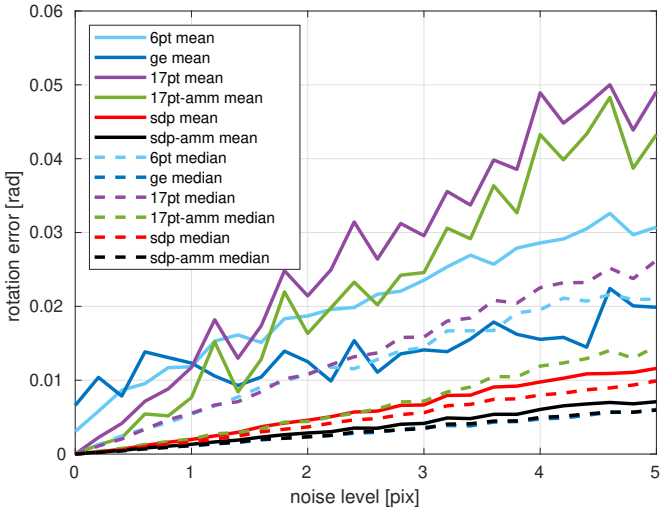
2.3.1 *Results on Synthetic Data*

Noise Resilience: The setup of our experiments is similar to the one proposed in [146]. This work first test image noise resilience. Each method is evaluated for various noise levels reaching from 0 to 5 pixels and over 1000 random experiments per noise level. The rotation errors of all method is shown in Fig. 2.4a. Translation errors follow a similar trend, but are omitted here for the sake of space limitations.

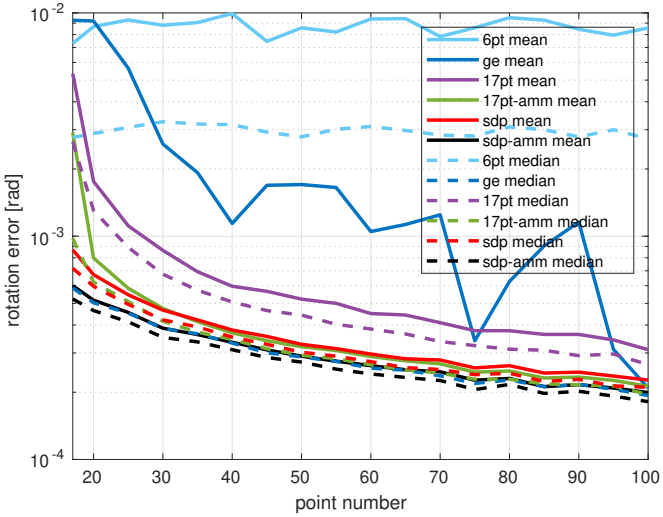
Looking at Fig. 2.4a, this work make the following observations: (1) `sdp-amm` degrades the least, and has a relatively obvious advantage over other methods in terms of both accuracy and robustness. This is partially due to the fact that our method does not depend on any initialization and can always find the global optimum. By contrast, `ge` strongly depends on a good initial value. (2) `sdp-amm` consistently performs better than `sdp`, which underlines the effectiveness of the *suggest-and-improve* framework for general QCQPs [145]. (3) `sdp` still has smaller error than previous state-of-the-art methods.

Number of correspondences: In our next experiment, this work fix the image noise level to 0.5 pixel in standard deviation and vary the number N of point correspondences. `6pt` can only take a subset of the point correspondences, while other methods utilize all point correspondences. To make the comparison more fair, this work randomly sample 20 minimal sets of point correspondences for `6pt`, and take the best result in each experiment. The best here is defined as the result that leads to the smallest algebraic error over all simulated correspondences. This work again show the rotation error in Fig. 2.4b, and make the following observations: (1) As expected, the errors of `17pt`, `17pt-amm`, `sdp`, and `sdp-amm` all decrease as the number of point correspondences is increased. (2) `ge` still depends on a good initialisation. (3) `sdp-amm` still leads to the smallest error among all methods.

Optimality Gap: This work compare the residuals of our method against those of `17pt` and `17pt-amm`, respectively. The corresponding scatter plots are indicated in Fig. 2.5. As can be observed `sdp-amm` has smaller residuals

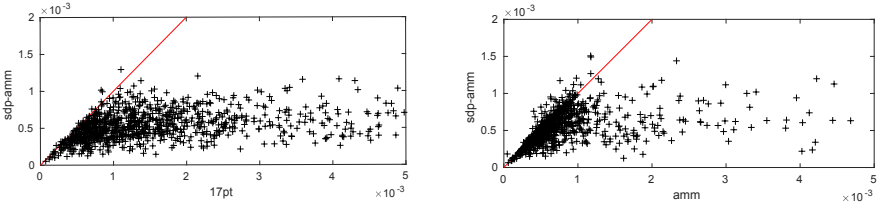


(a) Noise level variations



(b) Number of correspondences

FIGURE 2.4: Mean and median of rotation errors with respect to (a) noise level variations and (b) the number of point correspondences.



(a) ours vs. `17pt` [28] (b) ours vs. `17pt-amm` [26]

FIGURE 2.5: Scatter plot comparing the residuals between two methods. A point lying below the red line indicates that our method outperforms in terms of a smaller residual.

for most of the experiments. The residual of this work's method typically remains below 1.5×10^{-3}. By contrast, `17pt` and `17pt-amm` may have residuals as large as 5×10^{-3}.

Performance within RANSAC: The most relevant performance measure consists of testing all algorithms as part of a hypothesize-and-test framework. This work use the classical RANSAC framework [147], and the same model verification for all methods. For `6pt`, this work use an additional 3 points per hypothesis to disambiguate the solution multiplicity. This has no effect on the cost of the disambiguation, and is safer than disambiguation with only one point, especially regarding the high number of solutions and the cost of hypothesis generation. For `17pt`, `ge`, and our methods, this work sample 17, 12, and 12 points in each iteration, respectively.

The noise is kept at 0.5 pixel. The total number of point correspondences is fixed to 100, and this work vary the outlier fraction. For each outlier fraction generate 2000 synthetic scenes and report the mean and median number of identified inliers. Figure 2.6 reports the number of inliers found by the different methods when integrated into RANSAC. As can be observed, the median of the methods is nearly ideal for all methods except `17pt`. However, `sdp-amm` obtains the largest mean number of identified inliers. In fact, `sdp-amm` is the only method that consistently finds all inliers in each experiment.

2.3.2 *Results on Real Data*

To conclude the evaluation, this work perform experiments on real data and demonstrate that the advantage of the proposed method applies here as well. This work evaluate two datasets. The first one is captured by a custom-made,

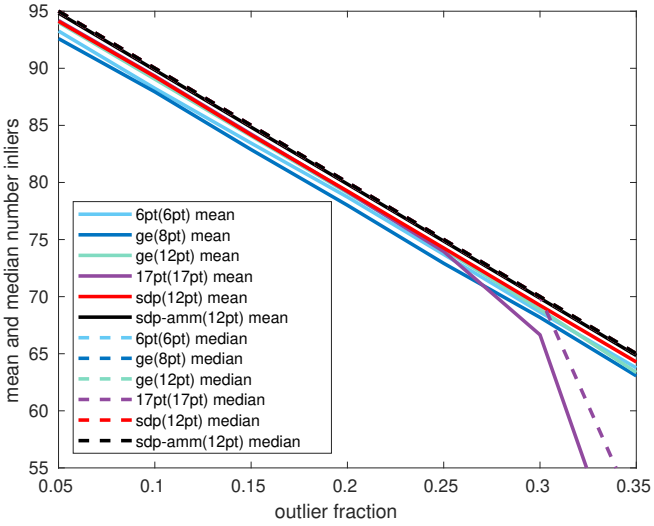
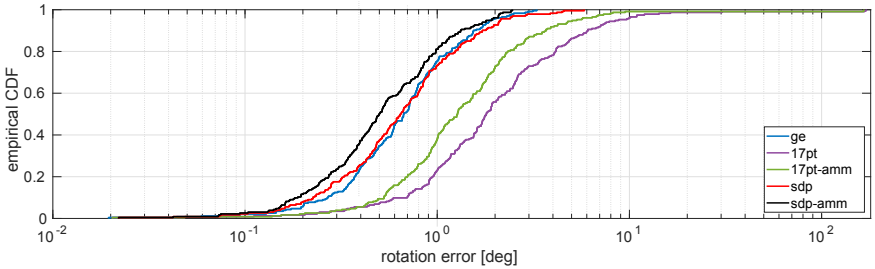
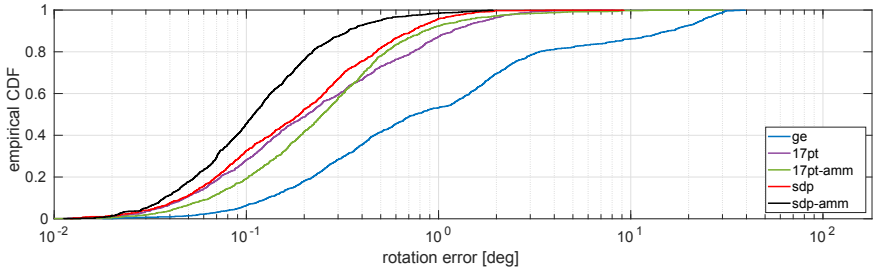


FIGURE 2.6: Mean and median number of identified inliers over outlier fraction.

synchronized 4-camera system mounted on a small-scale automated guided vehicle (AGV), and ground truth is provided by an external motion tracking system. The cameras have a 1216×1936 resolution, are equipped with 48° field-of-view lenses, and are pointing forward, left, right, and backward. The second dataset is taken from the KITTI [148] benchmark, which only has a forward facing stereo camera. This work ignore the overlap in their fields of view, and treat it as a general multi-camera array. Ground truth is provided by a Velodyne LiDAR and a differential GPS. Figures 2.7a and 2.7b show the cumulative distribution functions (CDFs) of respective rotation errors, demonstrating how `sdp-amm` remains the most accurate method. The difference to alternative methods is particularly important on the KITTI sequence. In this sequence, the bearings of the landmark measurements do not have an omni-directional distribution, which is known to be a challenging case for relative pose estimation with generalized cameras.



(a) Results on omni-directional 4-cam dataset



(b) Results on forward-facing 2-cam dataset

FIGURE 2.7: Empirical cumulative errors distributions for (a) a 4-camera dataset with roughly omni-directional measurement distributions (b) a 2-camera dataset with forward facing cameras.

2.4 CONCLUSIONS

This work introduced the first certifiably globally optimal solution to the non-minimal generalized relative pose estimation problem. Extensive experiments on both synthetic and real data demonstrate clearly improved accuracy and robustness over the previous state-of-the-art, including the ability to handle the difficult scenario of a limited combined field of view of all cameras. Furthermore, by including the essential matrix in our parameterization, the dimensionality of our formulation turns out to be even smaller than the one of a previous SDR based method for central cameras. Even without further polishing of our implementation, this technique already enables real-time processing.

3

PROGRESS ON GENERALIZED RELATIVE POSE AND SCALE

Geometric registration methods are the back-bone of many structure-from-motion (SfM) [149] and Visual Simultaneous Localization And Mapping (SLAM) [150] frameworks. At a high-level, they can be divided into two categories. On one hand, we have absolute problems in which we seek the absolute pose of imagery based on correspondences between 2D image points and 3D world points [151]. On the other hand, there are relative problems in which we seek the relative transformation between images based on pair-wise 2D-2D correspondences [69]. The present chapter focuses on the latter case, which is of crucial importance whenever no information about the 3D structure of the observed scene is available. Such scenarios are typical during the initial stages of structure-from-motion (SfM) [149] and in SLAM [150], as well as in purely frame-based SfM or SLAM frameworks where 3D structure data is not available.

Among relative pose estimation problems, the most basic one asks for the identification of the relative pose between two *central views* [67]. Any SfM or visual SLAM algorithm will at least require a solution to this problem before initial 3D points can be triangulated. A generalization of the central problem then consists of estimating the relative pose between two extrinsically calibrated multi-camera rigs (*i.e. generalized cameras*) [3, 19, 23, 25, 28, 132, 152]. In this setup, the measurements taken by individual cameras in one rig pose may be described by the generalized camera model. The latter approach represents 2D image points as spatial rays, originating from the center of each camera. They are mapped and aligned within a shared reference frame encompassing the entire camera rig. This effectively allows the rays to have different origins and lets the model express the measurements taken by a non-central or multi-perspective camera system. The generalized camera model is useful in many practical applications, such as vehicle-mounted surround-view multi-camera displacement calculation [76, 153, 154]. Note that – in contrast to the central case in which we can not observe scale – the generalized camera model exploits exact knowledge about the distances between the camera centers (i. e., the absolute extrinsic calibration parameters of the multi-camera system) to find a full Euclidean transformation between the two systems.

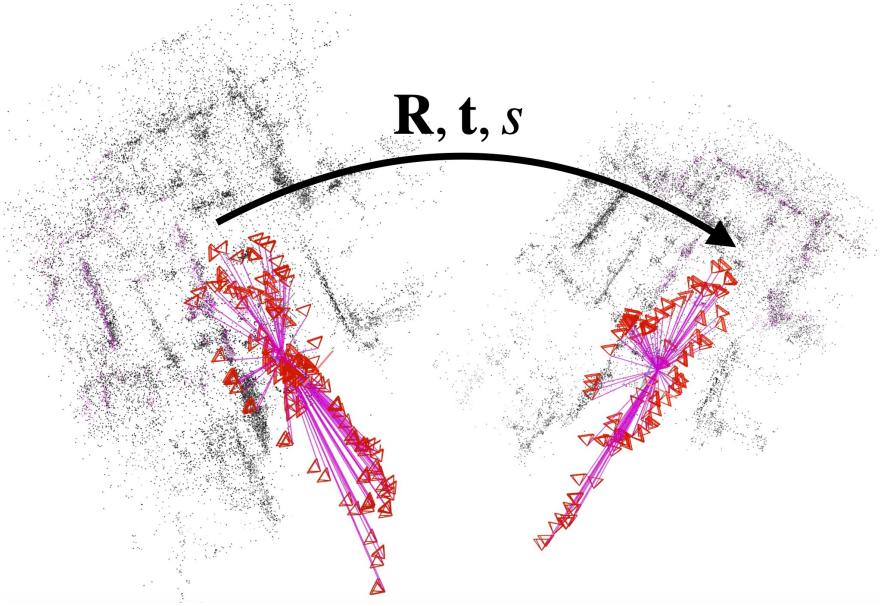


FIGURE 3.1: Typical application for the proposed method. Two view-graphs are reconstructed from images captured in the same environment, and the aim is to merge them. If no information about 3D points is available, it requires a solution to the generalized relative pose and scale problem. Owing to scale ambiguity, each graph appears in a different scale, and the algorithm needs to solve for the relative rotation **R**, translation **t**, and scale factor s.

This chapter focus on a further extension of the calibrated relative pose problem – the *generalized relative pose and scale* problem [33, 34]. It relaxes the requirement of known extrinsic calibration parameters and extends the generalized relative pose algorithm to scenarios in which the baselines of the generalized cameras are known only up to a global scale factor. As outlined in Fig. 3.1, the algorithm is essential in structure-less registration scenarios where we have two sets of views with intra-view transformations derived by SfM or SLAM. Our target is to register them in the same coordinate system. Addressing the inherent memory and computing-intensive nature of estimating and maintaining 3D point clouds [149, 155–158], this work focus on using 2D image features. Besides being lightweight, leveraging 2D features has shown to be more robust to noise and outliers than using

3D points [33] for this problem. Furthermore, as RANSAC-like robust estimation is sensitive to the sample size, this work aim to propose efficient solvers that find transformations from a small number of correspondences.

This work denote the above problem *view-graph registration*. It occurs in both single and multi-agent, purely frame-based monocular SfM and SLAM scenarios [159]. In single-agent scenarios, the requirement for scale-aware registration of sets of registered frames occurs in loop closure. On the one hand, relative pose estimation in single frames is not enough due to scale unobservability. On the other hand, due to potential scale drift along the loop, the registration of sets of views from 2D-2D correspondences requires the identification of an additional scale factor. In multi-agent scenarios, we may think of several agents sending keyframes and relative displacement information to a central node that constructs a global frame-based map. Again, the scale discrepancy between local estimations and the global map requires scale-aware registration of view sets.

Our problem thus consists of reconciling the relative pose between two generalized cameras and an unknown scale factor, i. e., a 7 degree-of-freedom (DOF) similarity transformation between the view-graph reference frames. The present chapter aims at computationally efficient, accurate, and robust methods to recover relative pose and scale. Our detailed contributions are as follows:

- This work present the first closed-form solution for the generalized relative pose and scale problem using 26 point correspondences. This work ensure robustness with rank deficiency checks and multi-dimensional null-spaces. This approach is particularly effective for the overdetermined case, e. g., in the local optimization of LO-RANSAC [160].
- This work present a new solver that uses only 9 correspondences. This is made possible by employing affine-covariant feature detectors, such as ASIFT [161], MODS [162] and DoG-AffNet-HardNet [163, 164]. The resulting affine correspondences (AC) consist of a point correspondence and a 2×2 linear transformation and yield three constraints on the geometric model estimation [11–13]. The requirement for fewer correspondences is crucial for improving robust estimation frameworks such as RANSAC [165].
- A novel 2 AC-based minimal solver that calculates the relative pose and scale in the case of a given directional correspondence between view graphs. This extra information removes two degrees of freedom and is available *by default* in recent capturing devices, e. g., smartphones. With this directional correspondence, the proposed 2 AC

minimal solver achieves state-of-the-art results when combined with a locally optimized RANSAC [160].

3.1 PRELIMINARIES

An *Affine Correspondence (AC)* is a triplet $(\mathbf{x}_i, \mathbf{x}'_i, \mathbf{A}_i)$, where $\mathbf{x}_i = [u_i \; v_i \; 1]^\top$ and $\mathbf{x}'_i = [u'_i \; v'_i \; 1]^\top$ are a corresponding pair of normalized, homogeneous points in two images and $\mathbf{A}_i$ is a 2×2 local affine transformation defining the image warping between the local neighbourhoods of $\mathbf{x}_i$ and $\mathbf{x}'_i$. Its elements in row-major order are a_1, a_2, a_3, and a_4. This work use the definition provided in [166], which defines $\mathbf{A}_i$ as the first-order Taylor-approximation of a planar 3D $\rightarrow$ 2D projection function (i. e., a homography).

In the generalized case, the AC is measured between two cameras C_i and C'_i with extrinsic parameters $\mathbf{R}_{bi}, \mathbf{t}_{bi}$ and $\mathbf{R}'_{bi}, \mathbf{t}'_{bi}$ in the first and second multi-view body frames, respectively. If generalized solvers are applied to extrinsically calibrated multi-perspective rigs, we often assume inter-camera correspondences only, hence $C_i = C'_i$, $\mathbf{R}_{bi} = \mathbf{R}'_{bi}$, and $\mathbf{t}_{bi} = \mathbf{t}'_{bi}$. When applied to continuous non-central cameras, C_i and C'_i are generally not the same and individual for each correspondence. In our primary target case of view-graph registration, C_i and C'_i are not equal but picked from their individual finite sets of possible transformations (the cardinality of each set being equal to the number of images in the respective view-graph).

The *landmark observation vectors* $(\mathbf{f}_i, \mathbf{f}'_i)$ expressed in the multi-view body frame are given as

$$\mathbf{f}_i = \mathbf{R}_{bi}\mathbf{x}_i, \quad \mathbf{f}'_i = \mathbf{R}'_{bi}\mathbf{x}'_i. \tag{3.1}$$

The corresponding 6-dimensional *Plücker line vectors* are denoted $\mathbf{l}_i = [\mathbf{f}_i^\top, (\mathbf{t}_{bi} \times \mathbf{f}_i)^\top]^\top$, and $\mathbf{l}'_i = [\mathbf{f}_i'^\top, (\mathbf{t}'_{bi} \times \mathbf{f}'_i)^\top]^\top$. The incidence constraint of the *generalized relative pose and scale problem* [33, 34] is finally written as

$$\mathbf{f}_i{}^\top \mathbf{E} \mathbf{f}'_i + (\mathbf{t}_{bi} \times \mathbf{f}_i)^\top \mathbf{R} \mathbf{f}'_i + s\mathbf{f}_i{}^\top \mathbf{R}(\mathbf{t}'_{bi} \times \mathbf{f}'_i) = 0. \tag{3.2}$$

Analogous to the conventional epipolar constraint for a pinhole camera, the above constraint can be written as

$$\mathbf{l}_i{}^\top \begin{bmatrix} \mathbf{E} & s\mathbf{R} \\ \mathbf{R} & \mathbf{0} \end{bmatrix} \mathbf{l}'_i = 0, \tag{3.3}$$

where $\mathbf{R}$ is the relative rotation, $\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}$ is the essential matrix with $\mathbf{t}$ the relative translation, and s is a scalar by which the camera baselines $\mathbf{t}'_{bi}$ in the second multi-view frame need to be multiplied to enable registration.

3.2 LINEAR SOLVER FROM POINT CORRESPONDENCES

The linear solver partitions into two sub-problems. The first one is given by the recovery of the generalized essential matrix, while the second one consists of extracting relative rotation, translation and scale.

3.2.1 *Essential Matrix Estimation*

Similar to [28], this work propose a linear algorithm to solve the generalized relative pose and scale problem. Linear solvers are particularly useful for the overdetermined case (i. e., estimating the model from a larger-than-minimal sample) which are essential for using state-of-the-art RANSAC variants [167, 168] or ACs [15, 169]. Given n point correspondences, based on (3.2), we are given constraint

$$\mathbf{B}_{(n\times 27)} \begin{bmatrix} \mathrm{vec}(\mathbf{E}) \\ \mathrm{vec}(\mathbf{R}) \\ s\cdot\mathrm{vec}(\mathbf{R}) \end{bmatrix}_{(27\times 1)} = \mathbf{0}, \tag{3.4}$$

where vec(•) represents the column-first vectorization of the matrix •. Note that at least 26 point correspondences are required here, i. e., $n \geq 26$. The solution can be found by minimizing the norm

$$\|\mathbf{B} \begin{bmatrix} \mathrm{vec}(\mathbf{E}) \\ \mathrm{vec}(\mathbf{R}) \\ s\cdot\mathrm{vec}(\mathbf{R}) \end{bmatrix}\|, \text{ subject to } \|\mathrm{vec}(\mathbf{E})\| = 1. \tag{3.5}$$

Solving a problem of this form is discussed in [67]. For the specific problem here, this work write Eq. (3.4) as

$$\mathbf{B}_{\mathrm{E}}\mathrm{vec}(\mathbf{E}) + \mathbf{B}_{\mathrm{sR}} \begin{bmatrix} \mathrm{vec}(\mathbf{R}) \\ s\cdot\mathrm{vec}(\mathbf{R}) \end{bmatrix} = \mathbf{0}, \tag{3.6}$$

where $\mathbf{B}_{\mathrm{E}}$ is the submatrix of $\mathbf{B}$ consisting of the first 9 columns, with size $n \times 9$. $\mathbf{B}_{\mathrm{sR}}$ is the submatrix of $\mathbf{B}$ consisting of the last 18 columns, with size $n \times 18$. Finding the solution that satisfies $\|\mathrm{vec}(\mathbf{E})\| = 1$ is equivalent to solving

$$(\mathbf{B}_{\mathrm{sR}}\mathbf{B}_{\mathrm{sR}}^{\dagger} - \mathbf{I})\mathbf{B}_{\mathrm{E}}\mathrm{vec}(\mathbf{E}) = \mathbf{0}, \tag{3.7}$$

where $\mathbf{I}$ is the identity matrix, and $\mathbf{B}_{\mathrm{sR}}^{\dagger}$ is the Moore-Penrose pseudo-inverse of $\mathbf{B}_{\mathrm{sR}}$. Matrix $\mathbf{E}$ is then efficiently solved by the standard SVD decomposition.

3.2.2 *Extracting Rotation, Translation and Scale*

Once $\mathbf{E}$ is obtained, the rotation and up-to-scale translation are extracted by decomposing $\mathbf{E} = [\mathbf{t}]_{\times}\mathbf{R}$. To recover the translation $\mathbf{t}$ with proper scale and the scale factor s itself, this work use the known rotation $\mathbf{R}$ and substitute it into constraint (3.2). Can obtain the following equation:

$$\mathbf{C}_{(n\times 4)} \begin{bmatrix} \mathbf{t} \\ s \end{bmatrix}_{(4\times 1)} = \mathbf{b}, \tag{3.8}$$

where $\mathbf{C}$ and $\mathbf{b}$ are composed of known quantities and the decomposed $\mathbf{R}$. The translation $\mathbf{t}$ and the scalar s are then estimated using the linear least squares method. Since there are two rotations obtained from the decomposition of $\mathbf{E}$, two pairs of solutions are obtained. The final solution is defined as the one that leads to the smallest residual errors in the original incidence relation.

3.3 LINEAR SOLVER FROM AFFINE CORRESPONDENCES

Now turn our attention to the AC-based linear solver. This work first review the constraints on the relative pose originating from the measured affine transformation, and then discuss the estimation of the generalized essential matrix as well as the individual similarity transformation parameters.

3.3.1 *Affine Transformation Constraint*

Let us denote the euclidean transformation parameters from the coordinate frame of camera C_i' to the one of camera C_i by $\mathbf{R}_{C_iC_i'}$ and $\mathbf{t}_{C_iC_i'}$. The latter are given by

$$\begin{bmatrix} \mathbf{R}_{C_iC_i'} & \mathbf{t}_{C_iC_i'} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} {\mathbf{R}_{bi}}^{\top}\mathbf{R}\mathbf{R}_{bi}' & s{\mathbf{R}_{bi}}^{\top}\mathbf{R}\mathbf{t}_{bi}' + {\mathbf{R}_{bi}}^{\top}\mathbf{t} - {\mathbf{R}_{bi}}^{\top}\mathbf{t}_{bi} \\ \mathbf{0} & 1 \end{bmatrix}.$$

The essential matrix $\mathbf{E}_{C_iC_i'}$ between the frames of cameras C_i and C_i' is therefore given as

$$\mathbf{E}_{C_iC_i'} = [\mathbf{t}_{C_iC_i'}]_\times \mathbf{R}_{C_iC_i'} = \mathbf{R}_{bi}{}^\top [\mathbf{R}_{bi}\mathbf{t}_{C_iC_i'}]_\times \mathbf{R}\mathbf{R}_{bi}', \tag{3.9}$$

where $[\mathbf{R}_{bi}\mathbf{t}_{C_iC_i'}]_\times = s\mathbf{R}[\mathbf{t}_{bi}']_\times \mathbf{R}^\top + [\mathbf{t}]_\times - [\mathbf{t}_{bi}]_\times$. The relationship between essential matrix $\mathbf{E}_{C_iC_i'}$ and the local affine transformation $\mathbf{A}_i$ is given by [13]:

$$(\mathbf{E}_{C_iC_i'}^\top \mathbf{x}_i)_{(1:2)} = -(\hat{\mathbf{A}}_i^\top \mathbf{E}_{C_iC_i'}\mathbf{x}_i')_{(1:2)}, \tag{3.10}$$

where the lower index $(1:2)$ extracts the first two equations of the equation system and $\hat{\mathbf{A}}_i = [\mathbf{A}_i, \mathbf{0}; \mathbf{0}, 0]_{3\times 3}$. By substituting (3.9) into (3.10), can get

$$\begin{aligned} &(\mathbf{R}_{bi}'^\top \mathbf{R}^\top [\mathbf{R}_{bi}\mathbf{t}_{C_iC_i'}]_\times^\top \mathbf{R}_{bi}\mathbf{x}_i)_{(1:2)} \\ =\ &-(\hat{\mathbf{A}}_i^\top \mathbf{R}_{bi}{}^\top [\mathbf{R}_{bi}\mathbf{t}_{C_iC_i'}]_\times \mathbf{R}\mathbf{R}_{bi}'\mathbf{x}_i')_{(1:2)}. \end{aligned} \tag{3.11}$$

Using (3.1), the above formula is further reformulated and expanded as follows:

$$\begin{aligned} &(\mathbf{R}_{bi}'^\top (s[\mathbf{t}_{bi}']_\times \mathbf{R}^\top - \mathbf{E}^\top - \mathbf{R}^\top [\mathbf{t}_{bi}]_\times)\mathbf{f}_i)_{(1:2)} \\ =\ &(\hat{\mathbf{A}}_i^\top \mathbf{R}_{bi}{}^\top (s\mathbf{R}[\mathbf{t}_{bi}']_\times + \mathbf{E} - [\mathbf{t}_{bi}]_\times \mathbf{R})\mathbf{f}_i')_{(1:2)}. \end{aligned} \tag{3.12}$$

Combining (3.2) and (3.12), can obtain three constraints for each affine correspondence $(\mathbf{x}_i, \mathbf{x}_i', \mathbf{A}_i)$.

3.3.2 *Essential Matrix Estimation*

Knowing that a single AC provides three constraints and given m affine correspondences, (3.4) can be written as

$$\begin{bmatrix} \mathbf{B}_{c1(m\times 27)} \\ \mathbf{B}_{c2(m\times 27)} \\ \mathbf{B}_{c3(m\times 27)} \end{bmatrix} \begin{bmatrix} \mathrm{vec}(\mathbf{E}) \\ \mathrm{vec}(\mathbf{R}) \\ s\cdot \mathrm{vec}(\mathbf{R}) \end{bmatrix}_{(27\times 1)} = \mathbf{0}. \tag{3.13}$$

Matrix $\mathbf{B}_{c1}$ contains the coefficients from (3.2), $\mathbf{B}_{c2}$ the ones from the first equation of (3.12), and $\mathbf{B}_{c3}$ the ones from the second equation of (3.12). With ACs, the minimum number of correspondences is reduced from 26 to 9, i. e., $m \geq 9$. The reduced number of required correspondences is highly beneficial in RANSAC-style robust estimation, where the run-time depends exponentially on the sample size. Matrix $\mathbf{E}$ is estimated by running standard SVD decomposition on the coefficient matrix as discussed in Sec. 3.2.1.

3.3.3 *Extracting Rotation, Translation and Scale*

The rotation can be extracted by the method presented in Sec. 3.2.2. The solution for $\mathbf{t}$ and s is slightly different since here has constraints (3.12). Thus, (3.8) becomes

$$\begin{bmatrix} \mathbf{C}_{c1(m\times 4)} \\ \mathbf{C}_{c2(m\times 4)} \\ \mathbf{C}_{c3(m\times 4)} \end{bmatrix} \begin{bmatrix} \mathbf{t} \\ s \end{bmatrix}_{(4\times 1)} = \begin{bmatrix} \mathbf{b}_{c1} \\ \mathbf{b}_{c2} \\ \mathbf{b}_{c3} \end{bmatrix}, \tag{3.14}$$

where $\mathbf{C}_{ci}$ and $\mathbf{b}_i$ correspond to the constraint i, and $\mathbf{t}$ and s can be solved by linear least squares.

3.4 MULTI-DIMENSIONAL NULL SPACE

In the experiments, this work noticed that the proposed linear solvers often fail when fewer than 5 images are used in each view-graph. Taking the solver using point correspondences (called 26PC in this chapter) as an example, Fig. 3.2 illustrates the achievable average algebraic residual errors. As can clearly be observed, for the standard solver (i. e., a single null space vector), the error deviates from zero as fewer views are used. More interestingly, the error can be reduced by considering higher dimensional null spaces. Let the dimension of the considered null space be d. Let $\mathrm{NS(d)}_{(9\times \mathrm{d})}$ be the last d columns of $\mathbf{V}$. $\mathbf{V}$ is obtained from the SVD decomposition, and the last d columns are the ones corresponding to the d smallest singular values. The definition of the minimum algebraic residual error is

$$r = \|\mathrm{NS(d)}\ \cdot \lambda - \mathrm{vec}(\mathbf{E}_{gt})\|_2. \tag{3.15}$$

$\mathbf{E}_{gt}$ represents the ground-truth essential matrix, and $\lambda \in \mathbb{R}^d$ is obtained from the linear least squares solution $\lambda = \mathrm{NS(d)}^{\dagger} \cdot \mathbf{E}_{gt}$, where $(\bullet)^{\dagger}$ represents the pseudo-inverse of matrix $\bullet$. This work performed 1000 experiments for each null space dimension and for different view-graph sizes to obtain the average value of the algebraic residual error. If the residual is numerically close to 0 in the d-dimensional case, the correct $\mathbf{E}$ matrix exists in the d-dimension null space. Consequently, it can be expressed as a linear combination of the d column vectors in $\mathrm{NS(d)}$.

As can be observed in Fig. 3.2, when the view-graph size is greater or equal to 5, the solution generally exists in a 1-dim null space and the standard SVD method solves the problem. When the view-graph size is

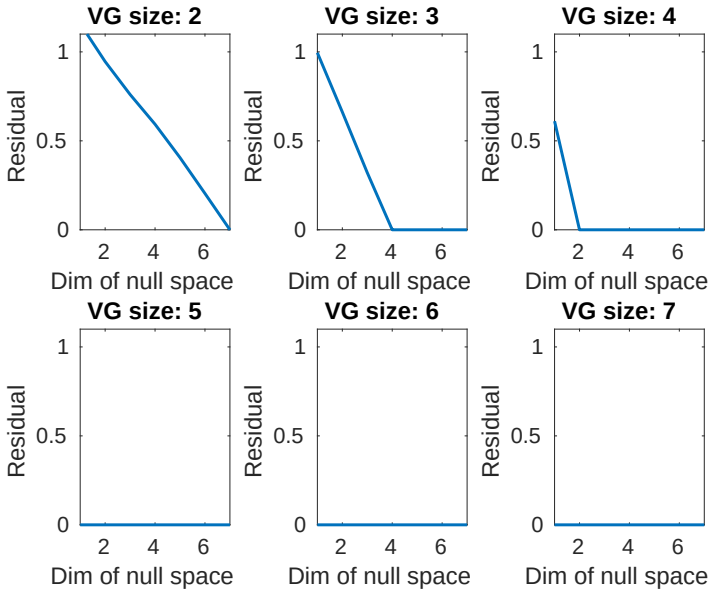


FIGURE 3.2: Effect of different null space dimensions for different view-graph sizes (VG size). Each figure indicates the achievable residual error (3.15) as a function of the null-space dimension, averaged over 1000 experiments. The experiment uses 26PC as an example, and 9AC shares the same property.

equal to 4, at least a 2-dim null space is required to generally get the solution. For a view-graph size equal to 3, the solution generally requires a 4-dim null space. For a view-graph size of 2, at least a 7-dim null space is required to generally obtain a solution. These properties equally hold for the linear solver from affine correspondences (9AC solver).

This work present an extension to the SVD method when the required null space dimension is less than or equal to 4 and greater than 2 (i. e., when the view-graph sizes are 3 or 4). Take d as 4 and the 1st, 2nd, 3rd, and 4th columns of NS(4) correspond directly to four 3×3 matrices $\mathbf{E}_1$, $\mathbf{E}_2$, $\mathbf{E}_3$, and $\mathbf{E}_4$, respectively. The final $\mathbf{E}$ is given as linear combination

$$\mathbf{E} = x\mathbf{E}_1 + y\mathbf{E}_2 + z\mathbf{E}_3 + \mathbf{E}_4. \tag{3.16}$$

for some scalars x, y, z. This problem and its solution are of the exact same form as the original five-point problem [69, 170–172]. Considering that $\mathbf{E}$ is

a 3×3 rank-2 essential matrix, we may simply use the following non-linear constraints on the essential matrix:

$$\begin{cases} \det(\mathbf{E}) = 0, \\ 2\mathbf{E}\mathbf{E}^\top\mathbf{E} - \text{trace}(\mathbf{E}\mathbf{E}^\top)\mathbf{E} = 0. \end{cases} \tag{3.17}$$

Parameters x, y and z can be solved by constructing polynomial equations from the above constraints and solving them using one of various algorithms that have been proposed in the literature [69, 170–172]. Here this work use the off-the-shelf method from Li and Hartley [170], which relies on the hidden variable resultant. After solving for x, y, and z, and substituting in (3.16), we can easily obtain the final $\mathbf{E}$.

3.5 SOLVER FOR PARTIALLY KNOWN ROTATION

In [33] a minimal solver is proposed for the generalized relative pose and scale problem assuming known vertical direction. The algorithm requires five point correspondences. In this section, This work show that two affine correspondences are sufficient to solve the problem.

3.5.1 *5PC Minimal Solver*

Reconsidering the generalized epipolar constraint with scale as proposed in [33] and using properties [25] of the scalar triple product, (3.2) is reformulated as

$$(\mathbf{f}_i^\top \times \mathbf{R}\mathbf{f}_i')^\top\mathbf{t} - s\mathbf{f}_i^\top\mathbf{R}[\mathbf{t}_{bi}']_\times\mathbf{f}_i' + \mathbf{f}_i^\top[\mathbf{t}_{bi}']_\times\mathbf{R}\mathbf{f}_i' = 0. \tag{3.18}$$

$\mathbf{t}$ and s are separated as in

$$\mathbf{m}_i^\top\tilde{\mathbf{t}} = 0, \tag{3.19}$$

where

$$\mathbf{m}_i = \begin{bmatrix} \mathbf{f}_i \times \mathbf{R}\mathbf{f}_i' \\ -\mathbf{f}_i^\top\mathbf{R}[\mathbf{t}_{bi}']_\times\mathbf{f}_i' \\ \mathbf{f}_i^\top[\mathbf{t}_{bi}]_\times\mathbf{R}\mathbf{f}_i' \end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{t}} = \begin{bmatrix} \mathbf{t} \\ s \\ 1 \end{bmatrix}. \tag{3.20}$$

Let us assume that data is prerotated such that $\mathbf{v}$ is the corresponding direction in both view-graphs. $\mathbf{v}\theta$ then becomes the axis–angle representation of the relative rotation matrix $\mathbf{R}$. The quaternion form is given by $(\cos(\theta/2), \mathbf{v}\sin(\theta/2))$. Using $\alpha = \frac{\cos(\theta/2)}{\sin(\theta/2)}$, we have

$$\mathbf{R} = 2(\mathbf{v}\mathbf{v}^\top + \alpha[\mathbf{v}]_\times) + (\alpha^2 - 1)\mathbf{I}, \tag{3.21}$$

which can be derived from the quaternion to rotation matrix formula. $[\mathbf{v}]_\times$ is the skew-symmetric matrix of the unit vector $\mathbf{v}$ and α is independent of $\mathbf{v}$. In case of known directional correspondence, the DoF is reduced from 7 to 5, and the intractable 4-parameter Quadratic Eigenvalue Problem (QEP) is reduced to a tractable 1-parameter QEP ($\mathbf{v}$ is known) and can be solved by Eigen Decomposition. Specifically, the minimal 5PC coefficient matrix $\mathbf{M}$ derived from (3.18) can be written in single-parameter form a

$$\mathbf{M} = \begin{bmatrix} \mathbf{m}_1 & \dots & \mathbf{m}_5 \end{bmatrix}^\top = \alpha^2\mathbf{A} + \alpha\mathbf{B} + \mathbf{C}, \tag{3.22}$$

where $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are 5×5 matrices formed from $\mathbf{m}_i$ in (3.19). The unknown parameters α and $\tilde{\mathbf{t}}$ in $\mathbf{M}\tilde{\mathbf{t}} = 0$ can be solved by standard Eigen decomposition of the matrix

$$\begin{bmatrix} -\mathbf{A}^{-1}\mathbf{B} & -\mathbf{A}^{-1}\mathbf{C} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}. \tag{3.23}$$

The solution is extracted from eigenvalue α for which the corresponding eigenvector is proportional to $[\alpha\tilde{\mathbf{t}}^\top, \tilde{\mathbf{t}}^\top]^\top$.

3.5.2 *2AC Minimal Solver*

Similarly, the additional constraints provided by ACs (i. e. (3.12)) can be reformulated to

$$\begin{aligned} &(\mathbf{R}_{bi}'^\top \mathbf{R}^\top [\mathbf{f}_i]_\times - \hat{\mathbf{A}}_i^\top \mathbf{R}_{bi}^\top [\mathbf{R}\mathbf{f}_i']_\times)_{(1:2)}\mathbf{t} \\ &+ s(-\mathbf{R}_{bi}'^\top [\mathbf{t}_{bi}']_\times \mathbf{R}^\top \mathbf{f}_i + \hat{\mathbf{A}}_i^\top \mathbf{R}_{bi}\mathbf{R}[\mathbf{t}_{bi}']_\times \mathbf{f}_i')_{(1:2)} \\ &+ (\mathbf{R}_{bi}'^\top \mathbf{R}^\top [\mathbf{t}_{bi}]_\times \mathbf{f}_i - \hat{\mathbf{A}}_i^\top \mathbf{R}_{bi}^\top [\mathbf{t}_{bi}]_\times \mathbf{R}\mathbf{f}_i')_{(1:2)} = \mathbf{0}. \end{aligned} \tag{3.24}$$

Based on (3.18) and (3.24), can get three constraints for each AC $(\mathbf{x}_i, \mathbf{x}_i', \mathbf{A}_i)$. So, it only takes 2 affine correspondences to solve the generalized relative pose and scale problem with known directional correspondence. (3.19) can be written as

$$\begin{bmatrix} \mathbf{M}_{c1(2\times5)} \\ \mathbf{M}_{c2(2\times5)} \\ \mathbf{M}_{c3(1\times5)} \end{bmatrix} \tilde{\mathbf{t}}_{(5\times1)} = \mathbf{0}, \tag{3.25}$$

where $\mathbf{M}_{c1}$ contains the coefficients of (3.18), $\mathbf{M}_{c2}$ the ones of the first equation of (3.24), and $\mathbf{M}_{c3}$ the ones of the second equation of (3.24). α and $\tilde{\mathbf{t}}$ remain solved by the standard eigen decomposition as discussed in Sec. 3.5.1.

3.6 EXPERIMENTS

This work compare our proposed 26PC (Sec. 3.2), 9AC (Sec. 3.3), 26PC-NS (26PC + Sec. 3.4). 9AC-NS (9AC + Sec. 3.4) and 2AC (Sec. 3.5.2) solvers with the 5PC solver from [33] and the eigen solver from [34]. This work do not compare with 3D structure-based algorithms, as they are outperformed by [33]. The eigen solver is implemented in MATLAB, while the other methods are in C++.

In the real data experiments, this work combine 2AC, 5PC, 9AC, and 26PC with a hypothesize-and-test framework, i. e., Graph-Cut RANSAC [167]. The 26PC solver is added here for refinement over all inliers both in the local optimization and after the robust estimation. Furthermore, the higher dimensional null-space solver (26PC-NS) is also added to make sure that the instabilities owing to the possibly low number of views are not impacting the quality of the result.

This work measure the rotation error $\varepsilon_{\mathbf{R}}$ by the norm of the Rodrigues vector corresponding to the residual rotation given by the product of the transpose of the ground truth (GT) rotation matrix and the estimated rotation matrix. Correspondingly, this work measure the translation error $\varepsilon_{\mathbf{t}}$ by the norm of the difference between GT and estimated translation vectors. This work also report the relative scale error ε_s.

Synthetic Scene. This work generate 5 cameras randomly placed in the cube $[-2, 2]^3$ and 50 3D points randomly sampled in the cube $[-1, 1]^2 \times [2, 20]$. Correspondences are generated from rays that point to the same 3D point. Translation $\mathbf{t}$ and rotation axis $\mathbf{v}$ are randomly generated in the unit cube, followed by normalization of the rotation axis. The rotation angle and scale s are randomly picked in the interval $[0, \pi]$ and $[0.1, 5.0]$, respectively. Affine transformations are generated in a similar manner to [13]. Each method uses the minimum number of correspondences required, and the eigen solver [34] uses 10. This work use $6 + 4 \times 5$ correspondences per camera for 26PC (26PC-NS), $1 + 4 \times 2$ correspondence per camera for 9AC (9AC-NS), one correspondence per camera for 5PC, two correspondences per camera for the eigen solver, and only two views with a unique correspondence for 2AC.

Numerical Stability Analysis. This work first test the numerical stability of all the solvers without noise. To illustrate the numerical stability, this work present the distribution of the maximal value ε of rotation error, translation error and scale error in Fig. 3.3. Each method was run on 5000 random problem instances. All the proposed solvers and 5PC are stable, with ε being

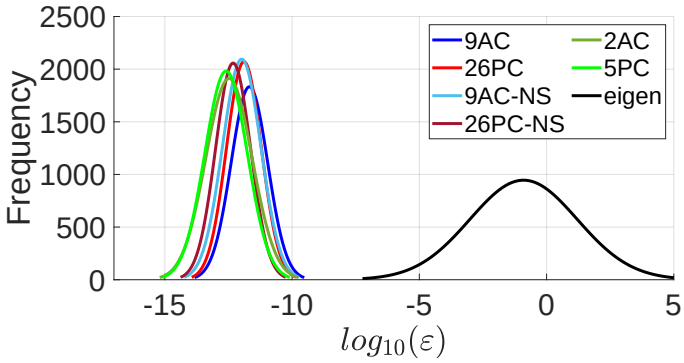


FIGURE 3.3: Histogram of $\log_{10}$ errors in the noise-free case.

smaller than 10^{-9} in over 99% trials. However, since the eigen solver [34] is an optimization-based algorithm, it relies heavily on reasonable initial values. Therefore, it exhibits inferior numerical stability. Furthermore, it uses a multi-start clustering technique implemented in MATLAB, thus making it very time-consuming with a runtime of $\sim 10^6$ μs. The runtime of other algorithms remains within the same order of magnitude. The times of 26PC, 9AC, 26PC-NS, 9AC-NS, 2AC, 5PC [33] are 186 μs, 229 μs, 220 μs, 245 μs, 52 μs, and 62 μs. The multi-dimensional null space-based solvers did not increase the computational burden. 5PC and 2AC with partially known rotation take less time as they need to solve for fewer unknowns.

Experiments on Synthetic Data. In this section, we investigate the performance of the solvers in simulation experiments. To add noise to the PCs, this work first extract an orthogonal plane from each bearing vector and add random noise in pixels by assuming a focal length of 800. The perturbed points in the plane are renormalized to obtain perturbed bearing vectors [146]. For noise on ACs, this work add random Gaussian noise to the affine parameters with standard deviation σ as done in [173]. This work randomly transform the rotation axis $\mathbf{v}$ by a rotation with maximum angle 0.05°.

This work analyze the point noise levels from 0 to 2 pixels and the affine noise levels from 0 to 5×10^{-3} separately in Fig. 3.4. As expected, the 9AC solver is more robust to pixel noise than 26PC for both general and null-space solvers. For the eigen solver, please note that this work do not show its mean error in $\mathbf{t}$ and s because they exceed $4m$ and 1, respectively. With the rotation axis known, although 2AC has a marginally larger mean

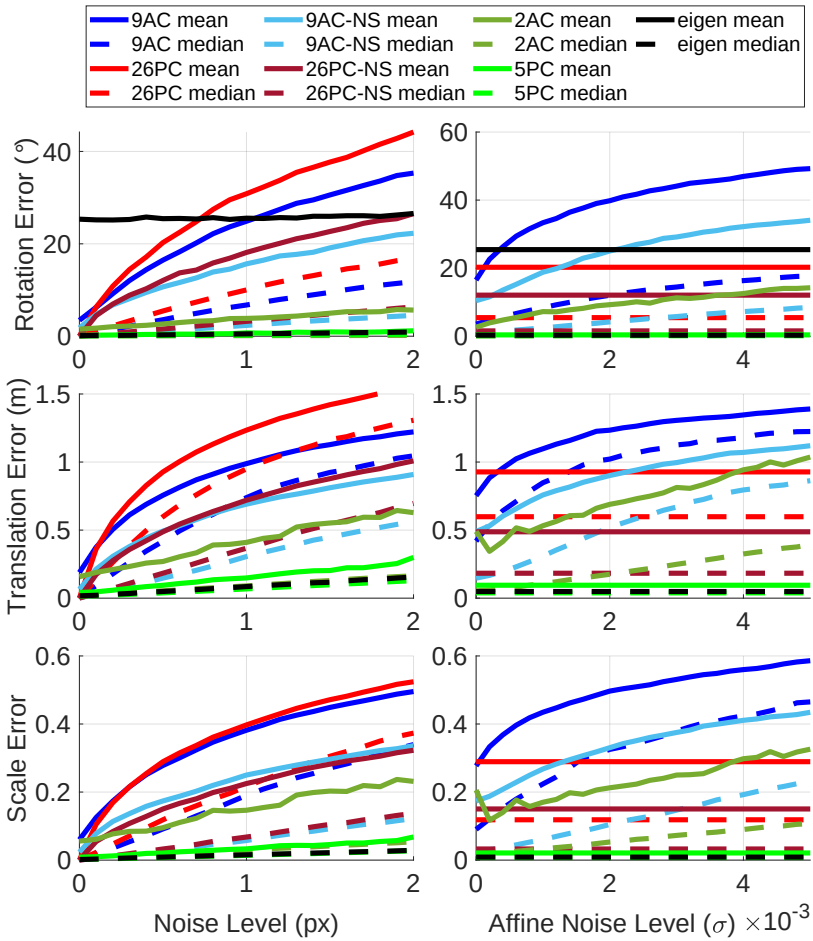


FIGURE 3.4: Mean and median errors of the generalized relative pose and scale solvers. *Left*: error as a function of the image noise added to the point coordinates with fixed 10^{-5} affine noise. *Right*: error as a function of the affine noise with fixed 0.5 px image noise. Note that for better visualization, this work modify the limits of y-axis and eigen solver's mean errors are too large to be shown.

Minimal solver	Non-minimal solver	AUC@ 0.25m/2°	AUC@ 0.5m/5°	AUC@ 1m/10°	$\varepsilon_{\mathbf{R}}$ (°) Med	$\varepsilon_{\mathbf{t}}$ (m) Med	ε_S Med	Iterations Avg	Runtime (s). Avg
2AC	26PC	0.38	0.66	0.72	13.06	0.16	0.23	14	**0.00**
	26PC-NS	**0.63**	**0.78**	**0.84**	**3.27**	**0.07**	**0.08**	**13**	0.01
5PC	26PC	0.11	0.45	0.65	13.06	0.28	0.19	24	**0.04**
	26PC-NS	**0.42**	**0.62**	**0.62**	**3.10**	**0.14**	**0.03**	**23**	0.09
9AC	26PC	0.20	**0.39**	**0.47**	13.06	0.30	0.15	27	**0.68**
	26PC-NS	**0.32**	0.32	0.32	**2.19**	**0.06**	**0.02**	**6**	1.08
26PC	26PC	0.14	0.55	**0.67**	13.06	0.26	0.13	3737	2.34
	26PC-NS	**0.55**	**0.67**	**0.67**	**1.90**	**0.10**	**0.03**	**1469**	**0.76**

TABLE 3.1: AUC scores at 0.25m/2°, 0.5m/5°, and 1m/10°, the median rot. ($\varepsilon_{\mathbf{R}}$; degrees), trans. ($\varepsilon_{\mathbf{t}}$, degrees), relative scale errors ($\varepsilon_{\mathbf{s}}$), avg. iteration number and run-time (seconds) of GC-RANSAC [167] combined with different minimal and non-minimal solvers on the EuRoC dataset [174] with a view-graph size of 4. The experiments are repeated 100 times on each sequence pair, randomly selecting retrieval images by [174]. The absolute best results are underlined.

error than 5PC, it has comparable median errors. This is understandable due to the affine part being usually more noise-sensitive [15], and 2AC uses less geometric information than 5PC. Its main benefit is in the reduced sample size, which is of utmost importance in robust estimation. Another key observation is that null space solvers significantly improve the stability of general solvers.

Please note that since the actual noise distributions for ACs are unknown in practice, the only firm conclusions this work can draw from these experiments are as follows: (i) both the PC and AC-based solvers act reasonably w.r.t. increasing noise levels; (ii) the null-space trick improves stability.

Experiments on Real-world Data. This work validate the proposed algorithms on the EuRoC [174] and the TUM RGB-D benchmarks [175]. Due to its high run-time, this work do not use the eigen solver [34] in real data experiments.

This work first focus on registering pairs of trajectories captured by two agents navigating the same environment. For this purpose, this work use the EuRoC dataset and run ORB-SLAM [56] on each sequence to get trajectories, thereby obtaining up-to-scale extrinsic calibrations among the images within the sequence. This work calculate the scale to ground truth for both trajectories and then take the ratio, which gives us the ground truth scale. To generate inter-sequence correspondences, this work perform image retrieval by [176] and find four pairs of images, connecting a sequence pair. Here use DoG-AffNet-HardNet [163, 164] to obtain tentative affine

Minimal solver	Non-minimal solver	AUC@ 0.25m/2°	AUC@ 0.5m/5°	AUC@ 1m/10°	$\varepsilon_{\mathbf{R}}$ (°) Med	$\varepsilon_{\mathbf{t}}$ (m) Med	ε_s Med	Iterations Avg	Runtime (s). Avg
2AC	26PC	0.17	0.37	0.61	9.72	0.32	0.03	**8**	**0.01**
	26PC-NS	**0.54**	**0.70**	**0.83**	**1.70**	**0.08**	**0.02**	**8**	**0.01**
5PC	26PC	0.17	0.37	0.61	9.75	0.31	0.03	**14**	**0.04**
	26PC-NS	**0.52**	**0.68**	**0.81**	**1.71**	**0.09**	**0.02**	**14**	0.07
9AC	26PC	0.19	0.39	0.62	9.74	0.31	0.03	435	**0.18**
	26PC-NS	**0.55**	**0.71**	**0.83**	**1.45**	**0.08**	**0.02**	**135**	0.23
26PC	26PC	0.19	0.38	0.62	9.92	0.31	0.03	4171	**1.61**
	26PC-NS	**0.50**	**0.66**	**0.80**	**2.09**	**0.09**	**0.02**	**3476**	1.81
3D-3D	3D-3D	0.28	0.43	0.55	4.14	0.48	0.21	2570	1.74

TABLE 3.2: AUC scores at 0.25m/2°, 0.5m/5°, and 1m/10° (translation/rotation error), median rot. ($\varepsilon_{\mathbf{R}}$; degrees), trans. ($\varepsilon_{\mathbf{t}}$, meters), relative scale errors (ε_s), avg. iteration number and run-time (seconds) of GC-RANSAC [167] combined with different minimal and non-minimal solvers on sequences *freiburg2_xyz* (1758 pairs) and *freiburg2_rpy* (1568 pairs) from the TUM-RGBD dataset [175] with a view-graph size of 4. The absolute best results are underlined.

correspondences between the retrieved image pairs. Finally, this work use GC-RANSAC [167] as a robust estimator and apply one of the tested solvers and either the 26PC or the 26PC-NS non-minimal solvers as a refinement over all inliers.

Table 3.1 reports the AUC scores (Area Under the recall Curve) thresholded at 0.25m/2°, 0.5m/5°, and 1m/10°, the median rotation and translation errors, the average numbers of iterations and run-times in seconds. In all cases, the null space trick (26PC-NS) significantly improves the median errors compared to using the simple 26PC solver in the local optimization. It doubles the AUC@0.25m/2° scores and, in most cases, substantially improves other metrics without severely increasing the run-time. The reduced iteration number also shows that the 26PC-NS solver obtains more stable results than 26PC, thus triggering the RANSAC termination criterion early. The proposed 2AC solver obtains the highest AUC scores, and its median errors are on par with the other methods. Moreover, it runs for only a few tens of milliseconds, being the fastest alternative. 2AC is substantially more accurate than its direct competitor, 5PC, which also relies on gravity direction. In general, the table shows that using ACs and additional prior information greatly impacts the robust estimation run-time, leading to real-time performance. Also, it highlights the importance of the 26PC-NS solver for non-minimal model estimation.

Next, this work show experiments on the TUM RGB-D dataset [175]. This work cut each sequence into multiple pieces to form pairs. The process is

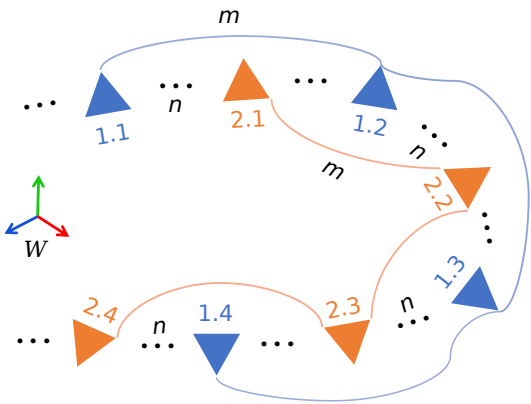


FIGURE 3.5: Design of real data experiments for a sequence in the TUM RGB-D dataset [175]. For *i.j*, *i* is the index of the view-graph, and *j* is the index of the frame. Feature matching runs between the *j*th frame of the two adjacent view-graphs. The distance between consecutive frames is set by taking every *m*th frame and *n* determines the distance between the two view-graphs.

visualized in Fig. 3.5. Starting from an initial frame, take every *m*th frame as part of view-graph 1. Starting from the *n*th frame ($n < m$), take every *m*th frame to form views of view graph 2. Here index views with *i.j*, where *i* is the view-graph and *j* is the frame index within the view-graph. This work match features between the *j*th frames of the view-graphs, and transform landmark observation vector $\mathbf{f}$ of each frame to world coordinate frame *W* using the GT rotation of each frame. Again, assuming the world frame as the reference of each view graph, the GT translations of each frame are used as camera offsets. Finally, the bearing vectors and camera offsets in the second view-graph are expressed w.r.t. a randomly translated, rotated, and scaled reference frame to alter the GT transformation away from a simple identity transformation. Translational displacements are picked within the range [-1,1], and the scale is picked from the interval [0.2, 5]. This work set $m = 50$ and view-graph sizes to 4 and $n = 2$.

To demonstrate the 3D registration is inferior for such scenarios, as shown in [33], this work run GC-RANSAC on 3D matches. Correspondences are obtained by randomly selecting two views from each view graph and matching all four resulting views. 3D points are obtained by triangulating the intra-sequence 2D matches.

Note that only the TUM dataset has 3D-3D results, which is due to the experimental setup of the TUM dataset where the images are relatively close to each other. In contrast, the image pairs chosen in the EuRoC dataset are too far apart, hence, triangulation is not feasible.

The results are reported in Table 3.2 and show the same trends as on the EuRoC dataset. The null space trick substantially improves the results in all cases. The best results are obtained by the 2AC and 9AC solvers, and 2AC-based estimation runs in real-time. All methods, except 3D-3D, achieve similar scale accuracy. The 3D-3D algorithm leads to the lowest accuracy compared to any solvers using 26PC-NS in the local optimization.

3.7 CONCLUSIONS

This work propose the first closed-form solutions to the generalized relative pose and scale problem, which permits the registration of calibrated sets of views directly from 2D-2D correspondences. The proposed 26PC-NS solver proved crucial for the local optimization in a state-of-the-art locally optimization RANSAC for achieving high accuracy, independently of the employed minimal solver. Leveraging affine correspondences, the proposed minimal solvers, 2AC and 9AC, achieve the most accurate results while allowing the robust estimation to run only for a few tens of milliseconds, being real-time. The relevance of our contribution is further underlined by a favorable comparison against the much simpler 3D-3D registration alternative. Besides the large accuracy difference on the TUM-RGBD dataset compared to 3D-3D, the proposed approaches do not need to store and maintain large point cloud maps of the environment, thus providing a lightweight alternative to the registration problem.

4

DL FOR IMPROVED POLYNOMIAL EQUATION SOLVING

Geometric closed-form solvers represent the cornerstones of structure from motion, as they permit the initialization of both intrinsic [43] or extrinsic camera parameters [41, 171] when no prior information is available. Given algebraic incidence relationships, the derivation of a closed-form solver usually starts by applying variable elimination techniques, thus leading to an initial system of polynomial constraints in typically few unknowns. A meanwhile established methodology of solving such polynomial systems of equations relies on algebraic geometry and the Gröbner basis method. The present chapter addresses the stability and accuracy of Gröbner basis solvers.

In simple terms, the Gröbner basis method proceeds by iteratively generating new polynomials (so-called *Syzygies*) vanishing on the original variety, each time adding them to the set of ideal generators if their reduction by the already existing generators does not lead to a zero remainder. The procedure is repeated until all possible polynomial reductions have taken place. The success of the Gröbner basis method relies on the insight that the expensive search for the basis does not need to be repeated for each new instance of a certain type of problem; the sequence of the generated polynomials remains the same for each *general* instance of a problem. As a result, efficient solvers are generated by translating the sequence of polynomial generations into an *elimination template*. At online stage, the elimination template is filled with all initial polynomials of a specific problem instance as well as their required multiplications by monomials, and then subjected to for example a Gauss-Jordan elimination. For more details on the Gröbner basis method as well as automatic solver generation, the reader is kindly referred to [36], [37] and [39]. Here we simply note that the elimination template for a particular problem typically remains fixed at online stage.

It is well-known that the complexity of finding a Gröbner basis depends on a number of factors. To start with, an effort has to be made to find a good parametrization for which the order of the equations and the remaining number of unknowns are kept as low as possible. The next step consists of finding suitable variable and monomial orderings for efficient retrieval of the Gröbner basis (and thus a compact size of the elimination template). Even though there are infinitely many monomial orderings, there are only

finitely many (reduced) Gröbner bases corresponding to a given set of equations [177]. Among these, the one corresponding to the graded-reverse lexicographical (*grevlex*) ordering is typically considered a good choice. On the other hand, [52] considers searching among all these finitely many Gröbner bases in pursuit of an optimal choice. The primary motivation of this work is given by the fact that—for one and the same instance of a certain type of problem—there may be different elimination templates with comparable computational complexity but substantial differences in the numerical stability of the retrieved solutions. Deciding for a single elimination template at offline stage therefore appears as a weakness, and a smart way of picking one of several comparable elimination templates at online stage may lead to substantial benefits in terms of the robustness and accuracy of the solutions. This work have two main contributions:

- This work demonstrate that simple variable reordering potentially has a significant impact on the robustness and accuracy of the solutions. We furthermore show that for a large class of common polynomial problems (including dense polynomial systems) variable reordering simply translates into a permutation of the columns of the original coefficient matrix, thus enabling one and the same elimination template to be used in as many ways as there are permutations of the input variables. We furthermore demonstrate that such column permutations potentially have a significant impact on the quality of the solutions. The result applies to a large class of polynomial problems that includes many 3D rotation-based solvers in geometric vision.

- This work demonstrate that the coefficients of the original set of polynomials do in fact contain sufficient information to train an artificial neural network that is able to predict which of many possible elimination templates will lead to a stable result. The networks are typically small and their inference represents an insignificant computational overhead compared to the actual elimination template. In other words, they are amenable to an online selection of the elimination template. In the case of permutation invariant elimination templates, they predict which column permutation to choose.

This work demonstrate the viability of the approach and its potential to improve solver stability at the hand of general dense polynomial problems as well as a popular solver for finding the absolute pose of a camera from a single image. The chapter is organized as follows. Section 4.1 introduces back-ground theory and the permutation invariance of elimination tem-

plates for a certain class of problems. Section 4.2 then depicts the details of our online classifier including training from purely synthetic data. Section 4.3 finally presents our results on multiple solvers, followed by a brief discussion.

4.1 PERMUTATION INVARIANT POLYNOMIAL SYSTEMS

The present chapter looks at a special class of polynomial systems for which the *support*[1] is invariant with respect to variable permutations. In the continuation, this work describe such systems as *permutation invariant*. The present section introduces some necessary notations and defines permutation invariant polynomials and polynomial systems. After demonstrating the potential impact of different variable orderings on solver stability, we will then see how—in the case of permutation invariant polynomials—variable reordering can be translated into column permutations of the original coefficient matrix, thus enabling one and the same elimination template to be reused in different ways, notably one for each variable ordering.

4.1.1 *Notations*

Let $\mathbf{x} = \{x_1, \dots, x_n\}$ be the set of variables involved in a polynomial problem defined over the polynomial ring $\mathbb{C}[\mathbf{x}]$. Let $\{f_1, \dots, f_m\}$, $f_j \in \mathbb{C}[\mathbf{x}]$ furthermore be the original set of polynomials defining the ideal $I = \{\sum_j h_j f_j \mid h_j \in \mathbb{C}[\mathbf{x}]\}$ for which we want to retrieve the zero-dimensional variety $\{\mathbf{x} \in \mathbb{C}^n, \text{ s.t. } f_j(\mathbf{x}) = 0, j = 1, \dots, m\}$[2]. Each polynomial is of the form

$$f_j = \sum_{i=1}^{k} c_{ji} \mathbf{x}^{\boldsymbol{\alpha}_i}, \tag{4.1}$$

where c_{ji} is assumed to be a coefficient drawn from the field of real numbers $\mathbb{R}$, and $\boldsymbol{\alpha} = \{\alpha_1, ..., \alpha_n\}$ denotes the set of exponents of a particular monomial such that

$$\mathbf{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1} \cdots x_n^{\alpha_n}. \tag{4.2}$$

In a slight abuse of notation, $\mathbf{x}$ and $\boldsymbol{\alpha}$ will in the following be used interchangeably to denote either the ordered set as defined above, or a column vector composed of the same elements in the same order. Now let $\mathbf{P}$

1 The *support* of a polynomial is given by the set of its monomials.

2 Note that we make the assumption that—at least in the complex field $\mathbb{C}$—the ideal properly defines a zero-dimensional variety, i.e. a non-empty finite set of points.

be an $n \times n$ permutation matrix representing a permutation π of n symbols. Then for a polynomial f_j as in (4.1), we can obtain a new polynomial, denoted $\pi(f_j)$, by permuting the multi-exponents of f_j according to $\mathbf{P}$, i.e.,

$$\pi(f_j) = \sum_{i=1}^{k} c_{ji} \mathbf{x}^{\mathbf{P}\boldsymbol{\alpha}_i}. \tag{4.3}$$

3

4.1.2 *Permutation-invariant polynomials*

Definitions: Let $\mathbf{m} = \{\mathbf{x}^{\boldsymbol{\alpha}_1}, \ldots, \mathbf{x}^{\boldsymbol{\alpha}_k}\}$ be the support of the polynomial. This work define a polynomial in variables $\mathbf{x} = \{x_1, \ldots, x_n\}$ to be *permutation-invariant* under a permutation matrix $\mathbf{P}_{n\times n}$ if and only if every element of

$$\mathbf{m}' = \{\mathbf{x}^{\mathbf{P}\boldsymbol{\alpha}_1}, \ldots, \mathbf{x}^{\mathbf{P}\boldsymbol{\alpha}_k}\} \tag{4.4}$$

is also contained in the original set $\mathbf{m}$, i.e. $\mathbf{x}^{\mathbf{P}\boldsymbol{\alpha}_i} \in \{\mathbf{x}^{\boldsymbol{\alpha}_1}, \ldots, \mathbf{x}^{\boldsymbol{\alpha}_k}\}, \forall i$. Similarly, this work define a polynomial system to be permutation invariant if all composing polynomials are permutation invariant for themselves, and a set of exponent vectors $\mathbf{o} = \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k\}$ is permutation invariant if $\mathbf{P}\boldsymbol{\alpha}_i \in \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k\}, \forall i$.

The following are a few important examples of polynomials which all comply with this definition:

- *Symmetric polynomials*: Polynomials that simply do not change under a permutation. For example, given the polynomial $f(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + 1$ and a permutation $\mathbf{P} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$, we obtain $f(\mathbf{x}) \xleftrightarrow{\mathbf{x}^{\boldsymbol{\alpha}_i} \leftarrow \mathbf{x}^{\mathbf{P}\boldsymbol{\alpha}_i}} x_4^2 + x_2^2 + x_1^2 + x_3^2 + 1$. The resulting polynomial is the same.
- *Dense polynomials*: Polynomials in which all monomials up to a certain degree appear. For example, given the polynomial $f(\mathbf{x}) = c_1 x_1^2 + c_2 x_1 x_2 + c_3 x_2^2 + c_4 x_1 + c_5 x_2 + c_6$ and a permutation $\mathbf{P} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, we obtain $f(\mathbf{x}) \xleftrightarrow{\mathbf{x}^{\boldsymbol{\alpha}_i} \leftarrow \mathbf{x}^{\mathbf{P}\boldsymbol{\alpha}_i}} c_1 x_2^2 + c_2 x_2 x_1 + c_3 x_1^2 + c_4 x_2 + c_5 x_1 + c_6$. While the coefficients of identical monomials are changing, the support of

3 Note here that this is equivalent to a permuting the polynomial variables $x_1, \ldots, x_n$ in f_j according to the inverse permutation π^{-1}.

the polynomials remains unchanged. An example application that features both dense and symmetric polynomials is given by shuffled linear regression [178, 179].

- *Degree-wise dense polynomials*: Essentially same as dense polynomials, except that they do not necessarily contain all possible monomials up to a certain degree, but simply all possible monomials of certain degrees.
- *Special cases*: Consider the bi-variate examples $f_1(\mathbf{x}) = c_1x_1^2 + c_2x_2^2 + c_3x_3^2 + 1 \in \mathbb{C}[x_1, x_2, x_3]$ and $f_2(\mathbf{x}) = c_1x_1^2x_2 + c_2x_1x_2^2 \in \mathbb{C}[x_1, x_2]$. According to the above definition, they can also be defined as permutation-invariant. The coefficients of the monomials may change while supports remain unchanged.

4.1.3 *Impact of variable reordering*

Before this work can explain the relevance of permutation-invariance in the context of a Gröbner basis solver, we first need to understand the potential impact of variable reordering on the numerical conditioning of the problem. Suppose we are given the following polynomial problem of 3 equations in 3 unknowns, and notably in grevlex ordering based on a variable ordering of $x_1 > x_2 > x_3$:

$$\mathbf{c}_1x_3^2 + \mathbf{c}_2x_2^2 + \mathbf{c}_3x_1^2 + \mathbf{c}_4x_3 + \mathbf{c}_5x_2 + \mathbf{c}_6x_1 + \mathbf{c}_7 = 0, \tag{4.5}$$

where $\mathbf{c}_i$ are 3×1 vectors of coefficients. Generating a Gröbner basis solver for this problem would lead to a certain elimination template. Choosing the different variable ordering $x_2 > x_1 > x_3$ will lead to the system

$$\mathbf{c}_1x_3^2 + \mathbf{c}_3x_1^2 + \mathbf{c}_2x_2^2 + \mathbf{c}_4x_3 + \mathbf{c}_6x_1 + \mathbf{c}_5x_2 + \mathbf{c}_7 = 0. \tag{4.6}$$

Let us ignore for a moment that the supports remain unchanged due to the permutation-invariance property. Changing the variable ordering generally leads to a different Gröbner basis and elimination template, and therefore also numerical behavior. As a concrete example, let us consider the geometric vision problem presented in [41]. The constraints are given by 4 at most cubic polynomials in 4 unknowns, and the original solver relies on a single elimination template of dimension 141×149. By applying all possible 24 variable orderings and reducing the corresponding elimination templates, we can obtain 24 results for each instance of a problem. Figure 4.1

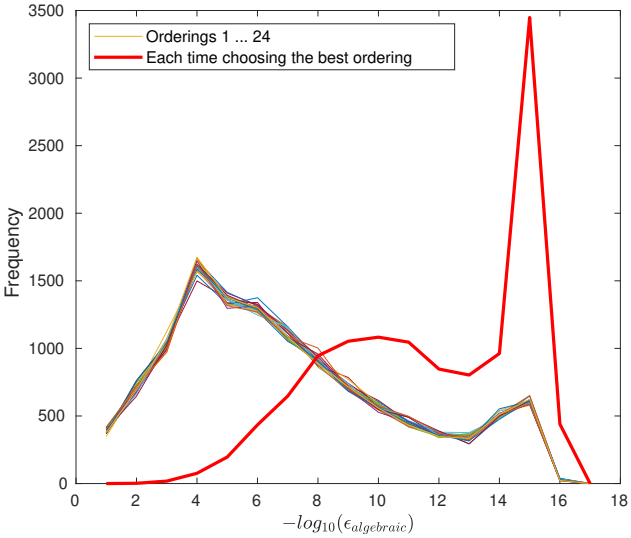


FIGURE 4.1: Error distribution over many random instances of the UPnP problem. Each curve represents the distribution obtained by one of the variable orderings. The red curve furthermore indicates the obtainable result if—for each instance—the best variable ordering is chosen.

finally illustrates error distributions for many random problem instances for each of the 24 possible variable orderings. It furthermore shows one more error distribution of a solution where—for each problem instance—the best of the 24 variable orderings is selected. As expected, on the average each permutation is equally good. However, for each coefficient instance, selecting the best permutation has a clear impact on the quality of the solution.

4.1.4 *Variable reordering and permutations*

Before proceeding, let us introduce further notations which are closely related to the elimination template itself. The coefficients of our polynomials $f_j, j = 1, \ldots, m$ may be grouped in the rows of a coefficient matrix $\mathbf{C}_{m \times h}$, where h represents the number of distinct monomials appearing in the

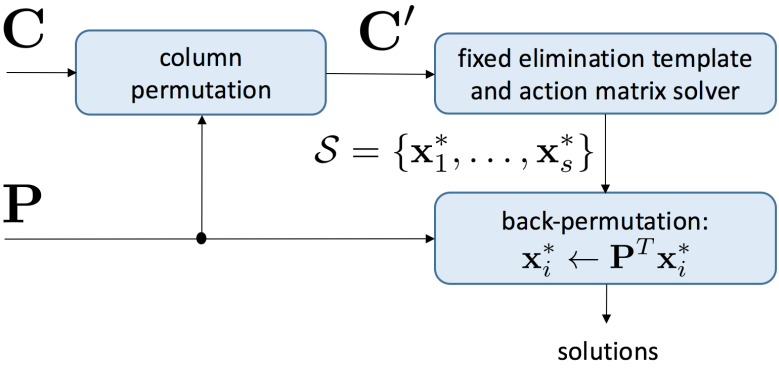


FIGURE 4.2: Permutation of the coefficient matrix $\mathbf{C}$ and back-permutation of the solutions $\mathcal{S} = \{\mathbf{x}_1^*, \ldots, \mathbf{x}_s^*\}$ for a potential improvement of the numerical stability of an elimination template.

entire system of equations. We may also redefine $\mathbf{m}$ as a column-vector containing these monomials, as in

$$\mathbf{m} = [\mathbf{x}^{\boldsymbol{\alpha}_1} \ldots \mathbf{x}^{\boldsymbol{\alpha}_h}]^T . \tag{4.7}$$

Each column $\mathbf{c}_i$ of $\mathbf{C}$ therefore contains the (potentially partially zero) coefficients of monomial $\mathbf{x}^{\boldsymbol{\alpha}_i}, i = 1, \ldots, h$, which lets us rewrite the original polynomial constraints as

$$\mathbf{C}\mathbf{m} = \mathbf{0}. \tag{4.8}$$

To conclude, this work assume that the ordering of the monomials in $\mathbf{m}$ obeys a total monomial ordering (e.g. grevlex), and also redefine the set $\mathbf{o} = \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_h\}$ such that it contains the exponent vectors of all monomials, i.e. the ones in $\mathbf{m}$. Note that the rows and columns of $\mathbf{C}$ are in fact a subset of the rows and columns in the overall elimination template, and they contain all original coefficients.

The main insight is given by the fact that—in the case of permutation invariant polynomials—the elimination templates corresponding to all variable orderings are in fact the same. This is easily recognized by applying a simple permutation of the unknowns to undo the variable reordering. Taking (4.6) as an example, applying the permutation $\mathbf{P} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ to $\mathbf{x}$ would re-establish a polynomial system of exactly same form and order as (4.5), except that the coefficient column-vectors would have been permuted. The same remains true more generally for any coefficient matrix

$\mathbf{C}$ for which the set $\mathbf{o}$ is permutation invariant. Note that—in accordance with our definition of a permutation invariant polynomial system—if all individual polynomials f_i are permutation invariant, the set $\mathbf{o}$ must be so as well. As illustrated in Figure 4.2, it is therefore possible to solve a permutation invariant polynomial system in as many ways as there are variable permutations. All that needs to be done is a back-permutation of each identified solution. The important point is that, despite the use of only a single elimination template, different permutations potentially lead to different numerical stability pretty much in the same way different matrix factorisations in linear algebra would yield different numerical solutions to the same system of linear equations.

The remaining question is how to permute the columns of the coefficient matrix. Let $\mathbf{C}' = [\mathbf{c}'_1 \dots \mathbf{c}'_h]$ be the target coefficient matrix with the permuted columns, and $\mathbf{P}$ be the permutation matrix. It can be obtained from the original coefficient matrix $\mathbf{C}$ by applying

$$\mathbf{c}'_i = \mathbf{c}_{\text{findindex}(\mathbf{P}^T \boldsymbol{\alpha}_i, \mathbf{o})}, \tag{4.9}$$

where $\text{findindex}(\boldsymbol{\alpha}, \mathbf{o})$ is a function that returns the index of $\boldsymbol{\alpha}$ inside the set $\mathbf{o}$.

4.2 ONLINE SELECTION VIA DEEP LEARNING

The previous section explained how for permutation invariant polynomial systems, a single elimination template can be used in many ways to retrieve the solution, each one potentially leading to different numerical accuracy. The present section addresses the question of how a good permutation for a particular instance of a problem can be found upfront with only very little computational overhead. This work start by seeing the basic idea which consists of applying a classifier that is able to select a good permutation at online stage. The remainder of the section then addresses the question of how to train such a classifier.

4.2.1 *Basic approach*

The basic idea consists of simply adding a classifier which is able to predict a good permutation $\mathbf{P}^*$ directly from the original coefficients. The modified flow-chart of the solver is depicted in Figure 4.3.

The classifier is a simple four-layer neural network that takes the vectorized coefficients $\text{vec}(\mathbf{C})$ as an input, and produces $n!$ output signals, each

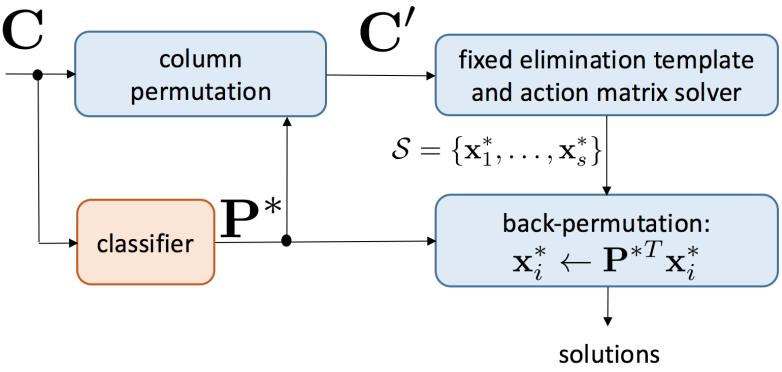


FIGURE 4.3: Permutation-aware Gröbner basis solver with added classifier able to predict a good permutation $\mathbf{P}^*$ directly from the original coefficient matrix $\mathbf{C}$.

one approximating the rank of a certain permutation by a number between 0 and 1 (0=worst, 1=best). The architecture of the network is illustrated in Figure 4.4. All layers are fully connected. Each of the three latent layers contains 500 neurons, and therefore has an output dimension of 500. Input dimensions are 500, except for the first layer, which contains $m \times h$ input variables. All activation functions are set to rectified linear units, and batch normalization is added during training. The output layer has 500 inputs and $n!$ outputs, and uses sigmoid activation functions to enforce rank numbers $p_i \in [0, 1]$. The outputs are concatenated into a float vector $\mathbf{p} = [p_1 \ldots p_{n!}]^T$. The chosen permutation $\mathbf{P}^*$ is the one corresponding to the largest element in $\mathbf{p}$.

4.2.2 *Training procedure*

The training procedure is relatively straightforward and relies on synthetic training dataset generation. For a given polynomial problem in n variables, this work will start by sampling coefficient matrices $\mathbf{C}$. Note that in practically all scenarios, this process can be done very efficiently. For example, if talking about a polynomial problem with arbitrary independent coefficients, the input coefficient matrix $\mathbf{C}$ can simply be chosen randomly. A more complicated case may be given by camera calibration problems for which the coefficients are no longer fully independent. However, it is easy to generate valid coefficient matrices through automatic simulators that

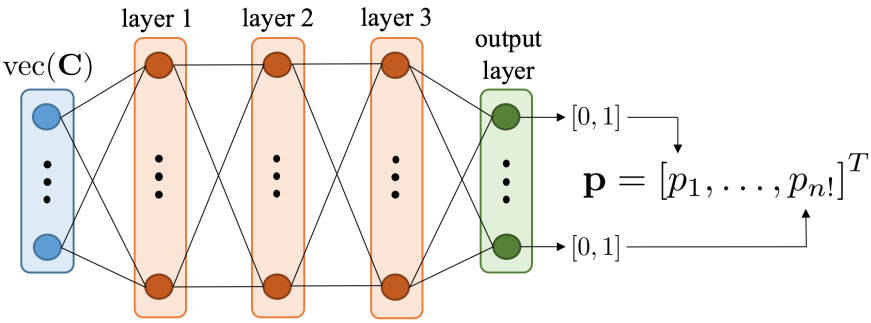


FIGURE 4.4: Architecture of our neural network for predicting and selecting a good permutation.

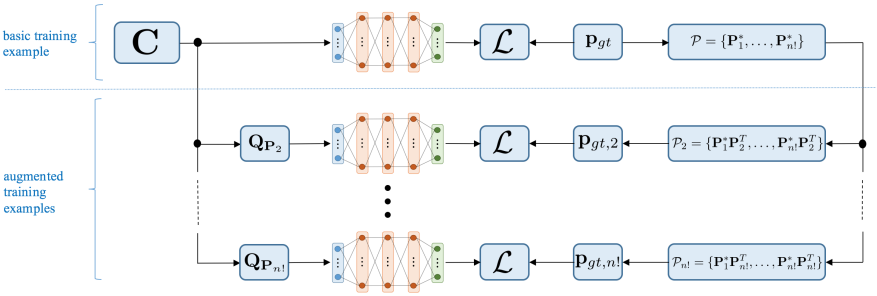


FIGURE 4.5: Training data augmentation for permutation-invariant classification performance.

start from random scenes and random camera calibration parameters, and then apply forward projection to find geometrically consistent coefficients.

For each sample coefficient matrix $\mathbf{C}$, the corresponding ground-truth training vector $\mathbf{p}_{gt}$ is then generated by brute-force looping through all possible $n!$ permutations, each time applying the solution strategy outlined in Figure 4.2. The solutions for each permutation are then back-substituted into the original polynomial constraints f_j in order to obtain the mean of the absolute algebraic residuals of each polynomial. The ground-truth vector $\mathbf{p}_{gt}$ is produced by ranking all solutions and thereby distributing the values $\frac{0}{n!-1}$ (worst) $\ldots$ $\frac{n!-1}{n!-1}$ (best) to each permutation. The training itself minimizes the MSE between $\mathbf{p}$ and $\mathbf{p}_{gt}$, as in $\mathbb{L} = \frac{1}{n!}\|\mathbf{p} - \mathbf{p}_{gt}\|_2^2$. The batch size is set to 128, and this work use the Adam solver with parameters set to $lr = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$.

4.2.3 *Permutation-invariant classification*

A remaining problem with the classifier of the previous section is that if the optimal permutation for $\{f_j\}$ is $\mathbf{P}^*$, and π is some permutation represented by matrix $\mathbf{P}$, then the predicted permutation for $\{\pi(f_j)\}$ need not be $\mathbf{P}^*\mathbf{P}^T$. Note that the latter would have been precisely the case for a permutation invariant classifier. This work implicitly enforce the permutation invariance of our classifier via simple training data augmentation.

More precisely, let $\mathbf{Q_P}$ be the permutation induced to the monomials $\mathbf{m}$ by a permutation $\mathbf{P}$. Then there is a permutation $\mathbf{Q}_\mathbf{P}^T$ induced from the right to $\mathbf{C}$, so that $\mathbf{C}\mathbf{Q}_\mathbf{P}^T$ is the coefficient matrix of $\{\pi(f_j)\}$ with respect to the monomials $\mathbf{Q_P}\mathbf{m}$. Then for each basic training example $\{\mathbf{C}, \mathbf{p}_{gt}\}$, this work add $n! - 1$ further training examples $\left\{\left\{\mathbf{C}\mathbf{Q}_{\mathbf{P}_2}^T, \mathbf{p}_{gt,2}\right\}, \ldots, \left\{\mathbf{C}\mathbf{Q}_{\mathbf{P}_{n!}}^T, \mathbf{p}_{gt,n!}\right\}\right\}$, as outlined in Figure 4.5. However, rather than reapplying the above-outlined brute-force search strategy to identify the groundtruth classification results $\{\mathbf{p}_{gt,2}, \ldots, \mathbf{p}_{gt,n!}\}$, each one of them is directly and consistently derived from the original groundtruth classification result $\mathbf{p}_{gt}$. This works as follows. This work take $\mathbf{p}_{gt}$ and extract a sequence of permutation matrices ordered in decreasing quality, denoted $\mathcal{P} = \{\mathbf{P}_1^*, \ldots, \mathbf{P}_{n!}^*\}$. For the ith augmented training example generated by the permutation matrix $\mathbf{P}_i$, then extract the consistent ordered sequence $\mathcal{P}_i = \{\mathbf{P}_1^*\mathbf{P}_i^T, \ldots, \mathbf{P}_{n!}^*\mathbf{P}_i^T\}$. The groundtruth classification result $\mathbf{p}_{gt,i}$ is readily extracted from here.

4.3 RESULTS

This work test the method on two generic dense polynomial solvers and one concrete example from geometric vision.

4.3.1 *Potential improvement on general solvers for dense polynomial systems*

Our first example of a generic dense polynomial problem has three variables and three equations of order three:

$$\mathbf{m} = [x_1^3, x_1^2x_2, x_1x_2^2, x_2^3, x_1^2x_3, x_1x_2x_3, x_2^2x_3, x_1x_3^2,$$
$$x_2x_3^2, x_3^3, x_1^2, x_1x_2, x_2^2, x_1x_3, x_2x_3, x_3^2, x_1, x_2, x_3, 1]^{\mathrm{T}}$$

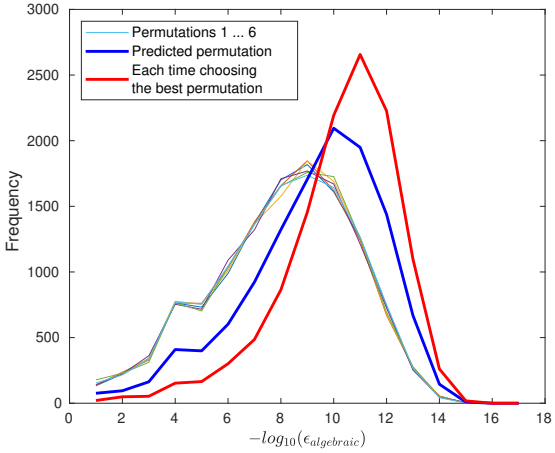
The improvement in solver stability resulting from the utilization of a permutation classifier is indicated in Figure 4.6a. Our second problem has four variables and four equations of maximum degree 2:

$$\mathbf{m} = [x_1^2, x_1x_2, x_2^2, x_1x_3, x_2x_3, x_3^2, x_1x_4, x_2x_4, x_3x_4,$$
$$x_4^2, x_1, x_2, x_3, x_4, 1]^{\mathrm{T}}$$

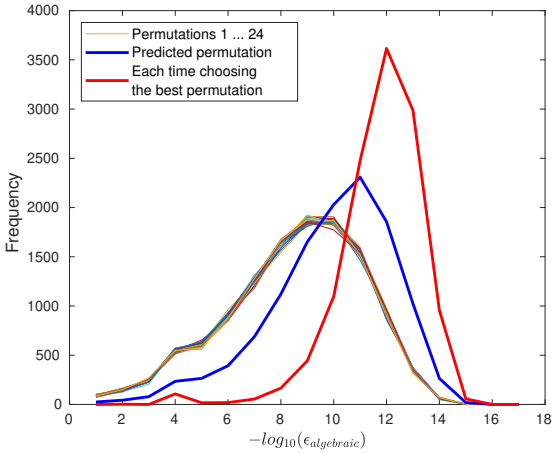
The result is indicated in Figure 4.6b. As can be observed, adding the classifier enables an improvement of the numerical stability of the solver. The coefficients of $\mathbf{C}$ are chosen randomly from the range [0,1] and [0,10]. Note that there is no fixed pattern in $\mathbf{C}$, the range and the coefficients are chosen fully randomly for each new experiment. Examples with no real solutions are ignored. For examples with multiple solutions, this work simply take the average of the sum of absolute residuals of each solution as a ranking or evaluation error. For each problem, this work generate $100000 \times n!$ samples, where $76000 \times n!$ are used for training, $12000 \times n!$ for validation, and $12000 \times n!$ for testing. Each case is trained for 200 epochs. In terms of timing, $n!$ evaluations would need 2.44ms and 5.43ms for these two problems respectively. Running the solver once with prior neural network based prediction of a permutation takes only 0.63ms and 0.89ms, respectively. The potential speed-up factor becomes larger as the number of variables is increasing.

4.3.2 *What works and what not?*

The order of the polynomials as well as the number of variables needs to be sufficiently high in order to see substantial differences between the error distributions for each individual permutation or each time choosing the best one. For example, Figure 4.7a shows the distributions for a quadratic problem in 4 unknowns, indicating that the application of a classifier would not lead to any substantial benefits. What furthermore matters is the presence of sufficient structure in the problem. Our first test consisted of simply choosing the same order of magnitude for all random coefficients, which did not lead to any successful outcomes. It can be concluded that numerical stability only becomes predictable if the coefficients cover a sufficiently large spectrum of possible values. To illustrate this further, here modified both cases outlined in the previous section to include coefficients from three different ranges, i.e. [0,1], [0,10], and [0,100]. For the four-variable problem, furthermore ran one further experiment where include yet another range into the random coefficient generation, i.e. [0,1000]. The results are

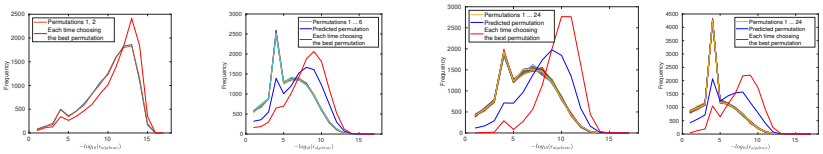


(a)



(b)

FIGURE 4.6: Error distribution over many random instances of the 3 variable/max-degree 3 (a) and the 4 variable/max-degree 2 (b) problems. Coefficients are randomly chosen from the range [0,1] or [0,10]. Each curve represents the distribution obtained by one of the permutations. The red curve indicates the obtainable result if—for each instance—the best permutation is chosen. The blue curve is the distribution obtained by using the permutation predicted by the classifier.



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

FIGURE 4.7: Error distributions for further cases: order 4 in 2 variables (a), order 3 in 3 variables (b), and order 2 in 4 variables (c) and (d). For (b) and (c), the coefficients are randomly drawn from intervals that are either [0,1], [0,10] or [0,100]. For (d) also include [0,1000] as a further possible range. Each curve represents the distribution obtained by one of the permutations. Red curves indicate the obtainable result if—for each instance—the best permutation is chosen. Blue curves are the distributions obtained by using the permutation predicted by the classifier.

indicated in Figured 4.7b, 4.7c, and 4.7d, respectively. As can be observed, the gap between the distribution over individual permutations and the best permutation is increasing along with the possible order of magnitude for the coefficients, and the permutation chosen by the classifier leads to a more pronounced improvement.

4.3.3 *Improvement of camera resectioning algorithm*

Our final experiment consists of an application to a state-of-the-art camera resectioning algorithm, the UPnP algorithm by Kneip et al. UPnP [41]. The goal of the algorithm consists of using an arbitrary number of correspondences between 2D image point measurements and 3D world point coordinates to calculate the six degree-of-freedom absolute pose of the camera. The algorithm makes the assumption of known intrinsic camera parameters, and furthermore apply it in the central, single camera case. Here choose small but non-minimal numbers of correspondences in each experiment, and do not add any noise in order to properly evaluate numerical accuracy.

The UPnP algorithm first eliminates the translation parameters from the estimation, and aims at solving the first-order optimality conditions given by

$$\mathrm{E} = \tilde{\mathbf{s}}^{\mathrm{T}}\mathbf{C}\tilde{\mathbf{s}}, \tag{4.10}$$

where, $\tilde{\mathbf{s}} = [\mathbf{s}^{\mathrm{T}}\ 1]^{\mathrm{T}}$, and

$$\mathbf{s} = [q_0^2, q_1^2, q_2^2, q_3^2, q_0q_1, q_0q_2, q_0q_3, q_1q_2, q_1q_3, q_2q_3]^{\mathrm{T}}$$

represents a vector of all second-order monomials of the quaternion parameters $\mathbf{q} = [q_0, q_1, q_2, q_3]^T$. Completed by a unit-norm constraint on $\mathbf{q}$, the final polynomial problem is permutation invariant and of order 3 in 4 variables.

The input to our system is a 55-dimensional vector. This work train the classifier by generating a total of $100000 \times 4!$ samples, and use $76000 \times 4!$ samples for training, $12000 \times 4!$ for validation, and $12000 \times 4!$ for testing. This work run a total of 200 epochs. Working on a geometric solver lets us furthermore replace the algebraic residual errors by $\left\|\mathbf{T} - \mathbf{T}_{\mathrm{gt}}\right\|_{\mathbf{Frob}}$, where $\mathbf{T}$ and $\mathbf{T}_{gt}$ represent the calculated and the groundtruth pose. Our result is indicated in Figure 4.8 and demonstrates how the addition of the classifier contributes to a significant increase in the numerical stability and accuracy of the solver. Here deem this result as quite important. First, it shows that geometric problems potentially contain the necessary structure in the coefficients. Second, many solvers from geometric vision parametrize the problem as a function of the rotation, and thus appear in permutation invariant form and may potentially benefit from the addition of a similar classifier.

4.4 DISCUSSION

Our chapter demonstrates two main ideas. First, this work show that for certain types of problems there may be multiple ways to execute an elimination template, and notably with similar computational efficiency but different numerical stability. In particular, this work define permutation invariant polynomial systems as a large class of problems for which such a choice is easily achieved by simple permutations. Second, this work prove that the original coefficients can be used to predict a good permutation, thus enabling a cost-effective improvement of the numerical stability of Gröbner basis solvers. The significance of our contribution is increased by the fact that the class of permutation invariant polynomial problems include many solvers from the field of geometric vision, most notably all solvers that apply a polynomial parametrization of rotation matrices, such as Cayley or quaternion parameters. It is also easy to recognize that the method can be transparently extended to problems that contain permutation-variant polynomials, but remain permutation invariant from the point of the view

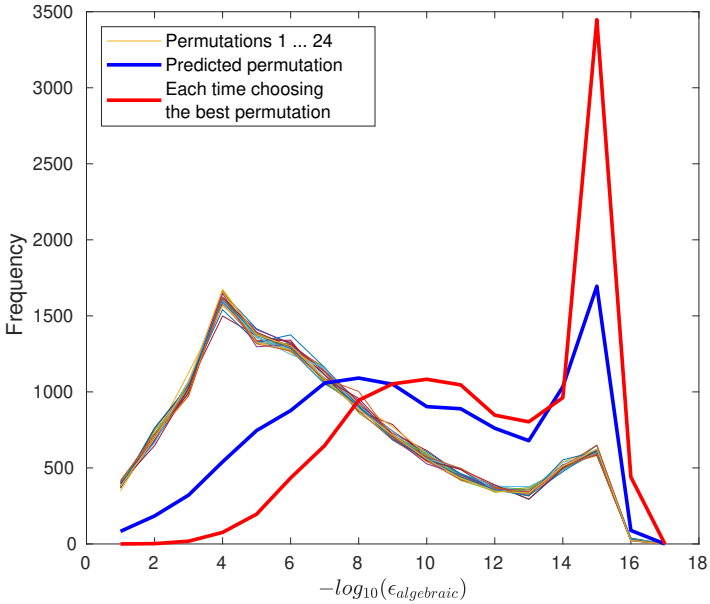


FIGURE 4.8: Error distribution over many random instances of the UPnP problem [41]. Each curve represents the distribution obtained by one of the permutations. The red curve furthermore indicates the obtainable result if—for each instance—the best permutation is chosen. The blue curve is the distribution of the predicted permutation.

of the entire system. A further applicable case is when only a subset of the variables or monomials appears in permutation invariant form. Our future work therefore consists of extending the idea to more general cases including not only such partially permutation invariant problems, but also cases for which simply different, similarly efficient elimination templates exist.

5

EVENT CAMERA'S VELOMETER ON A CAR

Visual Odometry (VO) is a fundamental computer vision problem with important applications in robotics and automotive [180, 181]. Research over the past two decades has therefore lead to significant progress and mature visual odometry frameworks for regular cameras [**murAcceptedTRO2015**, 182–184]. However, while some of these methods specifically target the application on non-holonomic platforms, current methods face challenges posed by scenarios involving high dynamics and high-dynamic-range conditions. In such situations, regular cameras are easily affected by motion blur or over exposure. To address these challenges and pave the way for further progress in the field of VO, the community has recently started to explore the use of dynamic vision sensors [81, 83, 84, 88, 98, 99, 102, 107, 185].

Event cameras are innovative visual sensors that offer distinct advantages in high-speed scenarios. Drawing inspiration from the biological retina, each pixel of an event camera functions independently and only responds when a change in brightness level occurs and exceeds a certain threshold. This fundamental circuit-level design provides event cameras with an improved power dissipation, high dynamic range, and reduced response latency, and therefore holds great potential for promoting advancements in visual odometry. Consequently, event-based visual odometry has garnered increasing attention from academia and industry alike [64]. However, the asynchronous mechanism at the pixel level and the relatively immature design of event cameras still pose significant obstacles to the development of event-based visual odometry systems. More specifically, the resulting noise and spatial sparsity in event streams hinder stable and accurate feature extraction as well as the establishment of correspondences over time. The community has therefore converged onto visual-inertial fusion [95–97] or even event-frame-inertial fusion strategies [98, 102] for stable motion estimation with event cameras.

The present chapter presents a reliable purely event-based visual odometry framework for planar ground vehicles by including a non-holonomic motion model into the estimation. As indicated in Figure 5.1, this work use the Ackermann model, a common representation to approximate the instantaneous planar motion of ground vehicles by circular arcs as a function

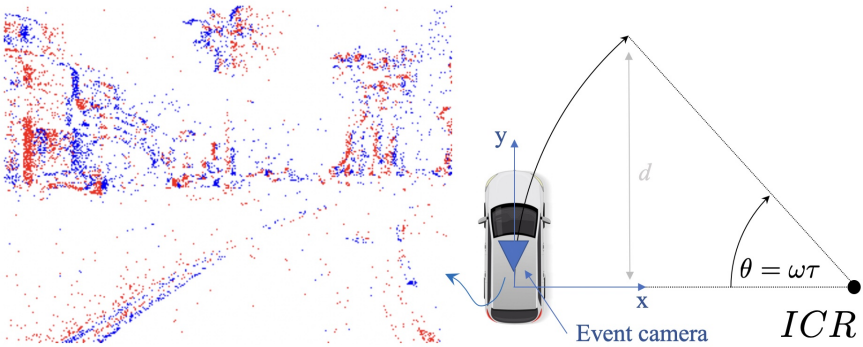


FIGURE 5.1: This work consider visual odometry with a forward-facing event camera mounted on an Ackermann steering vehicle for which motion can be locally approximated by an arc of a circle about an Instantaneous Centre of Rotation (ICR). This work assume constant rotational velocity during the time interval for which events are considered.

of two degrees of freedom [73, 78]: the turning radius and the speed of the motion. Combined with the radius, the forward velocity of the vehicle is equivalently expressed by rotational velocity. Based on the assumption of locally constant rotational velocity and turning radius, this work introduce a *one-track solver* that is able to initialize the observable part of the instantaneous camera motion from a short, temporal trail of events generated by a single 3D point observed under non-holonomic motion. The algorithm is derived from the standard camera-based one-point algorithm for the two-view scenario [73] and its n-linear extension incorporating both points and lines [78]. As will show in this work, the extension to event streams alleviates limitations in high-speed and high-dynamic scenarios, while the introduction of the constrained motion model is sufficient to robustify the geometric estimation. The contributions of our work are summarized as follows:

- This work introduce a continuous-time incidence relation for tracked event corners based on the non-holonomic Ackermann motion model and a locally constant rotational velocity assumption.
- This work employ Taylor expansions to approximate trigonometric functions, resulting in n-linear constraints that depend on higher orders of the rotational velocity. This work analyze three distinct expansion levels.

- This work utilize rank minimization to solve the rotational velocity, and eliminate outliers using histogram voting.

As demonstrated in our experimental results section, the resulting algorithm leads to accuracy comparable to conventional camera alternatives, and eventually outperforms the latter under challenging illumination conditions. The remainder of this chapter is structured as follows. Section 5.1 provides a brief review of the Ackermann motion model, and presents our proposed n-linear constraint as well as our robust estimation strategy over multiple event tracks. Section 5.2 finally presents a thorough evaluation of the algorithm over simulation data, as well as a successful application to challenging real-world cases in which regular camera alternatives fail.

5.1 EVENT-BASED NON-HOLONOMIC SOLVER

Now proceed to the derivation of the solver. This work start with a review of the Ackermann motion model. Next introduce the core contribution, which is a novel algorithm for estimating up-to-scale non-holonomic motion dynamics from a single event trail. More specifically, this work propose a novel incidence relation that vanishes for all events generated by the same 3D point under continuous, constant velocity Ackermann motion. To conclude, this work adopt rank minimization to solve the constraint and histogram voting over multiple trails to remove outliers.

5.1.1 *The Ackermann Motion Model*

As introduced in the original work of Scaramuzza et al. [73, 75], the motion of a planar Ackermann steering vehicle can be approximated to lie on a circular arc contained in the horizontal plane. The heading of the vehicle furthermore stays tangential to the arc. This work adopt the parametrization of Huang et al. [78], where the body frame is defined such that the x-axis points rightward, the y-axis forward, and the z-axis upward. The x-axis is defined to lie at the height of the non-steering back-wheel axis, which is why the Instantaneous Centre of Rotation (ICR) intersects with the x-axis. As a result, the instantaneous velocity of the vehicle points along the y-axis. A minimal parametrization of a relative displacement is now given by the circle radius r and the inscribed arc-angle θ. The geometry is explained in Figure 5.1. Note that this model is valid under three assumptions: The motion is slip-free, the back-wheel axis is non-steering, and the motion has constant velocity. While the latter assumption does not hold in practice, this

work demonstrate that the assumption holds sufficiently well over the short time intervals over which the estimation happens.

Let us now define the Euclidean transformation between subsequent frames as a function of our minimal parametrization. As explained in [78], the relative transformation variables $\mathbf{R}$ and $\mathbf{t}$ that allow us to transform points from the later frame back to the initial frame according to the equation $\mathbf{p}_0 = \mathbf{R}\mathbf{p}_1 + \mathbf{t}$ are given by

$$\mathbf{R} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1' \end{bmatrix}, \; \mathbf{t} = \frac{d}{\sin(\theta)} \begin{bmatrix} 1-\cos(\theta) \\ \sin(\theta) \\ 0 \end{bmatrix},$$

where we have replaced $r = \frac{d}{\sin(\theta)}$ to define scale via the forward displacement d along the y axis and avoid numerical issues if r tends to infinity and θ tends to zero. As can easily be verified by the application of L'Hôpital's rule, we now obtain $\mathbf{t} = [0 \; d \; 0]^\top$ as $\theta \to 0$. Note that the convention here is that both r and θ are positive for a forward right turn (cf. situation illustrated in Figure 5.1), and both r and θ are negative for a forward left turn.

5.1.2 *Single Event Trail Constraint*

Let us assume that the constant velocity assumption is satisfied for a short period of time. For each event $\mathbf{e}_i = \{u_i, v_i, t_i, s_i\}$ belonging to a short temporal slice of the space-time volume of events, let (u_i, v_i) define the pixel location of the event on the image plane, t_i its timestamp, and s_i its polarity. Using the continuous-time representation $\theta = \omega\tau$—where ω represents the rotational velocity of the camera—the relative $\mathbf{R}_i$ and $\mathbf{t}_i$ corresponding to the exact time of event $\mathbf{e}_i$ is given by

$$\mathbf{R}_i = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) & 0 \\ -\sin(\theta_i) & \cos(\theta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\omega\tau_i) & \sin(\omega\tau_i) & 0 \\ -\sin(\omega\tau_i) & \cos(\omega\tau_i) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{5.1}$$

$$\mathbf{t}_i = \frac{d}{\sin(\theta)} \begin{bmatrix} 1-\cos(\theta_i) \\ \sin(\theta_i) \\ 0 \end{bmatrix} = \frac{d}{\sin(\omega\tau)} \begin{bmatrix} 1-\cos(\omega\tau_i) \\ \sin(\omega\tau_i) \\ 0 \end{bmatrix}. \tag{5.2}$$

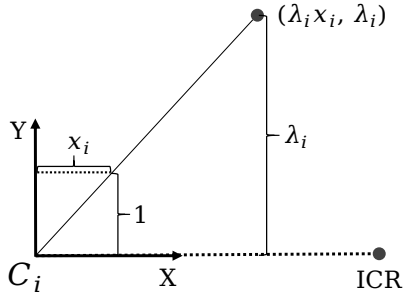


FIGURE 5.2: The model used in the derivation, where the coordinate system C_i is consistent with the event $\mathbf{e}_i$. Please see text for detailed explanations.

Note that τ here represents a fixed time interval used to again fix the scale as the locally constant turning radius is replaced by $\frac{d}{sin(\omega\tau)}$. By again applying L'Hôpital's rule, we obtain $\mathbf{t}_i = \begin{bmatrix} 0 & \frac{d\tau_i}{\tau} & 0 \end{bmatrix}^\top$ as $\omega \to 0$, which is as required. Note furthermore that $\mathbf{p}_0 = \mathbf{R}_i\mathbf{p}_i + \mathbf{t}_i$, where $\mathbf{p}_i$ expresses the world point corresponding to $\mathbf{p}_0$ in the displaced frame at time t_i. τ_i represents the time relative to the start time t_0 of the considered time interval. That is, for event $\mathbf{e}_i$ with timestamp t_i, we have $\tau_i = t_i - t_0$.

We have $\mathbf{p}_i = \mathbf{R}_i^\top(\mathbf{p}_0 - \mathbf{t}_i)$, and using $\mathbf{p}_i = [p_i^x, p_i^y, p_i^z]^\top$, and by replacing $\mathbf{R}_i$, $\mathbf{t}_i$ with Eqs. (5.1), (5.2), respectively, we obtain

$$\begin{aligned}
\begin{bmatrix} p_i^x \\ p_i^y \\ p_i^z \end{bmatrix} &= \begin{bmatrix} \cos(\omega\tau_i) & -\sin(\omega\tau_i) & 0 \\ \sin(\omega\tau_i) & \cos(\omega\tau_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} p_0^x \\ p_0^y \\ p_0^z \end{bmatrix} - \frac{d}{\sin(\omega\tau)} \begin{bmatrix} 1-\cos(\omega\tau_i) \\ \sin(\omega\tau_i) \\ 0 \end{bmatrix} \right) \\
&= \begin{bmatrix} p_0^x\cos(\omega\tau_i) - p_0^y\sin(\omega\tau_i) - \frac{d}{\sin(\omega\tau)}(\cos(\omega\tau_i) - 1) \\ p_0^x\sin(\omega\tau_i) + p_0^y\cos(\omega\tau_i) - \frac{d}{\sin(\omega\tau)}\sin(\omega\tau_i) \\ p_0^z \end{bmatrix}.
\end{aligned}$$

Given that this constitutes motion on a ground plane, the z coordinate remains unchanged and we may simply ignore the third equation. Furthermore, for the sake of simplicity, we define the camera frame as being identical with the vehicle frame. Hence, we may directly express $(p_i^x, p_i^y) = (\lambda_i x_i, \lambda_i), i = 1, ..., n$ (illustrated in Fig. 5.2), i. e. express the world points in the camera frame as normalized image points multiplied

by their depth along the principal axis (in this case the y axis). Note that the entire problem is formulated by projection onto the horizontal plane, and only horizontal bearing measurements are used rather than complete image point measurements (i. e. the coordinate along the row dimension is dropped, while x_i is used). The preceding equation can thus be rewritten as

$$\lambda_i \begin{bmatrix} x_i \\ 1 \end{bmatrix} = \begin{bmatrix} p_0^x \cos(\omega\tau_i) - p_0^y \sin(\omega\tau_i) - \frac{d}{\sin(\omega\tau)}(\cos(\omega\tau_i) - 1) \\ p_0^x \sin(\omega\tau_i) + p_0^y \cos(\omega\tau_i) - \frac{d}{\sin(\omega\tau)} \sin(\omega\tau_i) \end{bmatrix}.$$

The λ_i can be effortlessly eliminated by dividing the first row by the second row, resulting in

$$x_i = \frac{p_0^x \cos(\omega\tau_i) - p_0^y \sin(\omega\tau_i) - \frac{d}{\sin(\omega\tau)}(\cos(\omega\tau_i) - 1)}{p_0^x \sin(\omega\tau_i) + p_0^y \cos(\omega\tau_i) - \frac{d}{\sin(\omega\tau)} \sin(\omega\tau_i)}. \tag{5.3}$$

Following simple derivations, Eq. (5.3) can be reformulated into the simple matrix form

$$\begin{bmatrix} a_{i1} & a_{i2} & a_{i3} \end{bmatrix} \begin{bmatrix} p_0^x \\ p_0^y \\ d \end{bmatrix} = 0, \tag{5.4}$$

where

$$\begin{aligned} a_{i1} &= -x_i \sin(\omega\tau_i) + \cos(\omega\tau_i), \\ a_{i2} &= -x_i \cos(\omega\tau_i) - \sin(\omega\tau_i), \\ a_{i3} &= \frac{x_i \sin(\omega\tau_i) - \cos(\omega\tau_i) + 1}{\sin(\omega\tau)}. \end{aligned} \tag{5.5}$$

Equ. (5.4) represents the constraint between the associated events and motion parameters, and—by stacking multiple measurements from multiple events—it is what we call an *n-linearity*. However, the solution is challenging as the left-hand matrix remains a highly non-linear, trigonometric function of ω.

5.1.3 *Transformation Into a Polynomial Constraint*

In the following, this work adopt a sequence of transformations and approximations in order to transform the n-linearity constraint from Eq. (5.4) into a simple, uni-variate polynomial. Required by the validity of the constant velocity assumption, τ_i has to remain small. As a result, $\omega\tau_i$ also remains

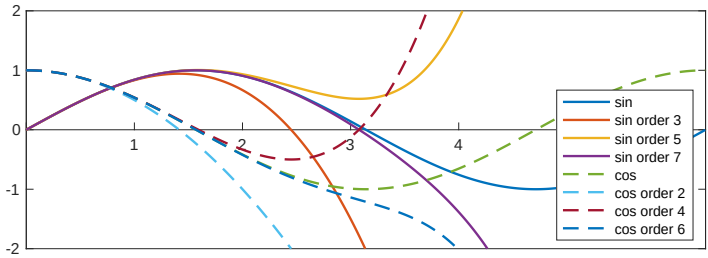


FIGURE 5.3: The Taylor series expansion approximations for sin and cos, the corresponding highest orders are 3, 5, 7 and 2, 4, 6 respectively.

small. In order to facilitate the subsequent polynomial functions of $\theta = \omega\tau$, this work employ Taylor series expansions to approximate trigonometric functions as in

$$\begin{aligned} \sin(\theta) &\approx \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} + \cdots + \frac{(-1)^n\theta^{2n+1}}{(2n+1)!} + \cdots, \\ \cos(\theta) &\approx 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} + \cdots + \frac{(-1)^n\theta^{2n}}{(2n)!} + \cdots. \end{aligned} \tag{5.6}$$

The Taylor series approximations for sin and cos presented in Fig. 5.3 correspond to the cut-off orders of 3, 5, 7 and 2, 4, 6, respectively.

Let us denote the *s3c2* algorithm as the one that sets the highest order of the sin approximation to 3 and the one of the cos approximation to 2. The *s5c4* and *s7c6* algorithms are defined in a similar way.

Upon substituting Eq. (5.6) into Eq. (5.5), with obtain the following form for *s3c2*

$$\begin{aligned} a_{i1} \approx \widetilde{a}_{i1} &= x_i\left(\frac{(\omega\tau_i)^3}{6} - \omega\tau_i\right) - \frac{(\omega\tau_i)^2}{2} + 1, \\ a_{i2} \approx \widetilde{a}_{i2} &= x_i\left(\frac{(\omega\tau_i)^2}{2} - 1\right) + \frac{(\omega\tau_i)^3}{6} - \omega\tau_i, \\ a_{i3} \approx \widetilde{a}_{i3} &= \frac{-(\tau_i(-x_i\tau_i^2\omega^2 + 3\tau_i\omega + 6x_i))}{\tau(\tau^2\omega^2 - 6)}. \end{aligned} \tag{5.7}$$

For *s5c4* and *s7c6*, due to space constraints, this work have placed the results in appendix A.1. The denominator in a_{i3} can be eliminated by multiplication of both the left and right-hand sides of Eq. (5.4). We obtain

$$\begin{bmatrix} b_{i1} & b_{i2} & b_{i3} \end{bmatrix} \begin{bmatrix} p_0^x \\ p_0^y \\ d \end{bmatrix} \approx 0, \tag{5.8}$$

where

$$b_{ij} = c\widetilde{a}_{ij},\ j = 1, 2, 3, \tag{5.9}$$

and for *s3c2*, *s5c4*, and *s7c6*, the value of c is given by

$$\begin{aligned} s3c2:\ c &= \tau(\tau^2\omega^2 - 6), \\ s5c4:\ c &= \tau(\tau^4\omega^4 - 20\tau^2\omega^2 + 120), \\ s7c6:\ c &= \tau(\tau^6\omega^6 - 42\tau^4\omega^4 + 840\tau^2\omega^2 - 5040). \end{aligned} \tag{5.10}$$

Given n corresponding events, the constraints related to each event can again be assembled into an n-linear problem, resulting in the formulation

$$\begin{aligned} &\begin{bmatrix} b_{01} & b_{02} & b_{03} \\ \vdots & \vdots & \vdots \\ b_{i1} & b_{i2} & b_{i3} \\ \vdots & \vdots & \vdots \\ b_{(n-1)1} & b_{(n-1)2} & b_{(n-1)3} \end{bmatrix} \begin{bmatrix} p_0^x \\ p_0^y \\ d \end{bmatrix} \\ &= \mathbf{B}[k](\omega) \begin{bmatrix} p_0^x \\ p_0^y \\ d \end{bmatrix} \approx \mathbf{0}, \end{aligned} \tag{5.11}$$

where $\mathbf{B}[k](\omega)$ represents a degree-k matrix in ω, and $\mathbf{B}$ refers to a matrix of dimensions $n \times 3$.

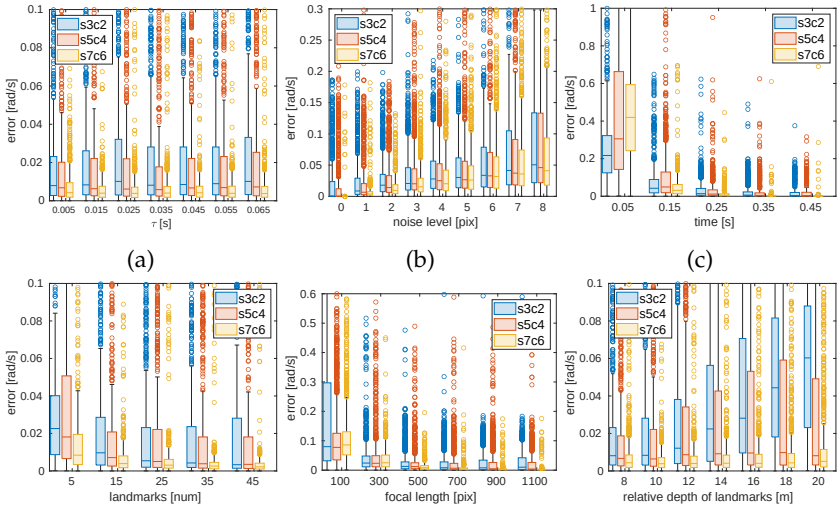


FIGURE 5.4: Impact of different factors on the accuracy of the recovered rotational velocity. sacb means that the highest order of the Taylor series expansions for sin is a and the highest order of the Taylor series expansion for cos is b.

5.1.4 *From Rank Minimisation to a Univariate Polynomial Objective*

To obtain a non-trivial solution for ω in objective (5.11), it is essential for $\mathbf{B}$ to display rank deficiency. Therefore, the solution for ω can be facilitated by addressing the rank minimization problem

$$\omega_{\text{opt}} = \underset{\omega}{\text{argmin}}\ \text{rank}(\mathbf{B}[k](\omega)). \tag{5.12}$$

Given that $\text{rank}(\mathbf{B}) = \text{rank}(\mathbf{B}^{\text{T}}\mathbf{B})$, the optimisation goal finally becomes

$$\omega_{\text{opt}} = \underset{\omega}{\text{argmin}}\ \text{rank}(\mathbf{M}[2k](\omega)), \tag{5.13}$$

where $\mathbf{M}[2k](\omega) = (\mathbf{B}[k](\omega))^{\text{T}}(\mathbf{B}[k](\omega))$ defines a 3×3 polynomial matrix function of ω. As a positive semi-definite matrix, $\mathbf{M}$'s rank can be minimized by minimising its smallest eigenvalue. The objective becomes

$$\omega_{\text{opt}} = \underset{\omega}{\text{argmin}}\ \min(\underset{\lambda}{\text{solve}}(\det(\mathbf{M} - \lambda\mathbf{I}))), \tag{5.14}$$

where λ represents the smallest eigenvalue, and $\mathbf{I}$ is a 3×3 identity matrix. Despite its compact appearance, this objective poses significant optimisation challenges due to its reliance on a repetitive, internal determination of $\mathbf{M}$'s smallest eigenvalue, which is computationally difficult to achieve in closed form. However, it becomes evident that in the perfect noise-free scenario, the rank deficiency condition is met when the smallest eigenvalue of $\mathbf{M}$ at the optimal point simply reduces to zero. Consequently, this work approximate $\lambda_{\text{min}} = 0$ and proceed to resolve the final objective

$$\omega_{\text{opt}} = \underset{\omega}{\text{argmin}}(\det(\mathbf{M})). \tag{5.15}$$

Observe that this corresponds to identifying the real roots of a univariate polynomial in ω, resolved via Sturm's root bracketing methodology. The determinant polynomials for *s3c2*, *s5c4*, and *s7c6* exhibit an order of 30, 54, and 78, respectively. In practice however, the number of real roots is typically much lower, and much fewer solutions will have to be disambiguated to find the best one.

5.2 EXPERIMENTS

In the following, this work will present results for the algorithm collected both in simulation and on real data. In the real-world case, this work test on

image sequences for both day and night conditions. This work employ the histogram voting technique outlined in [78] for handling outlier removal and refining the solution.

5.2.1 *Experiments on Synthetic Data*

This work conduct a variety of experiments with different impact factors, allowing only one parameter to vary within a certain range per experiment. Generally, for the invariant parameters, this work configure the noise of each event with a standard deviation of 1 pixel, the time interval length to 0.3 seconds, and the number of observed landmarks to 15. The landmarks have a randomly distributed relative depth between [2, 18] units, with an average value of 10. Additionally, the standard focal length used for the experiments was 700 pixels. In each experiment and setting, this work conducted 1000 random experiments. The error ϵ between the recovered angular velocity ω_{rec} and the ground truth angular velocity ω_{gt} is calculated using

$$\epsilon = |\omega_{rec} - \omega_{gt}|. \tag{5.16}$$

Each experiment evaluates the impact of different highest orders of our Taylor series expansions, including *s3c2*, *s5c4*, and *s7c6*. To recap, in this notation "*s3*" means that the highest order of the Taylor series approximation for sin is 3 *etc.*. It is worth noting that attempting to use higher orders of Taylor series expansion beyond *s7c6* would lead to slower calculations despite no further improvements in accuracy. Hence, higher orders are not incorporated into the analysis.

Following is an analysis of each influencing factor:

- τ. τ is merely a constant to define the traversed angle within a certain period of time. Together with the sine of that angle and a fixed forward displacement, the constant defines the turning radius. In short, it only impacts on the scale of the problem, and has no implications on the accuracy of the recovered turning rate, which is consistent with the results of Fig. 5.4a.

- *Event noise level*. This work conducted performance tests of the algorithm under various noise levels. As shown in Fig. 5.4b, this work observed that up to noise levels of 5 pixels, the error decreases with an increase in the highest order of the Taylor series expansion. Consequently, *s7c6* achieves the smallest error among the tested options.

However, interestingly, at noise levels 6, 7, and 8 pixels, *s5c4* outperforms other configurations. One possible explanation for this phenomenon is that *s7c6*, being of higher order, becomes more sensitive to noise, which adversely affects its performance in high noise conditions. In contrast, *s5c4* strikes a better balance between accuracy and noise sensitivity in those cases.

- *Variation of time interval*. Vary the time interval from 0.05s to 0.45s. As demonstrated in Fig. 5.4c, the error decreases as the length of the time intervals goes up. This observation aligns with the analysis presented in [107, 185]. Nevertheless, it is essential to note that for time intervals exceeding 0.35s, the impact on accuracy is relatively limited. In other words, increasing the time interval beyond this threshold has no significant impact on the algorithm's accuracy.

- *Variation of the number of landmarks*. This parameter indicates the number of landmarks that can be detected from the scene. As seen can be seen in Fig. 5.4d, a higher number of landmarks tends to contribute to more accurate results. However, there is a saturation point where a further increase in the number of landmarks has no more impact on accuracy. Furthermore, the superiority of *s7c6* and *s5c4* over *s3c2* implies that higher orders have a positive effect in reducing the error.

- *Focal length*. As depicted in Fig. 5.4e, the error diminishes as the focal length increases. Notably, within the focal length range of 100 to 500, there is a substantial reduction in the error. However, for focal lengths between 500 and 1100, the impact on accuracy becomes less pronounced. This observation holds valuable insights for practical camera applications.

- *Relative depth of landmarks*. For this parameter, the horizontal coordinate represents the mean value of the relative depth distribution, which is uniformly distributed within 8m of that value. In Fig. 5.4f, the errors of *s3c2* and *s5c4* exhibit an increase as the distance increases, and it is particularly pronounced for *s3c2*. However, in the case of *s7c6*, the error remains relatively stable even as the distance increases.

Based on the above results, although *s7c6* may exhibit slightly weaker performance than *s5c4* under very high levels of noise, in practical scenarios such extreme noise levels are not commonly encountered, and that *s7c6* demonstrates better performance in other evaluation metrics. Consequently,

for the subsequent real-data experiment, this work adopt *s7c6* as a fixed configuration.

5.2.2 *Experiments on Real-World Data*

To evaluate our algorithm, this work conducted tests on KITTI [186] as well as on our own Self-Collected Data (SCD). The results are compared against the image-based algorithm **1FPN** by Huang et al. [78]. Note that all qualitative trajectory results are generated by taking additional ground truth scale information into account given that monocular odometry is scale-invariant.

For KITTI, this work test on sequences 0046, 0095, and 0104, respectively. As presently event cameras have relatively low resolution, here cropped the raw KITTI images (Fig. 5.5a) to a resolution of 608×375 (Fig. 5.5b). This step was taken to achieve a better approximation of real-world conditions and enhance computational efficiency. After cropping the image, retained only the middle part of the raw image, the principal point and focal length are changed accordingly.

KITTI is lacking night images. In order to compare our algorithm with **1FPN** during both day (D) and night (N) conditions, this work first apply a dark filter to the cropped images to approximate nighttime images. The effect of the dark filter can be observed in Fig. 5.5c. It is important to note that—while they may appear differently from real nighttime images—the dark-filtered night images yield similar (or even better) feature extraction results than real night images, hence no disadvantage is given to traditional image-based methods such as **1FPN**.

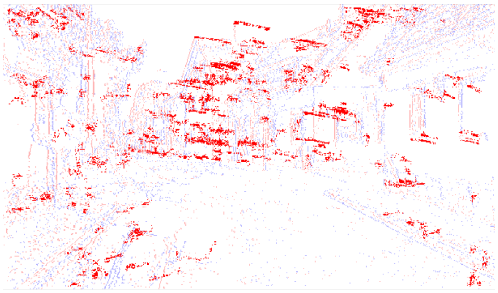
The event-based sequences generated on KITTI by application of vid2e [187] are generated using dark-filtered night images, only. An example result is indicated in Fig. 5.5d. For event-based feature tracking, this work use the work of Alzugaray et al. [188] to detected corner events and return continuous trajectories in the space-time volume. Trajectories are then evaluated for noise based on node count and duration, resulting in feature trajectories lasting between 0.15 to 0.25 seconds.

SCD was collected at night using a Prophesee Gen3.1 CD event camera, a FLIR Grasshopper3 traditional camera, and a Ouster OS0-128 LIDAR. The LIDAR here is added to obtain ground truth trajectories by running LeGO-LOAM [189]. As shown in Fig. 5.5e), SCD contains both events and real night time images.

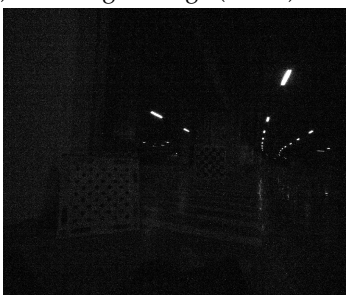
(a) Raw Image (KITTI)

(b) Cropped Image (KITTI)

(c) Dark Night Image (KITTI)



(d) Events and Corner Tracking (KITTI)



(e) Raw Image (SCD)

FIGURE 5.5: Example images and events with tracked corners used in our real-data experiments. (a), (b), and (c) represent a raw image, a cropped image, and a dark-filtered night image for KITTI, respectively. (d) shows the events generated using vid2e and demonstrates the corner tracking result. (e) is an example image from our self-collected data at nighttime, on which again almost no features can be extracted.

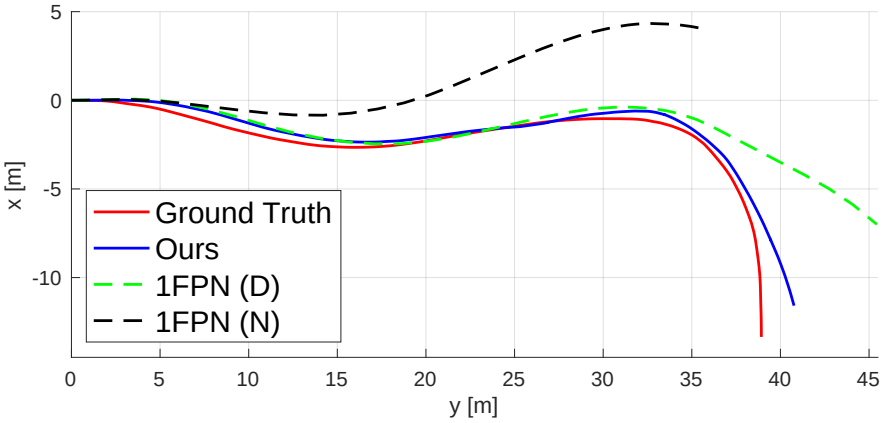


FIGURE 5.6: Comparison of our event-based method against an image-based method [78] on the KITTI 0046 sequence, using both the cropped and the dark-filtered night images (D=Day, N=Night).

Fig. 5.6 shows a qualitative trajectory result obtained for KITTI 0046. As can be seen, our results are closest to ground truth and even outperform the traditional image-based alternative applied at day-time (**1FPN** (D)). The latter algorithm applied at night time (**1FPN** (N)) exhibits strong drift right from the beginning and fails to complete the trajectory, thus indicating the lack of sufficient accurate features in low-light conditions. Fig. 5.7 displays a result obtained on our night sequences SCD. As can be observed, our method works well while the lack of features in the corresponding night images (cf. Fig. 5.5e) causes **1FPN** to fail.

In terms of quantitative comparison, this work calculate the errors ϵ and ϕ for the relative rotation angles and translation directions, respectively. The errors are evaluated using both the root mean square (RMS) μ and median ν. The errors obtained on the different sequences are summarized in Tab. 6.1, and a runtime efficiency comparison is presented in Tab. 5.2. As can be observed from Tab. 6.1, our method almost consistently achieves the lowest error, and significantly outperforms **1FPN** under dark conditions. As for the time comparison in Tab. 5.2, due to the continuous nature of the event camera, our algorithm reaches higher frequent outputs and achieves real-time performance.

Seq.	Method	Error			
		$\mu(\epsilon)$	$\nu(\epsilon)$	$\mu(\phi)$	$\nu(\phi)$
		[deg]	[deg]	[deg]	[deg]
0046	**1FPN** (D)	2.7197	1.6301	2.8090	1.2605
	1FPN (N)	1.8686	1.3022	8.1501	8.7737
	Ours	**0.9599**	**0.2154**	**1.5607**	**1.2110**
0095	**1FPN** (D)	0.6467	0.5450	1.7993	1.8080
	1FPN (N)	-	-	-	-
	Ours	**0.6187**	**0.4024**	**1.4001**	**0.9345**
0104	**1FPN** (D)	0.3732	0.2684	**0.4679**	0.4254
	1FPN (N)	0.5450	0.3841	0.8870	0.7334
	Ours	**0.0515**	**0.0357**	0.5686	**0.3608**
SCD	**1FPN** (N)	-	-	-	-
	Ours	**0.2786**	**0.1080**	**2.5317**	**2.0487**

TABLE 5.1: Error comparison on different sequences (D=Day, N=Night). Note that for SCD, there is only **1FPN** (N).

Seq.	Method	Time		
		Avg.	Num.	Rem.
		[ms]	[1]	-
0046	**1FPN** (D)	136.3835	41	whole
	1FPN (N)	78.0778	33	part
	Ours	**44.7761**	856	whole
0095	**1FPN** (D)	120.4234	89	whole
	1FPN (N)	-	-	failed
	Ours	**36.0974**	175	part
0104	**1FPN** (D)	118.2199	103	whole
	1FPN (N)	80.2444	103	whole
	Ours	**16.7670**	1381	whole
SCD	**1FPN** (N)	-	-	failed
	Ours	47.1908	833	whole

TABLE 5.2: Time comparison on different sequences (D=Day, N=Night). Note that the SCD is collected at night, there is only **1FPN** (N). "Avg." denotes average calculation time, "Num." is the count of frequent outputs, and "Rem." remarks completion status over the sequence.

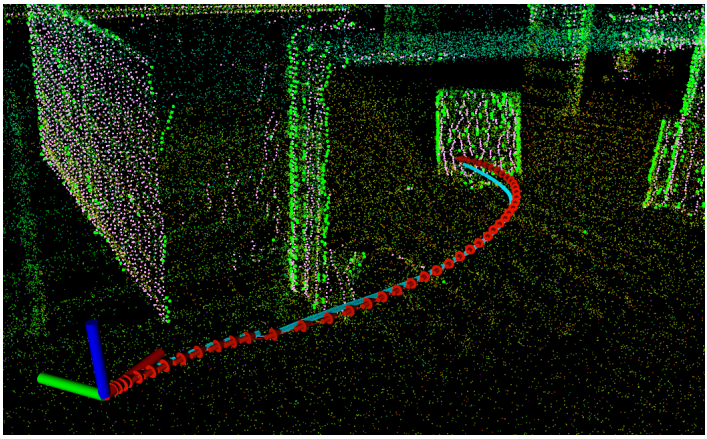


FIGURE 5.7: Results of our event-based method (in blue) on SCD with ground truth (in red). The scene was build by LeGO-LOAM. Note that this sequence was collected at night, and the image-based method by Huang et al. [78] does not work.

5.3 CONCLUSIONS

This work have introduced a continuous-time adaptation of a frame-based, constant velocity n-view relative displacement solver for Ackermann motion, thereby permitting a robust and accurate solution to monocular event-based visual odometry on a ground vehicle platform. The method clearly outperforms comparable frame-based solutions in difficult, low illumination conditions, and thereby alleviates one of the major disadvantages of traditional vision-based solution for self-driving cars.

6

EVENT-INERTIAL VELOMETER

Over the past two decades, monocular visual-inertial ego-motion estimation has turned from a scientific challenge into a mature, indispensable solution in restricted energy, payload, and budget applications such as low-end consumer service robotics, Unmanned Aerial Vehicles (UAVs), or intelligence augmentation devices. While popular open-source solutions mostly rely on sparse feature extraction (e.g. ORB-SLAM [56], VINS-Mono [116], OKVIS [117]), the community has also explored alternative feature-based methods (e.g. PL-VIO [131], PL-SLAM [130]), dense optical flow-based approaches (e.g. VOLDOR+SLAM [61]), or direct photometric error minimizers (e.g. DM-VIO [118]).

In the following, the discussion distinguishes between absolute world-centric and relative camera-centric parameters. The former are given by the absolute sensor pose as well as the absolute landmark coordinates in the world (i.e. in a world reference frame), while the latter are given by the dynamic ego-state (i.e. translational and rotational velocity and acceleration expressed in the camera frame) as well as the relative depth of the landmarks with respect to the current frame. The above mentioned SLAM frameworks all employ absolute world-centric representations, and the kinematics of the sensor are estimated as implicitly estimated sub-states rather than directly measured. This enables map construction and memorization as well as global localization and path planning. The down-side of such representations, however, is that the stability of the dynamic states depends on the stability of the overall state estimation, which includes the aforementioned absolute world-centric parameters. Events such as map tracking failures or loop closures may easily impact on the quality of the dynamics estimation, and hence prevent the reliable solution of safety-critical velocity-based control problems (e.g. UAV stabilization, obstacle avoidance). This motivates fail-safe approaches in which continuous epipolar geometry and inertial cues are coupled to directly estimate the dynamics of a platform [123].

This chapter present a novel solution to visual-inertial fusion for direct dynamics estimation by employing a dynamic vision sensor. The latter—also called an event camera—is a bio-inspired, low-latency sensor that operates fundamentally differently from a regular camera. Rather than measuring

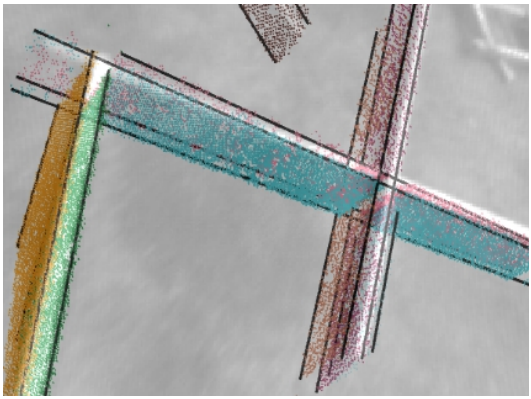


FIGURE 6.1: Example of a local event stream pattern used for camera velocity estimation. Each cluster of events corresponds to a locally observed, real-world line segment.

intensity images frame-by-frame, an event camera employs independent pixel-level CMOS circuits measuring the change of brightness patterns. Each individual pixel asynchronously fires time-stamped events whenever the logarithmic intensity change since the last event exceeds a predefined threshold. Event cameras have latencies in the order of micro-seconds, may fire events at a very high rate, and own high dynamic range. These properties make the event camera an excellent choice in challenging visual conditions caused by high dynamics or low illumination [97, 99, 100].

This work consider event cameras to be a highly intuitive choice for tight visual-inertial dynamics estimation, as events are a direct consequence of velocities and relative feature depths. Our solution relies on our previous result, a novel trifocal tensor-based [67] incidence relation that can be solved in closed-form and directly relates events triggered by moving appearance edges, the corresponding lines, and camera velocities [107]. Robustness is achieved through a novel two-layer RANSAC scheme, and a regularized, sliding-window optimizer ensures smooth velocity estimation over time. As demonstrated by our results on both simulated and real data, our approach generates velocity samples from temporal intervals of events that are comparable to the exposure time of regular images. Event camera-based visual-inertial fusion for direct velocity sensing achieves highly reliable results, and—owing to the absence of motion blur and a positive correlation between SnR and motion dynamics—appears as a highly sensible, fail-safe

sensor-fusion strategy able to cope with challenging high-velocity UAV motion.

The main contributions are listed as follows:

- To the best of our knowledge, the first comprehensive work on turning the highly bio-inspired setup of an event camera and an inertial sensor into a direct line-based visual-inertial 3D speed sensor.
- This work pioneer the use of RANSAC with events, and propose a novel nested two-layer RANSAC scheme for geometric, event-based velocity initialization from moving line observations. The outer layer is a minimal application within RANSAC of the trifocal tensor constraint inspired by our previous work [107], and a novel 4-event solver for 3D line reconstruction is proposed in the inner layer.
- This work further introduce a complete inertial event-based solution including a bootstrapper and a purely relative sliding window back-end optimizer. The regularization is formulated using manifold-based pre-integration of inertial readings.

The chapter is organized as follows. Section 6.1 gives a brief review of the continuous event-line constraint and the line representation used in this work. The theory then is divided into two parts—initialization through closed-form velocity calculation in Section 6.2, and tightly-coupled back-end optimization in Section 6.3. Experiments are in Sections 6.4 and 6.4.2.

6.1 PRELIMINARIES

This work start by reviewing the Continuous Event-Line Constraint (CELC), a fundamental geometric incidence relationship that establishes a link between events and the translational velocity of the camera. Given the difficulty of extracting and matching sparse features in event streams and the fact that events present high sensitivity to edges, this work choose a higher-level representation of the environment in order to formulate the geometry: lines. The section therefore also briefly reviews two representations of 3D lines—Plücker line coordinates and their orthonormal representation—which are utilized within the later nonlinear optimization back-end. The notation used throughout the chapter is introduced alongside the theory.

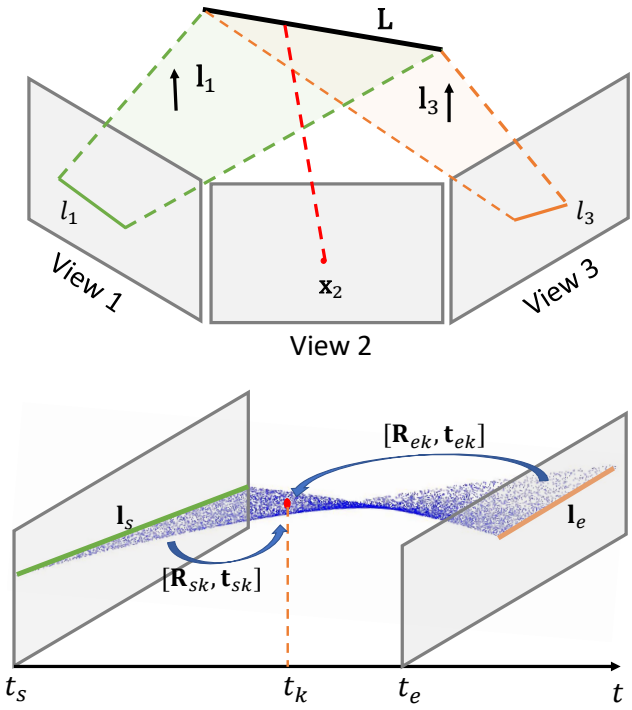


FIGURE 6.2: Geometry of CELC which indicates the relationship between events, lines and camera ego-motion. This work extract the intermediate line representations $\mathbf{l}_{sj}$ and $\mathbf{l}_{ej}$ by choosing two events in a small time interval Δt at the beginning and at the end of the interval.

6.1.1 *The Continuous Event-Line Constraint (CELC)*

CELC is a constraint proposed in our previous work [107]. It expresses the relation between the events generated by a moving observed line and the event camera's ego-motion using trifocal tensor geometry. Rather than using individual line feature detections as for example given by the ELiSeD detector by Brandli et al. [190], this work adopt the more general method of Le Gentil et al. [103], which identifies clusters in the space-time volume of events, each one being generated by a moving line observation in the image (i.e. a moving straight appearance boundary). By using this method, the event data is essentially left in its original form.

Let us denote the j-th cluster of events $\mathcal{E}_j$. Let $\mathcal{E}_j$ span a time interval $[t_{sj}, t_{ej}]$ in the space-time volume of events, and let e_{ij} furthermore represent the i-th event in the cluster. A 3D line has 4 degrees of freedom, which parametrize by the projections of the line onto the virtual camera frames at timestamps t_{sj} and t_{ej}. Let $\mathbf{l}_{sj} = \mathbf{K}^T \boldsymbol{\lambda}_{sj}$ and $\mathbf{l}_{ej} = \mathbf{K}^T \boldsymbol{\lambda}_{ej}$ denote the normalized plane coordinates corresponding to these line observations, where $\boldsymbol{\lambda}_{sj}$ and $\boldsymbol{\lambda}_{ej}$ denote the image-based representation of those lines (note that in this work $\boldsymbol{\lambda}_{sj}$ and $\boldsymbol{\lambda}_{ej}$ are only auxiliary variables representing the unknown line in 3D, and they will later on be hypothesized as a function of individual events), and $\mathbf{K}$ denotes an upper-triangular intrinsic camera matrix.

Finally, let us assume that the camera exhibits constant velocity motion in the local camera frame during the entire time interval. For any event $e_{ij} = \{x_{ij}, y_{ij}, t_{ij}, s_{ij}\}$ with timestamp t_{ij}, normalized coordinates $\mathbf{f}_{ij} = \mathbf{K}^{-1}[x_{ij}\ y_{ij}\ 1]^T$, and polarity s_{ij} (unused in the algorithm, but used when tracking lines.), we then have

$$\mathbf{f}_{ij}^{\mathsf{T}} \mathbf{B}_{ij} \mathbf{v} = 0, \tag{6.1}$$

where

$$\mathbf{B}_{ij} = \begin{bmatrix} (t_{ij} - t_{ej}) \mathbf{l}_{sj}^{\mathsf{T}} \mathbf{r}_1^{sij} \mathbf{l}_{ej}^{\mathsf{T}} - (t_{ij} - t_{sj}) \mathbf{l}_{ej}^{\mathsf{T}} \mathbf{r}_1^{eij} \mathbf{l}_{sj}^{\mathsf{T}} \\ (t_{ij} - t_{ej}) \mathbf{l}_{sj}^{\mathsf{T}} \mathbf{r}_2^{sij} \mathbf{l}_{ej}^{\mathsf{T}} - (t_{ij} - t_{sj}) \mathbf{l}_{ej}^{\mathsf{T}} \mathbf{r}_2^{eij} \mathbf{l}_{sj}^{\mathsf{T}} \\ (t_{ij} - t_{ej}) \mathbf{l}_{sj}^{\mathsf{T}} \mathbf{r}_3^{sij} \mathbf{l}_{ej}^{\mathsf{T}} - (t_{ij} - t_{sj}) \mathbf{l}_{ej}^{\mathsf{T}} \mathbf{r}_3^{eij} \mathbf{l}_{sj}^{\mathsf{T}} \end{bmatrix}. \tag{6.2}$$

$\mathbf{r}_1^{sij}$, $\mathbf{r}_2^{sij}$, $\mathbf{r}_3^{sij}$ mark the columns of the rotation $\mathbf{R}_{sij} = \exp(\lfloor \boldsymbol{\omega} \rfloor_\times (t_{ij} - t_{sj}))$ from the camera pose at time t_{ij} to the pose at time t_{sj}, while $\mathbf{r}_1^{eij}$, $\mathbf{r}_2^{eij}$, $\mathbf{r}_3^{eij}$ are the columns of the rotation $\mathbf{R}_{eij} = \exp(\lfloor \boldsymbol{\omega} \rfloor_\times (t_{ij} - t_{ej}))$ from the camera pose at time t_{ij} to the pose at time t_{ej}. Here uses $\boldsymbol{\omega}$ to represent the angular velocity, and $\lfloor \boldsymbol{\omega} \rfloor_\times$ is the 3×3 matrix skew symmetric matrix form of ω. $\mathbf{t}_{sij} = (t_{ij} - t_{sj})\mathbf{v}$ and $\mathbf{t}_{eij} = (t_{ij} - t_{ej})\mathbf{v}$ are the corresponding translations of these relative poses. $\boldsymbol{\omega}$ denotes the known rotational velocity taken from the IMU, and $\mathbf{v}$ the translational velocity that this work wish to identify. The constraint is derived from a continuous-time adaptation of the trifocal tensor, which uses a constant-velocity motion assumption. The geometry is illustrated in Fig. 6.2.

Note that in the following, this work assume that the camera and the IMU are mounted close enough such that the physical distance between the two sensors can be ignored and set to 0. Note furthermore that this work

assume that the system is calibrated and that the rotational transformation between the IMU and the camera is known. Without loss of generality, IMU signals are assumed to be pre-rotated and the camera and IMU frames are assumed to be identical (henceforth denoted as the body frame).

Given a sequence of events with M line clusters $\mathcal{E}_j$ where $j = 1, 2..., M$, the constraints from each event cluster with N_j events can be stacked into the single linear problem

$$\begin{bmatrix} \mathbf{B}_{11}^{\mathsf{T}}\mathbf{f}_{11} & \dots & \mathbf{B}_{ij}^{\mathsf{T}}\mathbf{f}_{ij} & \dots & \mathbf{B}_{N_M M}^{\mathsf{T}}\mathbf{f}_{N_M M} \end{bmatrix}^{\mathsf{T}} \mathbf{v} = \mathbf{0}. \tag{6.3}$$

With known angular velocity, (6.3) is linear in $\mathbf{v}$. By ignoring the presence of outliers and assuming known line reprojections at t_{sj} and t_{ej}, the translational velocity under an algebraic error criterion can be obtained from a simple SVD. Note that, in analogy to monocular structure-from-motion, results are determined only up to an unknown scale factor.

6.1.2 *Line Representation Methods*

A 3D line has 4 degrees of freedom (DoF). 3D lines may be represented in various ways [191]. One of the most common representations is given by Plücker line coordinates, a representation that easily enables geometric 3D lines transformations as linear operations. However, the representation is non-minimal and has 6 DoF. In order to perform minimal 4-DoF updates within non-linear optimization—a key requirement in back-end optimization—this work therefore also make use of the orthonormal representation proposed by Bartoli et al. [191]. The combined use of both representations is consistent with the works of Zhang et al. [192] and He et al. [131]. The following will review both of them.

6.1.2.1 *Plücker line coordinates*

As shown in Fig. 6.3, a 3D line $\mathbf{L}$ is represented by the Plücker line coordinates $\mathbf{L} = [\mathbf{d}^\top, \mathbf{m}^\top]^\top \in \mathbb{R}^6$, where the 3D vector $\mathbf{d}$ is the direction vector of the line, and the 3D vector $\mathbf{m}$ is the moment vector which is normal to the plane $\boldsymbol{\pi}$ determined by line $\mathbf{L}$ and the origin. $\mathbf{m}$ is perpendicular to $\mathbf{d}$ ($\mathbf{m}^\top\mathbf{d} = 0$), and is defined as

$$\mathbf{p} \times \mathbf{d} = \mathbf{m}, \tag{6.4}$$

where $\mathbf{p}$ is a vector from C to any arbitrary point on line $\mathbf{L}$. Note that $\mathbf{m}$ and $\mathbf{d}$ do not need to be unit vectors, but their ratio $\|\mathbf{m}\|/\|\mathbf{d}\|$ defines the

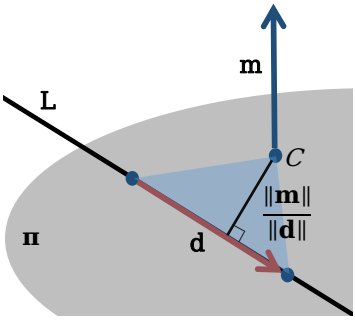


FIGURE 6.3: The geometry of Plücker line coordinates.

orthogonal distance between the origin C and the line $\mathbf{L}$. This work typically set $\mathbf{d}$ as a unit vector, that is $\|\mathbf{d}\| = 1$. Naturally, the 6-dimensional Plücker vector has two side constraints, which is in agreement with the 4 DoF of a 3D line.

While the existence of side constraints make Plücker line coordinates inconvenient to be applied in nonlinear optimization, they are very useful to construct geometric transformations with lines. For a given transformation from frame i to frame j, with rotation $\mathbf{R}_{ji}$ and translation $\mathbf{t}_{ji}$, the corresponding line geometry transformation [193] using Plücker line coordinates is

$$\begin{bmatrix} \mathbf{m}_j \\ \mathbf{d}_j \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{ji} & [\mathbf{t}_{ji}]_\times \mathbf{R}_{ji} \\ \mathbf{0} & \mathbf{R}_{ji} \end{bmatrix} \begin{bmatrix} \mathbf{m}_i \\ \mathbf{d}_i \end{bmatrix}. \tag{6.5}$$

6.1.2.2 *Orthonormal representation*

Based on Plücker line coordinates, the orthonormal representation is a minimal representation of a 3D line that can be regarded as the tangential space around a given Plücker coordinate. It is well suited for nonlinear optimization, and can be converted back-and-forth to regular Plücker line coordinates. For a given Plücker line $\mathbf{L} = [\mathbf{d}^\top, \mathbf{m}^\top]^\top$, the corresponding orthonormal representation $(\mathbf{U}, \mathbf{W}) \in SO(3) \times SO(2)$ can be obtained by using the QR decomposition

$$\begin{bmatrix} \mathbf{m} & \mathbf{d} \end{bmatrix}_{3\times 2} = \mathbf{U}_{3\times 3} \Sigma_{3\times 2}, \tag{6.6}$$

where

$$\begin{aligned} \mathbf{U}_{3\times 3} &= \left[\frac{\mathbf{m}}{\|\mathbf{m}\|} \quad \frac{\mathbf{d}}{\|\mathbf{d}\|} \quad \frac{\mathbf{m}\times\mathbf{d}}{\|\mathbf{m}\times\mathbf{d}\|}\right], \\ \Sigma_{3\times 2} &= \begin{bmatrix} \| \mathbf{m} \| & 0 \\ 0 & \| \mathbf{d} \| \\ 0 & 0 \end{bmatrix}. \end{aligned} \tag{6.7}$$

Note that vector $(\| \mathbf{m} \|, \| \mathbf{d} \|)^\mathsf{T}$ has only one DoF [131, 191], hence matrix Σ can be represented in a compressed way by using an $SO(2)$ matrix

$$\begin{aligned} \mathbf{W} &= \frac{1}{\sqrt{\| \mathbf{m} \|^2 + \| \mathbf{d} \|^2}} \begin{bmatrix} \| \mathbf{m} \| & - \| \mathbf{d} \| \\ \| \mathbf{d} \| & \| \mathbf{m} \| \end{bmatrix} \\ &= \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} = \begin{bmatrix} w_1 & -w_2 \\ w_2 & w_1 \end{bmatrix}, \end{aligned} \tag{6.8}$$

where ϕ is a rotation angle. $\mathbf{U}$ and $\mathbf{W}$ are actually rotation matrices of three and two dimensions respectively, and they can be locally updated during optimization using

$$\begin{aligned} \mathbf{U} &\leftarrow \mathbf{U}\mathbf{R}(\boldsymbol{\theta}), \\ \mathbf{W} &\leftarrow \mathbf{W}\mathbf{R}(\theta), \end{aligned} \tag{6.9}$$

with

$$\begin{aligned} \boldsymbol{\theta} &= \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \end{bmatrix}^\top, \\ \mathbf{R}(\boldsymbol{\theta}) &= \mathbf{R}_x(\theta_1)\mathbf{R}_y(\theta_2)\mathbf{R}_z(\theta_3), \end{aligned} \tag{6.10}$$

where $\mathbf{R}_x(\theta_1)$, $\mathbf{R}_y(\theta_2)$ and $\mathbf{R}_z(\theta_3)$ are 3D rotation matrices in $SO(3)$ around the x, y, and z axes with angles θ_1, θ_2 and θ_3, respectively. $\mathbf{R}(\theta)$ is a 2D rotation matrix in $SO(2)$. Thus, the four parameters for the construction of a minimal update are defined by $\mathbf{p} = [\boldsymbol{\theta}^\top, \theta]^\top$. The orthonormal representation can be directly converted to Plücker line coordinates by

$$\begin{aligned} \mathbf{L} &= [\mathbf{d}^\top, \mathbf{m}^\top]^\top \\ &= [\omega_2 \mathbf{u}_2^\top, \omega_1 \mathbf{u}_1^\top]^\top, \end{aligned} \tag{6.11}$$

where $\mathbf{u}_1$ and $\mathbf{u}_2$ are the first and second columns of $\mathbf{U}$ respectively.

6.2 INITIALIZATION

This work divide the whole pipeline into two parts: initialization and back-end optimization. Note that most event-based visual inertial systems pay little attention to the initialization part. For instance, IDOL [103] does not address initialization, and the sensor is assumed to be static for initialization in [98]. This work adopt the method of Le Gentil et al. [103] to cluster the events into groups each one generated by one moving line observation. This work operate over a thin temporal slice (about 0.1s$\sim$0.2s) of the events, and only consider clusters that stretch over the entire slice. In the following, the M clusters in discussion are simply the clusters that contain events over the temporal window. The starting and ending times of all clusters can hence be assumed equal to the starting and ending time of the temporal slice, i.e. $t_{sj} = t_s,\ t_{ej} = t_e,\ \forall j = 1, \ldots, M$. Then make use of CELC (cf. Section 6.1.1) within a two-layer nested RANSAC scheme for line-based translational velocity bootstrapping. Rather than using all events to construct Eq. (6.3), and create a hypothesis for the linear velocity $\mathbf{v}$ by sampling a small set of events from two different line clusters within the outer RANSAC layer. The hypothesised linear velocity is then verified as part of the inner RANSAC layer, which aims at robust regression of each 3D line using a novel 4-event minimal solver. The average geometric support expressed by the individual inlier ratios then implicitly makes a statement about the quality of the hypothesised dynamic motion parameters. The following will introduce the detailed functionality of both layers.

6.2.1 *Outer Layer RANSAC*

This work first introduce the outer layer RANSAC as described in Alg. 11. Note that in order to solve (6.3), at least 2 rows are required. Note furthermore that using constraints from a single cluster is not enough, as it is intuitively clear that this would leave the component of the velocity that is parallel to that 3D line unobserved [194]. This problem is also known as the *aperture problem*. Therefore need to sample events from at least two clusters, start by randomly sampling 2 clusters $\mathcal{E}_{j_1}$ and $\mathcal{E}_{j_2}$ from the entire set of clusters, and use each one to construct one row in (6.3). In each cluster randomly sample 5 events. The first two events—$e_{i_1 j}$ and $e_{i_2 j}$— are located at the beginning of the time interval to form the line observation $\mathbf{l}_{sj} = \mathbf{K}^\top([x_{i_1 j}\ y_{i_1 j}\ 1]^\top \times [x_{i_2 j}\ y_{i_2 j}\ 1]^\top)$. The last two events—$e_{i_4 j}$ and $e_{i_5 j}$—are located at the end of the time interval and used to form the line observation

Algorithm 2: Outer Layer RANSAC for Event-based Linear Velocity Estimation

Input: Event sequence, angular velocity $\boldsymbol{\omega}$ and intrinsic matrix K
Output: Linear velocity $\mathbf{v}$

Line clustering to get M event clusters $\mathcal{E}_j$ where $j = 1, 2..., M$
while $p_{mean} < p_{thres}$ && $k < Max\ Iteration$ **do**
  Sample 2 lines (clusters) and 5 events per line;
  Find $\mathbf{v}$ by linear solver in (6.3), $\|\mathbf{v}\| = 1$;
  for *each cluster* **do**
    Inner layer RANSAC (Alg. 7) to get $\mathbf{L}$ and p;
  end
  k++;
  Get average inlier percentage p_{mean} in Sec. 6.2.3;
end
return $\mathbf{v}$.

$\mathbf{l}_{ej} = \mathbf{K}^\top([x_{i_4 j}\ y_{i_4 j}\ 1]^\top \times [x_{i_5 j}\ y_{i_5 j}\ 1]^\top)$. The center event is finally used to substitute $\mathbf{f}_{ij}$ by $\mathbf{f}_{i_3 j}$ in (6.1), and construct the CELC constraint. Note that, in order to make $\mathbf{l}_{sj}$ and $\mathbf{l}_{ej}$ as accurate as possible, the first two events need to be sufficiently close in time and sufficiently spaced in the image. Here request a pixel distance of at least 3 pixels, and constrain the events to lie in a small sub-interval Δt at the beginning of the interval $[t_s, t_e]$ (cf. Fig. 6.2). Similar conditions are imposed on the last two events, except that the sub-interval is located towards the end of $[t_s, t_e]$. To conclude, the center event is constrained to lie within $[t_s/3, 2t_e/3]$. Adding these constraints improves the conditioning of the CELC constraint by avoiding near degenerate cases, such as the camera moving along a straight line with out rotation [107].

6.2.2 *Inner Layer RANSAC*

With an initial velocity hypothesis in hand, proceed to the inner RANSAC layer which aims at robust regression of each 3D line. The average geometric support (i.e. inlier ratio) for each line then later serves as a criterion to judge the quality of the initial velocity hypothesis. This work adopt the classical RANSAC framework [165] for the 3D line estimation and propose a 4-event, closed-form minimal solver to hypothesize 3D lines. The algorithm is outlined in Alg. 7. Given that the processing is individual for each cluster, index j is dropped in the following.

Algorithm 3: Inner Layer RANSAC for 3D Line Estimation

Input: event cluster $\mathcal{E}$, angular velocity $\boldsymbol{\omega}$, linear velocity $\mathbf{v}$ and intrinsic matrix K
Output: Plücker line $\mathbf{L}$ and inlier ratio p

```
while k < Max Iteration do
    Sample 4 events;
    Solve Plücker line by the event-based minimal line solver in Sec. 6.2.2.1 with constraints (6.12) and (6.13);
    Evaluate by the angular distance metric from Sec. 6.2.2.2;
    k++;
end
return L, p.
```

6.2.2.1 *Event-based Minimal Line Solver*

Given $\boldsymbol{\omega}$ and $\mathbf{v}$, each $\mathbf{f}_i$—the normalized coordinates of e_i pointing at a 3D point on the 3D line $\mathbf{L}$ from the camera pose at t_i—may be compensated by the rotation $\mathbf{R}_{si}$, thus resulting in $\mathbf{f}_i^{\omega} = \mathbf{R}_{si}\mathbf{f}_i$. We may furthermore obtain the start of this spatial direction vector by utilizing the camera center at time t_i given by $\mathbf{t}_{si} = \mathbf{v}(t_i - t_s)$.

The measurement of the point on $\mathbf{L}$ can hence be expressed by the Plücker line coordinates $[\mathbf{f}_i^{\omega\top}, (\mathbf{t}_{si} \times \mathbf{f}_i^{\omega})^{\top}]^{\top}$. Let $\mathbf{L} = [\mathbf{d}^{\top}, \mathbf{m}^{\top}]^{\top}$ furthermore be the Plücker coordinates of the 3D line. Finally, our incidence relation is simply given by the linear, Plücker coordinate-based line-line crossing constraint

$$\left[\mathbf{f}_i^{\omega\top}, (\mathbf{t}_{si} \times \mathbf{f}_i^{\omega})^{\top}\right] \begin{bmatrix} \mathbf{m} \\ \mathbf{d} \end{bmatrix} = \mathbf{c}^{\mathsf{T}}\mathbf{x} = 0. \tag{6.12}$$

The geometry is illustrated in Fig. 6.4. By stacking the vectors $\mathbf{c}$ for four events, a 4×6 matrix $\mathbf{C}$ is obtained. We have $\mathbf{C} = \mathbf{U}\mathbf{D}\mathbf{V}^{\top}$ by SVD, where the diagonal entries d_i of $\mathbf{D}$ are in descending numerical order. Then the general solution is $\mathbf{x} = \lambda_{n-1}\mathbf{v}_{n-1} + \lambda_n\mathbf{v}_n$, where $\mathbf{v}_{n-1}$ and $\mathbf{v}_n$ are the last 2 columns of $\mathbf{V}$.

The linear problem is obviously under-constrained owing to the fact that $\mathbf{L}$ has only 4 degrees of freedom, and only 4 events have been used. Extra constrains on the solution variable are needed, which are given by

$$\begin{cases} \|\mathbf{d}\|_2 = 1 \\ \mathbf{m}^{\top}\mathbf{d} = 0. \end{cases} \tag{6.13}$$

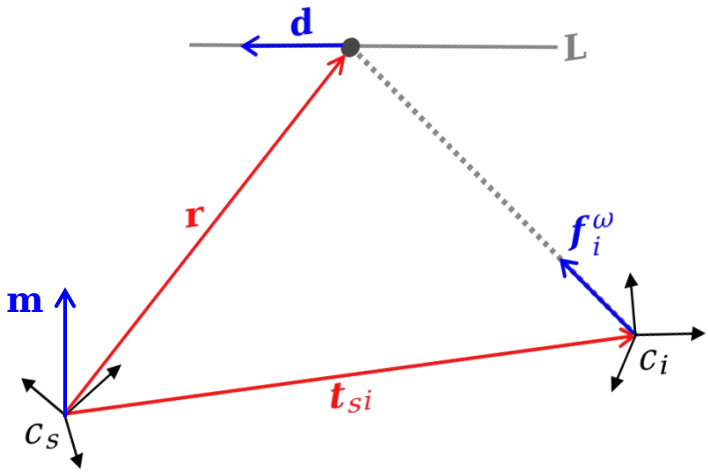


FIGURE 6.4: Geometry of the Plücker coordinates based line solver. The unknowns are the Plücker line, given by $\mathbf{d}$ and $\mathbf{m} = \mathbf{r} \times \mathbf{d}$, $\mathbf{r}$ is the position of the 3D point with respect to c_s. The measured or known variables are the landmark observation vector f_i^ω and the position of the camera center c_i at timestamp t_i with respect to c_s, given by $\mathbf{t}_{si}$.

In order to get λ_{n-1} and λ_n, we therefore need to solve the quadratic polynomial equation system given by

$$\begin{cases} \mathbf{x}(\lambda_{n-1}, \lambda_n)^\top (4:6) \cdot \mathbf{x}(\lambda_{n-1}, \lambda_n)(4:6) = 1 \\ \mathbf{x}(\lambda_{n-1}, \lambda_n)^\top (1:3) \cdot \mathbf{x}(\lambda_{n-1}, \lambda_n)(4:6) = 0, \end{cases} \tag{6.14}$$

where (j : k) represent the *j*-th to *k*-th elements of one vector. Using the Sylvester Resultant [195] method we can eliminate λ_{n-1} from the two polynomials in Eq. (6.14) and obtain a fourth-degree polynomial equation in λ_n, which can be further reduced to a quadratic polynomial by substituting $\gamma = \lambda_n^2$. The roots of the quadratic polynomial are easily obtained in closed-form. We finally obtain 4 solution pairs $\{\lambda_{n-1}, \lambda_n\}$, i.e. 4 solutions for $\mathbf{x} = [\mathbf{m}^\top, \mathbf{d}^\top]^\top$. The correct solution from the 4 is selected by the following strategy. Intuitively, there are two lines which intersect with Plücker line $[\mathbf{f}_i^{\omega\top}, (\mathbf{t}_{si} \times \mathbf{f}_i^\omega)^\top]^\top$. The first one is the line passing through the origin of c_s and c_i, and the other line is the one we want to solve for — $\mathbf{L} = [\mathbf{d}^\top, \mathbf{m}^\top]^\top$.

- The first line can be formulate as $\mathbf{m} = [0, 0, 0]^\top$, $\mathbf{d} = \pm\mathbf{v}/\text{norm}(\mathbf{v})$. Theoretically, $\pm\mathbf{v} \cdot (\mathbf{t}_{si} \times \mathbf{f}_i^\omega) = \mathbf{f}_i^\omega \cdot (\pm\mathbf{v} \times \mathbf{v}(t_i - t_s)) = 0$, which makes the line meet constraint (6.12). It is easy to see that the line also fulfills constraint (6.13). These two solutions are easily excluded by judging if $\mathbf{d} \times \mathbf{v} = 0$.
- The remaining candidates are given by $[\mathbf{d}^\top, \mathbf{m}^\top]^\top$ and $[-\mathbf{d}^\top, -\mathbf{m}^\top]^\top$. They actually represent the same line, and we only need to randomly pick one of them.

6.2.2.2 *Inlier metric*

In order to obtain precise lines from RANSAC, the inlier metric is of paramount importance. We utilize an angular error metric. Without providing exhaustive details, $\mathbf{L} = [\mathbf{d}^\top, \mathbf{m}^\top]^\top$ in c_s can be transformed to c_i using the hypothesized motion parameters, and we denote the transformed line by $\mathbf{L}_{trans} = [\mathbf{d}_{trans}^\top, \mathbf{m}_{trans}^\top]^\top$. $\mathbf{m}_i = \frac{\mathbf{f}_i^\omega \times \mathbf{d}_{trans}}{\|\mathbf{f}_i^\omega \times \mathbf{d}_{trans}\|}$ should have a small angle with $\mathbf{m}'_{trans} = \frac{\mathbf{m}_{trans}}{\|\mathbf{m}_{trans}\|}$, and use the error $\epsilon_i = 1 - \mathbf{m}'^\top_{trans}\mathbf{m}_i$. To facilitate understanding, a flowchart of this process is drawned in Fig. 6.5. Note that here also use this metric in order to decide the sense of the velocity direction, which is only determined up-to-scale. If the direction needs to be reversed, it is indicated by line triangulations behind the camera, which in turn can be recognized by wrongly directed moment vectors $\mathbf{m}_{trans}$.

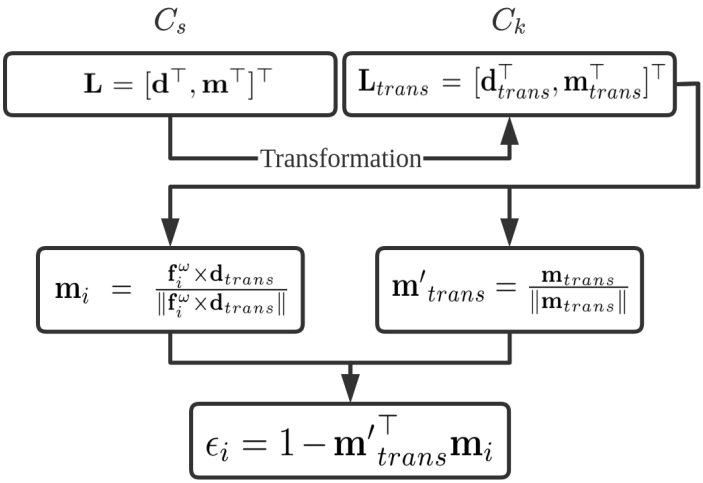


FIGURE 6.5: Flowchart of the proposed inlier metric.

6.2.2.3 *Efficiency Improvement*

Given there are many events, this work improve efficiency by randomly sub-sampling a maximum number of events from each cluster for line regression. The effect of sub-sampling is analyzed in Sec. 6.4.1.

6.2.3 *Convergence*

For each event e_{ij}, reproject each 3D line back to a virtual frame at time t_{ij} and evaluate the orthogonal event-to-line error. Given an inlier threshold, we may thus obtain an inlier ratio p_j for each line or cluster. If the estimated $\mathbf{v}$ is accurate, the inlier percentages p_j should all be simultaneously at a high level. Therefore, the metric we use to form a termination criterion in the outer layer RANSAC loop is given by

$$p_{mean} = \frac{\sum_{j=1}^{M} p_j}{M}, \tag{6.15}$$

and we terminate the algorithm as soon as this value exceeds a certain threshold p_{thres} or we reach the maximum number of iterations.

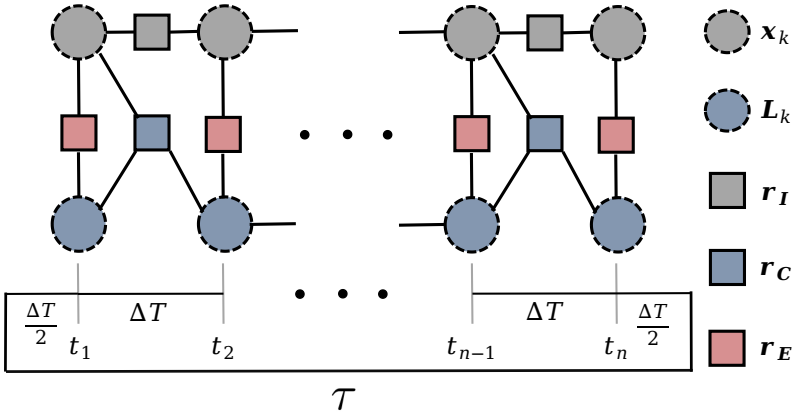


FIGURE 6.6: Graphical model and the defined temporal slices of the state variables and measurements involved in the optimization back-end.

6.3 BACK-END

We now proceed to the non-linear optimization back-end, a sliding-window tightly-coupled monocular visual-inertial velocity estimator. Fig. 6.6 illustrates the factor graph of our sliding window optimizer, where measurements are displayed as square boxes, and estimated variables as circular. The optimizer minimizes the continuous event-line errors over a larger time interval by parameterizing the velocity at multiple points in this interval and regularizing the estimated velocities via IMU preintegration terms.

6.3.1 *Formulation*

This work denote τ the length of the time interval over which the integral sliding window stretches. The latter is divided into n contiguous temporal slices of length $\Delta T = \frac{\tau}{n}$. The center time of each temporal slice is denoted by $t_1, \ldots, t_n$, and the IMU frame at t_k is denoted by $\mathcal{F}_k$. The full state vector in the sliding window is defined as

$$\begin{aligned} \mathcal{X} &= [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n, \mathbf{L}_1, \mathbf{L}_2, ..., \mathbf{L}_m], \\ \mathbf{x}_k &= [\mathbf{v}_k^w, \mathbf{q}_k^w, \mathbf{b}_{ak}, \mathbf{b}_{gk}], k = 1, ..., n, \end{aligned} \tag{6.16}$$

where $\mathbf{x}_k$ is the dynamic camera state at t_k. $\mathbf{x}_k$ contains the velocity $\mathbf{v}_k^w$ in the world frame, the accelerometer and gyroscope biases $\mathbf{b}_{ak}$ and $\mathbf{b}_{gk}$ expressed

in the body frame, and the exterior orientation of the body $\mathbf{q}_k^w$ with respect to a gravity-aligned world coordinate system expressed as a quaternion vector. The latter must be estimated in order to properly account for the gravity measurement of the accelerometer readings. m represents the total number of locally estimated line features. Each line $\mathbf{L}_k$ is defined locally within the reference frame of one temporal slice. This work represent the lines using Plücker line coordinates which simplifies geometric line transformations, and interchangingly employ the orthonormal representation for nonlinear optimization. Note that line features are represented locally for each temporal slice, no global representations are used. As the same line may be observed in multiple subsequent temporal slices, it means that the same line may be represented multiple times across several contiguous temporal slices. This work add further regularization terms to avoid strong deviations between such duplicate representations.

Formally, our objective consists of finding the maximum likelihood estimate by minimizing the cost function

$$\mathcal{X}^* = \underset{\mathcal{X}}{\operatorname{argmin}}\, F(\mathcal{X}), \tag{6.17}$$

where

$$F(\mathcal{X}) = \sum_{\alpha_i \in \mathcal{A}} \rho(\|\mathbf{r}_E^{\alpha_i}\|^2_{\Sigma_{\mathbf{r}_E^{\alpha_i}}}) + \sum_{\beta_j \in \mathcal{B}} \|\mathbf{r}_C^{\beta_j}\|^2_{\Sigma_{\mathbf{r}_C^{\beta_j}}} + \sum_{k=1}^{n-1} \|\mathbf{r}_I^k\|^2_{\Sigma_{\mathbf{r}_I^k}}. \tag{6.18}$$

$\rho(s)$ is the Huber loss [196], $\mathbf{r}_E$ and $\mathbf{r}_I$ are the residuals for the event camera and IMU measurements, respectively, and $\mathbf{r}_C$ is the above-mentioned consistency term for corresponding 3D line representations in different reference frames. $\mathcal{A}$ is the set of all possible associations $\alpha_i = \{a_i, l_i\}$ between the a_i-th event and the l_i-th line in the current sliding window. By definition, if the timestamp of the a_i-th event t_{a_i} lies within the temporal interval of the k-th frame (i.e. $t_{a_i} \in [t_k - \frac{\Delta T}{2}; t_k + \frac{\Delta T}{2}]$), the l_i-th line $\mathbf{L}_{l_i}$ must be defined in the k-th reference frame. The association set $\mathcal{A}$ is simply defined by the cluster slices, all events from one cluster slice will be associated to the same local line representation. $\mathcal{B}$ is the set of all possible line-line tuples $\beta_j = \{l_j, l'_j\}$ such that $\mathbf{L}_{l_j}$ and $\mathbf{L}_{l'_j}$ are corresponding lines defined in two adjacent temporal slices. Again, the set $\mathcal{B}$ is defined by sliced event clusters, as a correspondence β_j is defined for each adjacent pair of cluster slices. $\Sigma_\bullet$ denotes the error space covariance corresponding to the residual $\bullet$. Details on $\mathbf{r}_E$, $\mathbf{r}_I$ and $\mathbf{r}_C$ are provided in the following. Practically, the optimizer is realized using the Ceres Solver [197].

6.3.2 *Event Measurement Term*

Given one $\alpha_i = \{a_i, l_i\}$, the formation of the corresponding reprojection error first requires the 3D line $\mathbf{L}_{l_i}$ to be transformed from its local reference frame to the camera coordinate system at the timestamp of the event e_{a_i}. Again, let the local reference frame be the k-th coordinate frame within the window, i.e. $t_{a_i} \in [t_k - \frac{\Delta T}{2}; t_k + \frac{\Delta T}{2}]$. Denote the instantaneous angular velocity and the unknown instantaneous linear velocity at t_k in $\mathcal{F}_k$ as $\boldsymbol{\omega}_k$ and $\mathbf{v}_k$, respectively, where $\mathbf{v}_k = \mathbf{R}_w^k(\mathbf{q}_k^w)\mathbf{v}_k^w$. Based on the locally constant velocity assumption, we can easily derive the relative rotation $\mathbf{R}(t_{a_i}, t_k)$ and translation $\mathbf{t}(t_{a_i}, t_k)$ between the above mentioned two frames. Using (6.5) obtain the transformed 3D line in the instantaneous body frame at the time of the event t_{a_i}. This work simply denote this line $\mathbf{L}$ for the remainder of this section. Since only a single event and a single line are addressed, this work furthermore simply use e to refer to e_{a_i}, and its pixel coordinates are given by e_x, e_y.

The reprojection error of the event e is defined as the distance between the event and the projected line, expressed in the normalized image plane of the current camera coordinate system. Event e can be projected onto the normalized image plane by

$$\mathbf{e} = \mathbf{K}^{-1} \begin{bmatrix} e_x \\ e_y \\ 1 \end{bmatrix}. \tag{6.19}$$

It is easy to see that the third element of $\mathbf{e}$ must be 1, thus $\mathbf{e}$ is on the normalized image plane.

Furthermore, by [131] and [192], the projection of a 3D line $\mathbf{L} = [\mathbf{d}^\top, \mathbf{m}^\top]^\top$ is determined by the $\mathbf{m}$ component only, not its direction vector $\mathbf{d}$. Therefore, $\mathbf{L}$ can be projected to the camera image plane by

$$\mathbf{l} = \mathcal{K}\mathbf{m}, \tag{6.20}$$

where $\mathcal{K} = f_x f_y \mathbf{K}^{-\top}$ is the projection matrix for a line feature.

In our case, when projecting a line to the normalized image plane, $\mathcal{K}$ is an identity matrix. Then, by the formula of the distance between a point and a line in a plane, the distance $d(\mathbf{e}, \mathbf{l})$ between $\mathbf{e}$ and $\mathbf{l}$ is easily deduced as

$$\mathbf{r}_E = d(\mathbf{e}, \mathbf{l}) = \frac{\mathbf{e}^{\mathrm{T}}\mathbf{l}}{\sqrt{l_1^2 + l_2^2}}. \tag{6.21}$$

l_1 and l_2 here are the first and second elements of $\mathbf{l}$, respectively. Assuming isotropic Gaussian noise on the event position in the image plane, it is fair to assume isotropic Gaussian noise on the normalized event coordinates. Owing to their linear appearance in the residual expression, their propagation into error space is the identity transformation, and hence the covariance reweighting in Eq. 6.18 merely appears as a constant factor. Combined with an overall balancing of the different terms in Eq. 6.18, the covariance reweighting of the event measurement term simply becomes a diagonal constant weighting factor.

6.3.3 *IMU Measurement Term*

Next explain the IMU measurement terms, which use pre-integrated IMU signals in order to regularize delta-velocities across the window. Common visual-inertial frameworks such as VINS-Mono [198] and OKVIS [117] are position-based, and hence require double integration of the IMU signals. As this may lead to fast error accumulation, this work conceive it an advantage in our framework that only single signal integrations are required.

The IMU measurements are affected by acceleration bias $\mathbf{b}_a$, gyroscope bias $\mathbf{b}_w$, and additive noise. The raw gyroscope and accelerometer measurements, $\hat{\boldsymbol{\omega}}$ and $\hat{\mathbf{a}}$, are given by

$$\begin{aligned} \hat{\mathbf{a}}_t &= \mathbf{a}_t + \mathbf{b}_{a_t} + \mathbf{R}_w^t \mathbf{g}^w + \mathbf{n}_a, \\ \hat{\boldsymbol{\omega}}_t &= \mathbf{!}_t + \mathbf{b}_{\omega_t} + \mathbf{n}_\omega, \end{aligned} \tag{6.22}$$

where $\mathbf{g}^w = [0, 0, g]^\top$ is the gravity vector in the world frame. This work assume that the additive noise $\mathbf{n}_a$ and $\mathbf{n}_\omega$ are Gaussian white noise. Acceleration bias and gyroscope bias are modeled as random walk.

As indicated in [198], for two consecutive temporal slices in t_k and t_{k+1}, multiple inertial measurements in $[t_k, t_{k+1}]$ can be integrated in $\mathcal{F}_k$ by

$$\begin{aligned} \beta_{k+1}^k &= \int_{t \in [t_k, t_{k+1}]} \mathbf{R}_t^k (\hat{\mathbf{a}}_t - \mathbf{b}_{a_t}) \, \mathrm{dt}, \\ \gamma_{k+1}^k &= \int_{t \in [t_k, t_{k+1}]} \frac{1}{2} \Omega (\hat{\boldsymbol{\omega}}_t - \mathbf{b}_{w_t}) \gamma_t^k \, \mathrm{dt}, \end{aligned} \tag{6.23}$$

where,

$$\Omega(\boldsymbol{\omega}) = \begin{bmatrix} -\lfloor \boldsymbol{\omega} \rfloor_\times & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^\top & 0 \end{bmatrix}, \tag{6.24}$$

and $\lfloor \boldsymbol{\omega} \rfloor_\times$ is the skew-symmetric matrix of $\boldsymbol{\omega}$. IMU measurements are discrete, and this work use the midpoint integration here.

The final IMU pre-integration based residuals become

$$\begin{aligned} \mathbf{r}_I &= \begin{bmatrix} \delta\boldsymbol{\beta}^k_{k+1} \\ \delta\boldsymbol{\theta}^k_{k+1} \\ \delta\mathbf{b}_a \\ \delta\mathbf{b}_g \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}^k_w(\mathbf{q}_k)(\mathbf{v}^w_{k+1} + \mathbf{g}^w\Delta T - \mathbf{v}^w_k) - \hat{\boldsymbol{\beta}}^k_{k+1} \\ 2[\mathbf{q}^{w\,-1}_k \otimes \mathbf{q}^w_{k+1} \otimes (\hat{\gamma}^k_{k+1})^{-1}]_{xyz} \\ \mathbf{b}_{a,k+1} - \mathbf{b}_{a,k} \\ \mathbf{b}_{w,k+1} - \mathbf{b}_{w,k} \end{bmatrix}, \end{aligned} \tag{6.25}$$

where $\delta\boldsymbol{\beta}^k_{k+1}$, $\delta\boldsymbol{\theta}^k_{k+1}$, $\delta\mathbf{b}_a$ and $\delta\mathbf{b}_g$ are IMU measurement residuals for velocity, quaternion orientation, acceleration bias and gyroscope bias, respectively. $[\cdot]_{xyz}$ denotes the extraction of the vector part of the quaternion $\mathbf{q}$, which is used for the representation of the error-state. $\hat{\boldsymbol{\beta}}^k_{k+1}$ and $\hat{\gamma}^k_{k+1}$ are the preintegrated IMU measurement terms between t_k and t_{k+1}. Note that for the propagated residual space uncertainties, the reader is kindly referred to [198].

6.3.4 *Consistency Term*

Our final residual is the consistency constraint between corresponding local 3D line representations in adjacent temporal slices. For a $\beta^j = \{l_j, l'_j\}$, let $\mathbf{L}_{l_j} = [\mathbf{d}^\top_{l_j}, \mathbf{m}^\top_{l_j}]^\top$ denote the line in the earlier temporal slice, and $\mathbf{L}_{l'_j} = [\mathbf{d}^\top_{l'_j}, \mathbf{m}^\top_{l'_j}]^\top$ denote the line from the later temporal slice. Let $\mathbf{L}_{l_j}$ furthermore be a line defined in the $k-1$-th local reference frame, and $\mathbf{L}_{l'_j}$ be a line defined in the k-th local reference frame. The consistency term is easily formulated by transforming $\mathbf{L}_{l'_j}$ from $\mathcal{F}_k$ to $\mathcal{F}_{k-1}$, and evaluating the difference to the line $\mathbf{L}_{l_j}$ originally expressed in $\mathcal{F}_{k-1}$. Similar to Sec. 6.3.2, we can obtain the relative rotation $\mathbf{R}_{k-1,k}$ and translation $\mathbf{t}_{k-1,k}$. By Eq. (6.5), we have

$$\begin{bmatrix} \tilde{\mathbf{m}}_{l_j} \\ \tilde{\mathbf{d}}_{l_j} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{k-1,k} & [\mathbf{t}_{k-1,k}]_\times \mathbf{R}_{k-1,k} \\ \mathbf{0} & \mathbf{R}_{k-1,k} \end{bmatrix} \begin{bmatrix} \mathbf{m}_{l'_j} \\ \mathbf{d}_{l'_j} \end{bmatrix}, \tag{6.26}$$

where $[\tilde{\mathbf{d}}_{l_j}^\top, \tilde{\mathbf{m}}_{l_j}^\top]^\top$ is the predicted Plücker line at time t_{k-1}, and $[\mathbf{d}_{l_j}^\top, \mathbf{m}_{l_j}^\top]^\top$ is the optimized Plücker line at time t_{k-1}, The residual expresses the consistency as

$$\mathbf{r}_C = \begin{bmatrix} \delta\mathbf{d}_{j,k}^{k-1} \\ \delta\mathbf{m}_{j,k}^{k-1} \end{bmatrix} = \begin{bmatrix} \angle(\mathbf{d}_{l_j}, \tilde{\mathbf{d}}_{l'_j}) \\ \mathbf{m}_{l_j} - \tilde{\mathbf{m}}_{l'_j} \end{bmatrix}, \tag{6.27}$$

the angle $\angle(\cdot)$ here is measured in radians. Note that for the line consistency term, the covariance reweighting is again formed by employing a constant diagonal matrix of weighting factors. However, different weights are used for $\delta\mathbf{d}_{j,k}^{k-1}$ and $\delta\mathbf{m}_{j,k}^{k-1}$ to balance penalties imposed on direction and moment errors, respectively.

6.3.5 *Further Details*

The bootstrapping algorithm proposed in Sec. 6.2 is only used at the very beginning of the estimation process. The attentive reader will notice that the scale of the problem is still left uninitialized. While closed-form solutions exist, here we simply propagate a consistent scale factor through-out the window by enforcing line-depth consistencies, and subsequently minimize the IMU measurement term over a single, scalar scale factor. Once initialized, each run of subsequent sliding window optimization is initialized by simply reusing the values of the previous run or—for new data slices—by appending a first-order integration of the inertial measurements to the most recent, already optimized dynamic camera pose. 3D lines are initialized using fast linear line triangulation as explained in the following.

The clustering algorithm of IDOL [103] returns the set of events triggered by a moving reprojected line. Specifically, the line clustering is continuous and therefore tells us the location of the image plane projection $\mathbf{l}$ of the 3D line $\mathbf{L}$ at any given time. If sufficient displacement has taken place, line triangulation will performed.

As mentioned in Sec. 6.1.2, Plücker line coordinates are convenient to be used for geometry-related operations. We use them for the initialization of the 3D lines as well. A 3D line can be obtained by the intersection of

two planes. Let $\boldsymbol{\pi}_s \in \mathbb{R}^4$ and $\boldsymbol{\pi}_e \in \mathbb{R}^4$ be the observation planes for a 3D line obtained at two different times but already expressed in a common reference frame. The dual Plücker representation $\mathbf{L}^*$ for a 3D line is formed by the intersecting $\boldsymbol{\pi}_s$ and $\boldsymbol{\pi}_e$ [67] as in

$$\begin{aligned}\mathbf{L}^* &= \boldsymbol{\pi}_s \boldsymbol{\pi}_e^\top - \boldsymbol{\pi}_e \boldsymbol{\pi}_s^\top \in \mathbb{R}^{4\times 4} \\ &= \begin{bmatrix} [\mathbf{d}]_\times & \mathbf{m} \\ -\mathbf{m}^\top & 0 \end{bmatrix}.\end{aligned} \tag{6.28}$$

The Plücker line coordinates $\mathbf{L} = [\mathbf{d}^\top, \mathbf{m}^\top]^\top$ are readily extracted. Note that there are several assumptions made here:

- The cluster effectively originates from a moving line in the image.
- The planes obtained from lines fitted to events at the beginning and the end of the event cluster are accurate and not too much influenced by noise and outliers.
- The line is observed for a sufficiently long time to be able to contribute to the accuracy of the estimation.

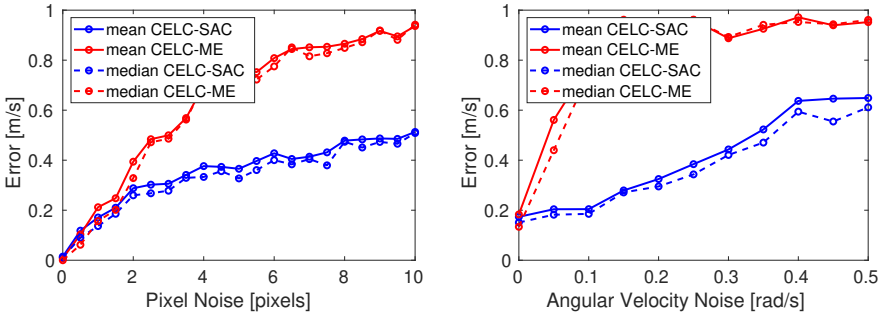
Those assumptions are generally not satisfied, and we need RANSAC to robustly initialize the lines. However, in practice, our experience has shown that the simple addition of the Huber loss $\rho(s)$ mentioned in Sec. 6.3.1 is sufficient to maintain stable and accurate estimation.

6.4 EXPERIMENTS

We proceed to the experimental validation of the method. Start with the velocity boot-strapping algorithm, and compare it against the M-estimator method proposed by Peng et al. [107] on both synthetic and real data. Finally, conclude with experimental results over the complete pipeline including sliding-window back-end optimization.

6.4.1 *Synthetic Data Results of the Velocity Initialization*

Here use this work's own simulator which generates events based on geometric models (such simulators are common in simulation experiments for traditional geometric vision chapters, e.g. [151, 172, 199]). The synthetic data is generated by randomly sampling 5 3D line segments in a cube



(a) (b)

FIGURE 6.7: Noise analysis. (a) Error as a function of noise on event locations. (b) Errors as a function of different levels of noise added to the angular velocity input. Errors generally increase with noise. Note that CELC-ME represents our previous work [107], while CELC-SAC means method presented in this chapter.

$([-2,2] \times [-2,2] \times [3,6])$ m^3 defined in the camera frame. Next, randomly generate angular and linear velocities by sampling random vectors $\boldsymbol{\omega} \in [0,1] \times [0,1] \times [0,1]$ rad/s, and random vectors $\mathbf{v} \in [1,1.5] \times [1,1.5] \times [1,1.5]$ m/s. Each event is generated randomly by choosing a 3D point on one of the lines and projecting it into the image plane with the camera pose sampled by a random timestamp within the interval $[0, 0.5]$ s. Note that the intrinsic matrix used here is taken from the real data used in the chapter. The pixel location of each generated event is finally disturbed by zero-mean Gaussian noise with a standard deviation of 1 pixel. Besides, 10% outliers was added to the data. Here also add an appropriate amount of Gaussian noise to the ground-truth endpoints of each starting and ending line pair l_{1j} and l_{3j} used in the reference by Peng et al. [107]. This reflects event location uncertainties, event distributions, and the dynamic nature of the line during the time interval δt. Note that this model does not necessarily adhere to a realistic event generation model but provides basic geometric evaluation cases in which simply have events generated randomly along the reprojected line.

The experiments analyze the proposed algorithm's performance for variations of both controlled and uncontrolled parameters: i) Evaluation of the solver's robustness against noise, varying speeds, varying outlier ratios, noise on angular velocity, and different numbers of lines; ii) Investigation of design parameter influences such as the interval size and the considered

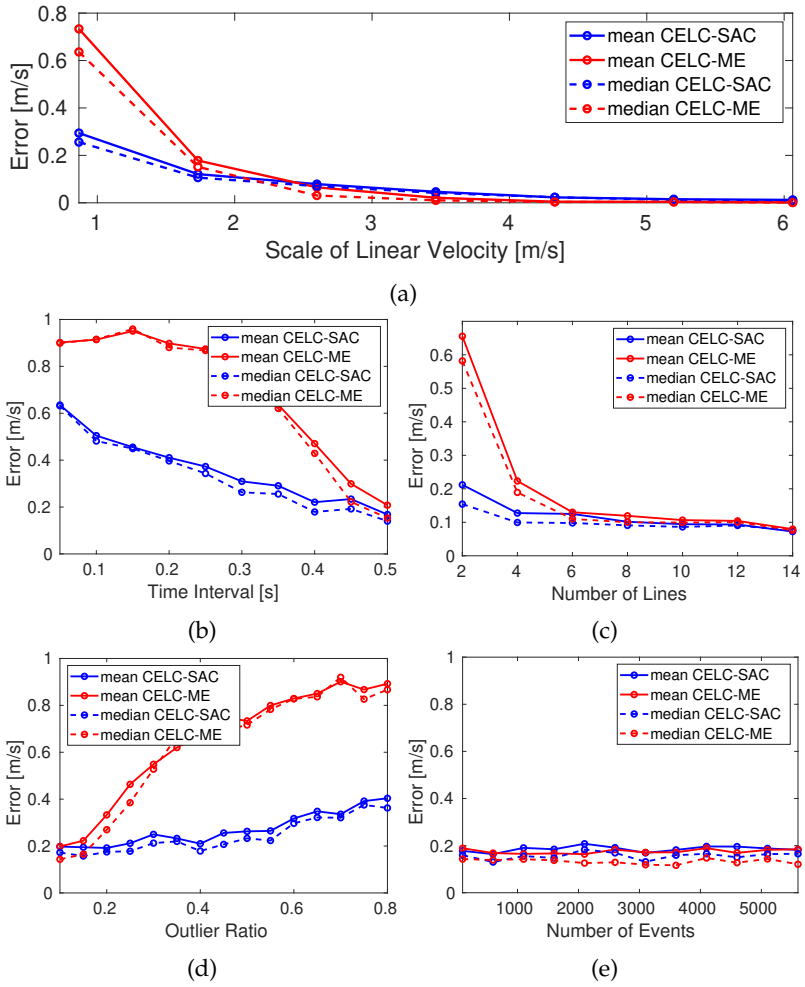


FIGURE 6.8: Accuracy for other motion or solver parameters. (a) Errors over an increasing scale of the velocity. (b) Error as a function of the time interval length. (c) Error for different numbers of observed lines. (d) Errors for different outlier ratios. (e) Errors for different numbers of events used within RANSAC.

number of events. This work evaluate the accuracy by the Euclidean distance ϵ between the estimated and the ground truth results, which is given by

$$\epsilon = \|\mathbf{v}_{\text{gt}} - \mathbf{v}_{\text{est}}\|_2, \tag{6.29}$$

where $\mathbf{v}_{\text{gt}}$ and $\mathbf{v}_{\text{est}}$ are the ground truth and estimated linear velocities, respectively. Note that, as the magnitude of the velocity cannot be found in the monocular case, it is manually set to ground truth. Both the mean and median of ϵ are indicated for both the proposed solver (CELC-SAC) and the reference implementation of our previous work [107] (CELC-ME). Results are shown in Fig. 6.7 and Fig. 6.8.

Robustness against event location noise: The disturbance of each event is varied with a standard deviation reaching from 0 to 10 pixels. As shown in Fig. 6.7a, our new approach is more robust than our previous work.

Robustness against noise in the angular velocity input: The noise is varied between 0 and 0.5 rad/s. Fig. 6.7b indicates the corresponding results. It is easy to see that our previous work is very sensitive to the quality of the IMU readings.

Effect of linear velocity: The coordinates of the linear velocity are randomly sampled from the range $[(0.5 + i \times 0.5), (1 + i \times 0.5)]$ m/s, $i = 0, ..., 6$. The magnitude of the linear velocity is manually reset to $\sqrt{(0.5 + i \times 0.5)^2 \times 3}$. The simulation results are shown in Fig. 6.8a. As can be observed, errors are decreasing with an increasing norm of the speed. Our solver performs better for low velocities, one of the primary weaknesses of the reference algorithm.

Effect of the time interval size: Vary the time interval from 0.05s to 0.5s. Results are indicated in Fig. 6.8b and show that the errors of all solutions are decreasing as the time interval is increasing. However, our newly proposed solver has higher accuracy for small intervals. Note that the constant velocity assumption only holds for small time intervals, which is why stronger performance in this situation is to be regarded as a substantial advantage over the reference implementation.

Effect of the number of lines: The number of lines is varied from 2 to 14. Fig. 6.8c presents the results. The accuracy of the solvers increases along with the number of lines. Moreover, our algorithm is more stable when fewer lines are observed.

Effect of the outlier ratio: Vary the outlier ratio from 0.1 to 0.8. As can be observed, the proposed solution shows high robustness against increasing outlier ratios (Fig. 6.8d).

Effect of the number of events: As detailed in Sec. 6.2.2.3, the set of events is subsampled in order to increase computational efficiency. The results for different numbers of events is shown in Fig. 6.8e. The number of events is varied from 100 to 5600. Results show that the number of events hardly affects performance.

Overall, the proposed algorithm is more robust against both increasing event location disturbances and higher noise levels in the inertial readings. As demonstrated by our experiments, our new solution achieves higher accuracy than our previous work [107] when the camera has smaller displacement, a crucial advantage for a direct small time interval-based solution of first-order kinematics.

To compare run times, here used the same settings as those described before. The time interval has been set to 0.3s. The event location noise is set to 0.5 pixels. Additionally, we've set the outlier ratio to 0.1. After 100 runs, our finding is that our algorithm and the algorithm from our previous work have mean run times of 0.0726s and 0.0095s, and median run times of 0.0746s and 0.0089s, respectively. Our algorithm is thus slower, which we trace back to the existence of the inner RANSAC scheme needed to obtain robust estimations of the 3D lines. However, it is important to note that still this work take less time than it takes to capture the data, hence the geometric estimation itself can be considered real-time. For reference, Contrast Maximization [88], takes 0.02-0.03s to process 0.01s of event data, which is below real-time capability. It is furthermore possible to optimize parameters by trading accuracy for efficiency (e.g. by limiting the maximum number of iterations), or speeding up execution by making use of parallel computing hardware.

6.4.2 *Real Data Results of the Velocity Initialization*

Next, evaluate the robust velocity initialization module on real data to verify the practicality of the method. Here directly evaluate the direction error between ground truth and the estimated linear velocity, which is given by

$$\theta = \arccos\left(\frac{\mathbf{v}_{gt}^{\mathsf{T}}\mathbf{v}_{est}}{\|\mathbf{v}_{gt}\|\|\mathbf{v}_{est}\|}\right). \tag{6.30}$$

The algorithm is evaluated over two datasets, which are collected by a DAVIS346 event camera with a resolution of 346×260 pixels. Ground truth is provided by an external motion tracking system. The first sequence is taken from [200] (Indoor45 9), which is captured by an unmanned aerial

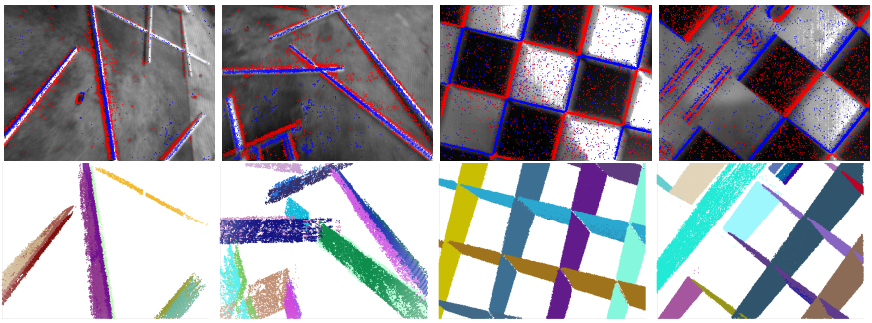


FIGURE 6.9: Examples taken from our real data experiments. The first row shows an example grayscale image captured during the time interval (unused) and the corresponding events (positive in blue, negative in red). The second row indicates the identified event clusters corresponding to real-world line segments in a Spatio-temporal view.

vehicle (UAV) carrying a DAVIS346 in a 45° downward-facing arrangement. The second dataset is collected by a small automated ground vehicle (AGV) and uses a downward facing camera. Examples are presented in Fig. 6.9.

Fig. 6.10 shows the histogram charts in polar coordinates of the angle errors θ, and Table 6.1 gives the mean and median errors. As can be seen, the distribution of our previous work (CELC-ME) is more scattered, while the distribution of our new approach (CELC-SAC) is more concentrated and has a smaller mean in the distribution region, thus the proposed method obtains more accurate results than our previous work [107], which is consistent with the results obtained in simulation. Note that the results of our previous work are worse than the results listed in [107] given that this work use a smaller time interval (about 0.1s vs 0.2s on the UAV dataset, and about 0.4s vs 0.7s on the AGV dataset), and this work do not favour the method by choosing smaller time intervals. Rather, smaller time intervals are a requirement because the constant velocity assumption is not valid over longer time intervals. This is why the newly proposed method furthermore succeeds in processing complete sequences rather than only subsets of the data in [107].

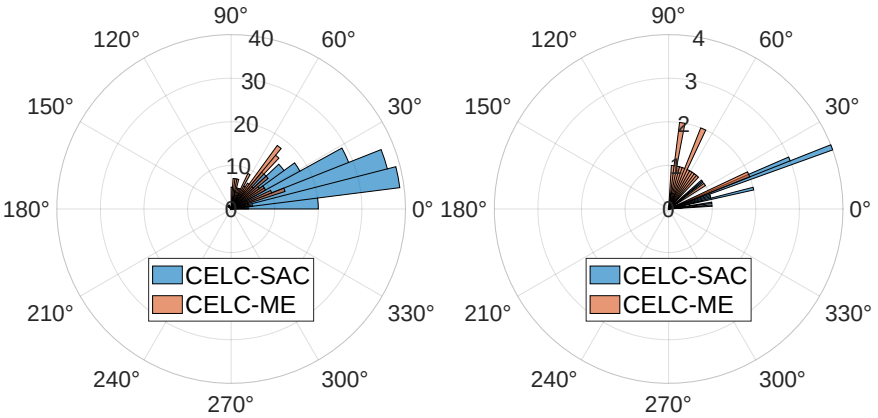


FIGURE 6.10: Histogram charts of angular errors in polar coordinates (unit: degrees). Note that CELC-ME represents our previous work [107], while CELC-SAC means the method presented in this chapter.

	CELC-SAC		CELC-ME	
	$\alpha(\theta)$	$\beta(\theta)$	$\alpha(\theta)$	$\beta(\theta)$
UAV	0.4515	0.3683	0.8214	0.8299
AGV	0.3517	0.3555	1.0531	1.1141

TABLE 6.1: Mean (α) and Median (β) angular errors on two datasets from a flying and a ground vehicle platform (unit: rad).

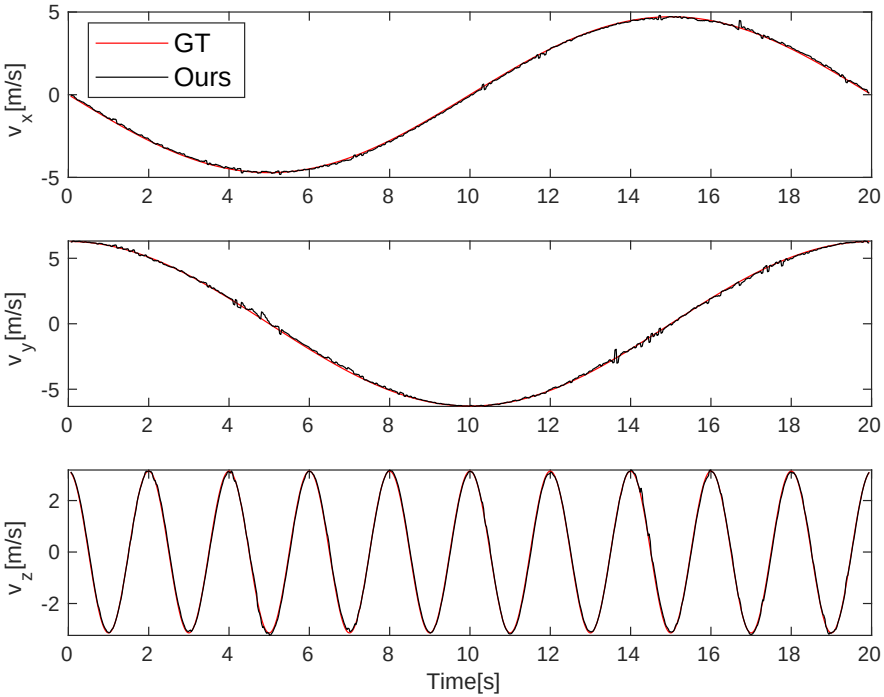


FIGURE 6.11: Velocity estimation by the proposed algorithm on synthetic data.

6.4.3 *Synthetic Data Results Including Back-end Optimization*

The synthetic data is generated using a modified version of the vio_data_simulation library[1]. The original version of the library is used to generate point and line features as measured by traditional cameras. This work modified it to generate events triggered by moving line observations.

The simulation experiments make use of 10 lines and the size of the sliding window is set to $\tau = 0.1s$. Furthermore, it is assumed that velocities are approximately constant within temporal sub-slices of 0.01s, and hence the overall sliding window is divided into 10 slices. The assumption shows sufficiently validity in practical scenarios, too. The estimated result versus ground truth is shown in Fig. 6.11. Mean and median errors are 0.1365 m/s and 0.1219 m/s, respectively, thus validating the general functionality of the complete framework.

1 https://github.com/HeYijia/vio_data_simulation

Sequence	Ultimate SLAM (Fr + E + I)				VINS				Ours			
	$\mu(\epsilon)$	$\nu(\epsilon)$	$\mu(\phi)$	$\nu(\phi)$	$\mu(\epsilon)$	$\nu(\epsilon)$	$\mu(\phi)$	$\nu(\phi)$	$\mu(\epsilon)$	$\nu(\epsilon)$	$\mu(\phi)$	$\nu(\phi)$
	[m/s]	[m/s]	[1]	[1]	[m/s]	[m/s]	[1]	[1]	[m/s]	[m/s]	[1]	[1]
Seq. 2	1.0383	1.0436	0.2846	0.2394	0.4683	0.4788	0.1550	0.1162	**0.4046**	**0.3563**	**0.0927**	**0.0794**
Seq. 4	0.8801	0.7462	0.2637	0.2062	0.5709	0.5771	0.2407	0.1656	**0.3468**	**0.2985**	**0.0868**	**0.0717**
Seq. 9	1.9077	1.8864	0.3097	0.3098	1.8285	1.7943	0.3105	0.3135	**0.6148**	**0.5327**	**0.1036**	**0.0946**

TABLE 6.2: Error comparison on different sequences (mean and median error).

6.4.4 *Real Data Results Including Back-end Optimization*

This work test the algorithm on real sequences of the UZH-FPV Drone Racing dataset [200]. The dataset is one of the most aggressive visual-inertial odometry dataset to date, and thus represents a significant challenge for state estimation, and a suitable candidate to illustrate the superiority of our algorithm. The UZH-FPV Drone Racing dataset contains many sensors. However, in this experiments only use data collected by the event camera (i.e. miniDAVIS346 (mDAVIS)), which provides events, regular frames, and inertial readings. The miniDAVIS346 (mDAVIS) is developed by iniVation[2] and has a spatial resolution of 346×260 pixels. It generates events with microsecond temporal resolution, and the provided grayscale frames are recorded at 50 Hz. In addition, this work used its built-in IMU, which reports inertial readings at 200Hz, and includes data from both a gyroscope and an accelerometer. The dataset contains many sequences. This work select three sequences with obvious line features in the scene and with publicly available ground truth, i.e. indoor 45° downward facing 2, 4 and 9. Their respective maximum velocities are 6.97 m/s, 6.55 m/s and 11.23 m/s. This work compare it against the open-source frameworks EVO [94], Ultimate SLAM [98], and VINS-Mono [198]. Given that EVO is a purely vision-based framework not using inertial readings, it requires a good map initialization, it is more suitable for slow scenarios and feature-rich environments. Due to the high dynamics and feature-deprived nature of the FPV sequences (especially in the beginning where there may be short periods with almost no features), EVO will experience tracking loss. Therefore, here only present the results of Ultimate SLAM and VINS-Mono.

The sliding window is set to $\tau = 0.1s$ and the length of the temporal sub-slices is set to $0.01s$, which is similar to the settings in the synthetic experiment. Here analyze the algorithm in one of two ways. The first one

2 https://inivation.com/buy/

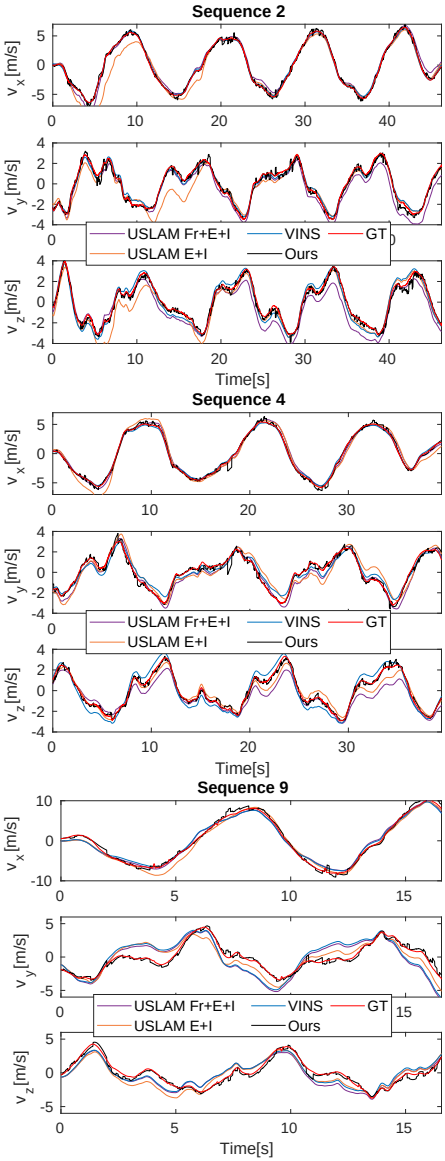


FIGURE 6.12: Estimated linear velocities compared against ground truth and VINS results on sequences.

Sequence	Ultimate SLAM (Fr + E + I)				VINS				Ours			
	std(ϵ)	max(ϵ)	std(ϕ)	max(ϕ)	std(ϵ)	max(ϵ)	std(ϕ)	max(ϕ)	std(ϵ)	max(ϵ)	std(ϕ)	max(ϕ)
	[m/s]	[m/s]	[1]	[1]	[m/s]	[m/s]	[1]	[1]	[m/s]	[m/s]	[1]	[1]
Seq. 2	0.5244	2.0186	0.1914	1.2596	**0.2163**	**1.5358**	**0.0433**	1.0894	0.2371	1.6830	0.0564	**0.3543**
Seq. 4	0.4552	1.8947	0.1156	1.0011	0.3008	**1.1049**	**0.0573**	1.0296	**0.2308**	2.1309	0.0726	0.8426
Seq. 9	0.6679	3.5409	0.0609	0.5336	0.6678	4.7264	0.0702	0.8853	**0.3695**	**2.4637**	**0.0526**	**0.3218**

TABLE 6.3: Error comparison on different sequences (standard deviation and maximum error).

Sequence	Ultimate SLAM (E + I)			
	$\mu(\epsilon)$	$\nu(\epsilon)$	$\mu(\phi)$	$\nu(\phi)$
	[m/s]	[m/s]	[1]	[1]
Seq. 2	1.2343	0.6956	0.2848	0.1579
Seq. 4	1.1104	1.1088	0.2659	0.2617
Seq. 9	1.5825	1.5014	0.2593	0.2288

TABLE 6.4: Error results of Ultimate SLAM (E + I) on different sequences (mean and median error).

is given by error statistics, the second one by a display of the estimated results together with the velocity estimation obtained by VINS and ground truth. Regarding error statistics, here use two different error metrics. The first is the absolute error, which is given by

$$\epsilon = \|\mathbf{v}_{gt} - \mathbf{v}_{est}\|_2, \tag{6.31}$$

where $\mathbf{v}_{gt}$ and $\mathbf{v}_{est}$ are the ground truth and estimated linear velocities, respectively. The second error metric is the relative error, which is given by

$$\phi = \frac{\|\mathbf{v}_{gt} - \mathbf{v}_{est}\|_2}{\|\mathbf{v}_{gt}\|_2}. \tag{6.32}$$

Ultimate SLAM has two modes: events-only mode (E + I) and mode including all sensors (Fr + E + I). The comparison results against Ultimate SLAM and VINS on different sequences are presented in Table 6.2 and Table 6.3, with their box plot error distribution shown in Fig. 6.13, (μ denotes the mean error, ν the median error, std the standard deviation, and max the maximum error). Note that due to space limitations, here has placed the results of Ultimate SLAM (E + I) in Table 6.4 and Table 6.5. As can be

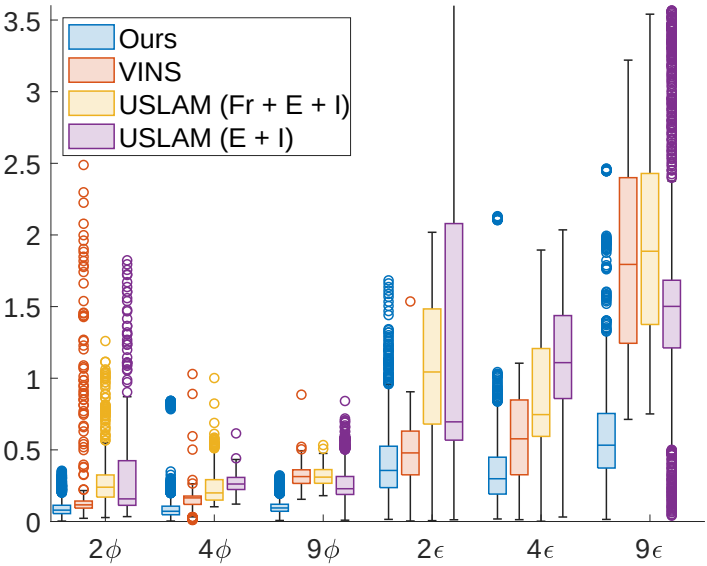


FIGURE 6.13: Box plot of error distribution on distinct sequences. The numbers here represent different sequences, while ϕ and ϵ represent different error metrics.

Sequence	Ultimate SLAM (E + I)			
	std(ϵ)	max(ϵ)	std(ϕ)	max(ϕ)
	[m/s]	[m/s]	[1]	[1]
Seq. 2	1.0920	5.1026	0.2756	1.8231
Seq. 4	0.3843	2.0355	0.0619	**0.6150**
Seq. 9	0.6883	4.4225	0.1218	0.8410

TABLE 6.5: Error results of Ultimate SLAM (E + I) on different sequences (standard deviation and the maximum error)

Sequence	Duration	Length	Max. Speed	Med. Speed
	[s]	[m]	[m/s]	[m/s]
Seq. 2	55.77	218.90	6.97	4.45
Seq. 4	47.36	168.06	6.55	4.22
Seq. 9	40.00	215.58	11.23	5.66

TABLE 6.6: Basic properties of the compared sequences.

observed, our estimation gives the best results under all metrics, except for the max metric of sequence 4, where Ultimate SLAM (E + I) achieved the best results. This is consistent with the results in Fig. 6.12. As can be observed, the estimated results are all closer to ground truth (GT) than the velocities obtained by Ultimate SLAM and VINS, especially in the last sequence in which the most challenging dynamics occur. The jumps in our results are due to the fact that there are some scenes in the sequences with very few line features. The specific error reduction rate compared against Ultimate SLAM and VINS for all sequences is provided in Table 6.7, 6.8 and 6.9.

The superior performance on the final dataset is grounded on two facts. The first one is the difference between traditional and event cameras. The elevated overall speed in sequence 9 is faster than in sequences 2 and 4 (as illustrated in Table 6.6), which causes an elevated amount of motion blur in the regular images. For example, Fig. 6.14 shows a selection of images captured in the same turns in three different sequences. The images taken from sequences 2 and 4 are relatively clear, while the image taken from

Sequence	Reduction Rate			
	$\mu(\epsilon)$ [%]	$\nu(\epsilon)$ [%]	$\mu(\phi)$ [%]	$\nu(\phi)$ [%]
Seq. 2	61.04	65.86	67.45	66.85
Seq. 4	60.59	59.99	67.07	65.22
Seq. 9	67.77	71.76	66.54	69.48

TABLE 6.7: Error reduction rate of our method compared to Ultimate SLAM (Fr + E + I).

Sequence	Reduction Rate			
	$\mu(\epsilon)$ [%]	$\nu(\epsilon)$ [%]	$\mu(\phi)$ [%]	$\nu(\phi)$ [%]
Seq. 2	67.22	48.77	67.47	49.75
Seq. 4	68.76	73.08	67.35	72.60
Seq. 9	61.15	64.52	60.04	58.67

TABLE 6.8: Error reduction rate of our method compared to Ultimate SLAM (E + I).

Sequence	Reduction Rate			
	$\mu(\epsilon)$ [%]	$\nu(\epsilon)$ [%]	$\mu(\phi)$ [%]	$\nu(\phi)$ [%]
Seq. 2	13.62	25.58	40.22	31.71
Seq. 4	39.24	48.28	63.93	56.71
Seq. 9	66.37	70.31	66.62	69.84

TABLE 6.9: Error reduction rate of our method compared to VINS.

sequence 9 has more motion blur. These images are indeed representative for the average image quality along the sequences. While motion blur easily affects frame-based algorithms such as VINS, and Ultimate SLAM (Fr + E + I) also uses images. This can also explain why Ultimate SLAM (E +

(a) (b) (c)

FIGURE 6.14: Comparison of grayscale frames while passing through the same scene in the three sequences. (a), (b), and (c) are from indoor 45° downward facing 2, 4, and 9 respectively.

I) achieved the best in the max metric for sequence 9. Our event-based solution indeed benefits from faster motion as this leads to larger, more easily observable line displacements during small time intervals, and thus better noise cancellation effects. The second argument in support of our newly proposed method is given by differences in the estimated state variables and consistency terms. Our method does not aim at absolute position estimation, hence first-order integrations of inertial signals are sufficient to formulate our optimization problem. This is reflected in our estimation results for sequences 2 and 4, which both reflect higher accuracy for the body-centric, event-inertial approach.

6.4.5 *Ablation Study for the Consistency Term*

This work also conducted an ablation study on the consistency term, to see if the consistency term can improve the system's robustness. For this reason, here optimize without consistency term on sequences 2, 4, and 9, and summarize the results in Table 6.10. As can be seen, compared against our results from Table 6.2, disabling the consistency term simply brakes the framework. Thus verify the effectiveness of the proposed consistency term and the validity of the proposed system design.

6.4.6 *Runtime Analysis*

As mentioned above, the two-layer RANSAC algorithm could be considered real-time and further pushed in computational efficiency by optimizing and tuning some of the RANSAC parameters. Furthermore, the autor believe

Sequence	Results Without Consistency Term			
	$\mu(\epsilon)$	$\nu(\epsilon)$	$\mu(\phi)$	$\nu(\phi)$
	[m/s]	[m/s]	[1]	[1]
Seq. 2	26.5778	27.1131	6.2653	5.7262
Seq. 4	23.9810	22.7299	6.1819	5.0613
Seq. 9	8.1640	7.1972	1.3389	1.1825

TABLE 6.10: Results without consistency term

that it would certainly be possible to make use of parallel computing techniques. Currently, the main obstacle towards real-time processing is given by our front-end line tracking method which is relatively slow, as this work utilize a point cloud processing method. It could again be sped up by considering parallelization or using other line tracking methods.

In regards to back-end optimization—and compared against other frameworks such as Ultimate SLAM [98]—it is worth mentioning that one of the main reasons why our algorithm is slower is because this work make use of all the events. Ultimate SLAM primarily focuses on creating virtual frames (event frames) from spatiotemporal event windows. It then carries out feature detection and tracking through traditional computer vision techniques, specifically using the FAST corner detector [201] and the Lucas-Kanade tracker [202]. This sparsifies data, and thus leads to much faster execution times. The author thus believe that our implementation could also be sped up by considering downsampling techniques.

6.5 CONCLUSION

This chapter introduces a novel method for visual-inertial fusion which estimates directly the velocity of the sensor system rather than its global position. This is interesting as it enables fail-safe motion control without depending on the construction or tracking of a globally consistent map. Another natural benefit of fusion at the velocity level is given by the fact that it only requires single integration of inertial readings. The main novelty consists of the use of an event camera instead of a regular frame-based camera. This work demonstrate that this is not only a highly intuitive choice for a dynamic vision sensor, but also beneficial one. As shown, velocity samples can be directly calculated from thin slices of the space-time

volume of events. Furthermore, our method indeed keeps performing well in highly dynamic scenarios where regular camera-based approaches fail. Our future work consists of extending the approach from line features to a combination of line and point features. The author also consider the addition of a global position and map estimation layer around the presently proposed velocity estimation framework. Our current version is not real-time as it involves processing a large number of events. Especially, using point cloud processing for line tracking will result in slow performance. As the processing frequency increases, it requires solving more state variables, and the current version has not yet been parallelized. The embedding of the method into a more suitable computing architecture is part of the ongoing work.

7

SUMMARY

7.1 CONCLUSION

This thesis reviews the latest research advancements in the field of Visual Simultaneous Localization and Mapping (SLAM) and delves into novel solutions for the challenging problem of relative pose estimation geared towards frame-based and neuromorphic cameras. It primarily utilizes the knowledge of multiview geometry to solve the relative pose estimation in challenging scenarios and applications such as generalized cameras, event cameras, etc. The main innovations include:

Through this study, the thesis propose for the first time a method based on convex optimization that provides a provably global optimal solution to the generalized relative pose problem. For the generalized relative pose and scale problem, the paper further proposes efficient and precise solution methods, including the first closed-form solution based on the 26-point correspondences, an innovative solver that relies only on 9 affine correspondences (AC), and a minimal solver for two affine correspondences with known directional correspondences.

Specifically, the minimal solver for the 7 degrees of freedom in the generalized relative pose and scale problem remains a challenging task. To better address this issue, the thesis explored the Gröbner basis technique in the field of algebraic geometry. Specifically, an innovative method has been proposed which works for permutation invariant polynomial problems.

However, under extreme lighting conditions and high-speed scenarios, traditional cameras fail to function, which is why the thesis introduced the use of event cameras and methods that fuse with inertial signals. Specifically, in the context of autonomous driving applications, the thesis adopted a continuous-time model based on the nonholonomic Ackermann motion model, the proposed method is a "1-track" solver that uses n-linearities with the reduced continuous motion model to produce scalar constraints on the rotational velocity. The thesis also implemented the fusion of event cameras with inertial sensors and innovatively developed a visual-inertial 3D velocity estimation technique based on line features.

7.2 OUTLOOK

Despite the groundbreaking progress made in various aspects of this study, the thesis have identified existing limitations and potential directions for future research. For instance, while the thesis have proposed effective algorithms to address the generalized relative pose and scale problems, the minimal solver under full degrees of freedom remains a challenge, necessitating further research and exploration. Additionally, as a novel sensor, research on event cameras is still in its early stages, with many related issues yet to be explored and resolved. Based on the findings of this thesis, future work could further explore minimal solutions for generalized relative pose and scale problems as well as more solutions for relative pose estimation related to event cameras:

- **Using Emerging Homotopy Continuation Techniques for Solving the 7 Degrees of Freedom in the Generalized Relative Pose and Scale Problem:** Homotopy continuation is a numerical method used in mathematics and engineering to solve systems of equations. It constructs a continuous path (homotopy) from a known, easily solvable equation to the target equation, progressively approximating the solution to the target equation. Compared to the Gröbner basis method, this approach is particularly suited for complex equation systems that are difficult to solve directly. The utility of homotopy continuation methods has been demonstrated in early works for studying the structure of minimal problems. However, these off-the-shelf solvers are much slower than the current Gröbner basis solvers (by a factor of $10^3 - 10^5$). Nevertheless, recent papers have developed the potential of homotopy continuation methods in solving difficult minimal problems through integration with deep learning, multi-threading, and GPU implementations, making it very suitable for solving generalized relative pose and scale problems.

- **5 and 6 Degrees of Freedom Velocity Estimation with Event Cameras:** The thesis has presented velocity estimation for monocular event cameras based on point and line features. However, the velocity estimation based on point features assumes the Ackermann model, and velocity estimation under the full 5 degrees of freedom scenario still needs to be addressed. Our next step will explore the possibility of extending the point feature-based five-point and eight-point methods, widely used in traditional cameras, to event cameras. Furthermore,

in the discussion of generalized relative pose estimation, especially when compared to monocular cameras, a significant advantage of generalized cameras is their ability to estimate the 6 degrees of freedom of pose, including three translational and three rotational parameters. This is because generalized camera systems usually comprise multiple cameras, providing more information and thus making pose estimation more accurate and reliable. However, when the environment becomes complex, such as in rapidly moving scenes or under severe lighting changes, the pose estimation of traditional camera systems may face challenges. Under these circumstances, the accuracy and robustness of pose estimation may decrease. At this point, developing generalized camera pose estimation algorithms for event cameras becomes particularly important.

A

APPENDIX

A.1 a_{ij} FOR *s5c4* AND *s7c6*

For *s5c4*, we have

$$\begin{aligned}
a_{i1} \approx \widetilde{a}_{i1} =& -x_i\left(\frac{(\omega\Delta t_i)^5}{120} - \frac{(\omega\Delta t_i)^3}{6} + \omega\Delta t_i\right) \\
&+ \frac{(\omega\Delta t_i)^4}{24} - \frac{(\omega\Delta t_i)^2}{2} + 1, \\
a_{i2} \approx \widetilde{a}_{i2} =& -x_i\left(\frac{(\omega\Delta t_i)^4}{24} - \frac{(\omega\Delta t_i)^2}{2} + 1\right) \\
&- \frac{(\omega\Delta t_i)^5}{120} + \frac{(\omega\Delta t_i)^3}{6} - \omega\Delta t_i, \\
a_{i3} \approx \widetilde{a}_{i3} =& \frac{\Delta t_i(x_i\Delta t_i^4\omega^4 - 5\Delta t_i^3\omega^3 - 20x_i\Delta t_i^2\omega^2 + 60\Delta t_i\omega + 120x_i)}{\tau(\tau^4\omega^4 - 20\tau^2\omega^2 + 120)}.
\end{aligned}$$

For *s7c6* we obtain

$$\begin{aligned}
a_{i1} \approx \widetilde{a}_{i1} =& x_i\left(\frac{(\omega\Delta t_i)^7}{5040} - \frac{(\omega\Delta t_i)^5}{120} + \frac{(\omega\Delta t_i)^3}{6} - \omega\Delta t_i\right) \\
&- \frac{(\omega\Delta t_i)^6}{720} + \frac{(\omega\Delta t_i)^4}{24} - \frac{(\omega\Delta t_i)^2}{2} + 1, \\
a_{i2} \approx \widetilde{a}_{i2} =& x_i\left(\frac{(\omega\Delta t_i)^6}{720} - \frac{(\omega\Delta t_i)^4}{24} + \frac{(\omega\Delta t_i)^2}{2} - 1\right) \\
&+ \frac{(\omega\Delta t_i)^7}{5040} - \frac{(\omega\Delta t_i)^5}{120} + \frac{(\omega\Delta t_i)^3}{6} - \omega\Delta t_i, \\
a_{i3} \approx \widetilde{a}_{i3} =& \frac{-(\Delta t_i(-x_i\Delta t_i^6\omega^6 + 7\Delta t_i^5\omega^5 + 42x_i\Delta t_i^4\omega^4 - 210\Delta t_i^3\omega^3}{\tau(\tau^6\omega^6 - 42\tau^4\omega^4 + 840\tau^2\omega^2 - 5040)} \\
&\frac{-(\Delta t_i(-840x_i\Delta t_i^2\omega^2 + 2520\Delta t_i\omega + 5040x_i))}{\tau(\tau^6\omega^6 - 42\tau^4\omega^4 + 840\tau^2\omega^2 - 5040)}.
\end{aligned}$$

BIBLIOGRAPHY

1. Clipp, B., Kim, J.-H., Frahm, J.-M., Pollefeys, M. & Hartley, R. *Robust 6DOF motion estimation for non-overlapping, multi-camera systems* in *IEEE Workshop on Applications of Computer Vision* (2008), 1.
2. Kazik, T., Kneip, L., Nikolic, J., Pollefeys, M. & Siegwart, R. *Real-time 6D stereo visual odometry with non-overlapping fields of view* in *IEEE Conference on Computer Vision and Pattern Recognition* (2012), 1529.
3. Lee, G. H., Faundorfer, F. & Pollefeys, M. *Motion estimation for self-driving cars with a generalized camera* in *IEEE Conference on Computer Vision and Pattern Recognition* (2013), 2746.
4. Migita, T. & Shakunaga, T. *Evaluation of epipole estimation methods with/without rank-2 constraint across algebraic/geometric error functions* in *IEEE Conference on Computer Vision and Pattern Recognition* (2007), 1.
5. Hartley, R. *Minimizing Algebraic Error in Geometric Estimation Problem* in *IEEE International Conference on Computer Vision* (1998), 469.
6. Kneip, L. & Lynen, S. *Direct optimization of frame-to-frame rotation* in *IEEE International Conference on Computer Vision* (2013), 2352.
7. Briales, J., Kneip, L. & Gonzalez-Jimenez, J. *A Certifiably Globally Optimal Solution to the Non-Minimal Relative Pose Problem* in *IEEE Conference on Computer Vision and Pattern Recognition* (2018), 145.
8. Mevissen, M. & Kojima, M. SDP relaxations for quadratic optimization problems derived from polynomial optimization problems. *Asia-Pacific Journal of Operational Research* **27**, 15 (2010).
9. Kahl, F. & Henrion, D. Globally Optimal Estimates for Geometric Reconstruction Problems. *International Journal of Computer Vision* **74**, 3 (2007).
10. Zhao, J. An Efficient Solution to Non-Minimal Case Essential Matrix Estimation. *arXiv:1903.09067* (2019).
11. Bentolila, J. & Francos, J. M. Conic epipolar constraints from affine correspondences. *Computer Vision and Image Understanding* **122**, 105 (2014).

12. Raposo, C. & Barreto, J. P. *Theory and practice of structure-from-motion using affine correspondences* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), 5470.

13. Barath, D. & Hajder, L. Efficient recovery of essential matrix from two affine correspondences. *IEEE Transactions on Image Processing* **27**, 5328 (2018).

14. Eichhardt, I. & Chetverikov, D. *Affine correspondences between central cameras for rapid relative pose estimation* in *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), 482.

15. Barath, D., Polic, M., Förstner, W., Sattler, T., Pajdla, T. & Kukelova, Z. *Making affine correspondences work in camera geometry computation* in *European Conference on Computer Vision* (2020), 723.

16. Guan, B., Zhao, J., Li, Z., Sun, F. & Fraundorfer, F. *Minimal solutions for relative pose with a single affine correspondence* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), 1929.

17. Hajder, L. & Barath, D. *Relative planar motion for vehicle-mounted cameras from a single affine correspondence* in *2020 IEEE International Conference on Robotics and Automation (ICRA)* (2020), 8651.

18. Eichhardt, I. & Barath, D. *Relative pose from deep learned depth and a single affine correspondence* in *European Conference on Computer Vision* (2020), 627.

19. Stewénius, H., Nistér, D., Oskarsson, M. & Aström, K. *Solutions to minimal generalized relative pose problems* in *Workshop on omnidirectional vision* **1** (2005), 3.

20. Kim, J. H., Hartley, R., Frahm, J. M. & Pollefeys, M. *Visual Odometry for Non-overlapping Views Using Second-Order Cone Programming* in *Asian Conference on Computer Vision* (2007), 353.

21. Kim, J. H., Li, H. & Hartley, R. Motion Estimation for Nonoverlapping Multicamera Rigs: Linear Algebraic and L_∞ Geometric Solutions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**, 1044 (2010).

22. Lim, J., Barnes, N. & Li, H. Estimating Relative Camera Motion from the Antipodal-Epipolar Constraint. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**, 1907 (2010).

23. Lee, G. H., Pollefeys, M. & Fraundorfer, F. *Relative pose estimation for a multi-camera system with known vertical direction* in *IEEE Conference on Computer Vision and Pattern Recognition* (2014), 540.

24. Liu, L., Li, H., Dai, Y. & Pan, Q. Robust and efficient relative pose with a multi-camera system for autonomous driving in highly dynamic environments. *IEEE Transactions on Intelligent Transportation Systems* **19**, 2432 (2018).

25. Kneip, L. & Li, H. *Efficient computation of relative pose for multi-camera systems* in *IEEE Conference on Computer Vision and Pattern Recognition* (2014), 446.

26. Campos, J., Cardoso, J. R. & Miraldo, P. *POSEAMM: A Unified Framework for Solving Pose Problems using an Alternating Minimization Method* in *IEEE International Conference on Robotics and Automation* (2019), 3493.

27. Alyousefi, K. & Ventura, J. *Multi-camera motion estimation with affine correspondences* in *International Conference on Image Analysis and Recognition* (2020), 417.

28. Li, H., Hartley, R. & Kim, J.-h. *A Linear Approach to Motion Estimation using Generalized Camera Models* in *IEEE Conference on Computer Vision and Pattern Recognition* (2008), 1.

29. Guan, B., Zhao, J., Barath, D. & Fraundorfer, F. *Efficient recovery of multi-camera motion from two affine correspondences* in *2021 IEEE International Conference on Robotics and Automation (ICRA)* (2021), 1305.

30. Ventura, J., Arth, C. & Lepetit, V. *An efficient minimal solution for multi-camera motion* in *Proceedings of the IEEE International Conference on Computer Vision* (2015), 747.

31. Guan, B., Zhao, J., Barath, D. & Fraundorfer, F. *Minimal Cases for Computing the Generalized Relative Pose using Affine Correspondences* in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), 6068.

32. Zhao, J. & Guan, B. On Relative Pose Recovery for Multi-Camera Systems. *arXiv preprint arXiv:2102.11996* (2021).

33. Sweeney, C., Kneip, L., Höllerer, T. & Turk, M. *Computing Similarity Transformations from Only Image Correspondences* in *IEEE Conference on Computer Vision and Pattern Recognition* (2015), 3305.

34. Kneip, L., Sweeney, C. & Hartley, R. *The generalized relative pose and scale problem: View-graph fusion via 2D-2D registration* in *IEEE Winter Conference on Applications of Computer Vision* (2016), 1.

35. Buchberger, B. *Multidimensional Systems Theory - Progress, Directions and Open Problems in Multidimensional Systems* 184 (Reidel Publishing Company, Dodrecht - Boston - Lancaster, 1985).

36. Cox, D., Little, J., O'Shea, D. & Sweedler, M. Ideals, varieties, and algorithms. *American Mathematical Monthly* **101**, 582 (1994).

37. Cox, D., Little, J. & O'Shea, D. *Using algebraic geometry* (Springer Science & Business Media, 2006).

38. Stewénius, H. & Nistér, D. *Solutions to Minimal Generalized Relative Pose Problems* in *Workshop on Omnidirectional Vision (ICCV)* (Beijing, China, 2005).

39. Kukelova, Z., Bujnak, M. & Pajdla, T. *Automatic generator of minimal problem solvers* in *Proceedings of the European Conference on Computer Vision (ECCV)* (2008), 302.

40. Zheng, Y., Kuang, Y., Sugimoto, S., Astrom, K. & Okutomi, M. *Revisiting the PnP Problem: A Fast, General and Optimal Solution* in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (2013).

41. Kneip, L., Li, H. & Seo, Y. *Upnp: An optimal o (n) solution to the absolute pose problem with universal applicability* in *Proceedings of the European Conference on Computer Vision (ECCV)* (2014), 127.

42. L. Kneip, R. Siegwart & M. Pollefeys. *Finding the Exact Rotation Between Two Images Independently of the Translation* in *Proceedings of the European Conference on Computer Vision (ECCV)* (Firenze, Italy, 2012).

43. Kukelova, Z., Bujnak, M. & Pajdla, T. *Real-time solution to the absolute pose problem with unknown radial distortion and focal length* in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (2013).

44. Lee, G., Pollefeys, M. & Fraundorfer, F. *Relative Pose Estimation for a Multi-Camera System with Known Vertical* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2014).

45. Zhao, J., Kneip, L., He, Y. & Ma, J. Minimal Case Relative Pose Computation using Ray-Point-Ray Features. *Transactions of Pattern Analysis and Machine Intelligence* (2018).

46. Bujnak, M., Kukelova, Z. & Pajdla, T. *Making minimal solvers fast* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2012).

47. Faugère, J., Gianni, P., Lazard, D. & Mora, T. Efficient Computation of Zero-dimensional Gröbner Bases by Change of Ordering. *Journal of Symbolic Computation* **16**, 329 (1993).

48. Kukelova, Z., Bujnak, M., Heller, J. & Pajdla, T. *Singly-bordered block-diagonal form for minimal problem solvers* in *Proceedings of the Asian Conference on Computer Vision (ACCV)* (2014).

49. Ask, E., Yubin, K. & Astrom, K. *Exploiting p-fold symmetries for faster polynomial equation solving* in *Proceedings of the International Conference on Pattern Recognition (ICPR)* (2012).

50. Larsson, V., Astrom, K. & Oskarsson, M. *Polynomial Solvers for Saturated Ideals* in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (2017).

51. Larsson, V., Astrom, K. & Oskarsson, M. *Efficient solvers for minimal problems by syzygy-based reduction* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 820.

52. Larsson, V., Oskarsson, M., Astrom, K., Wallis, A., Kukelova, Z. & Pajdla, T. *Beyond Gröbner bases: Basis selection for minimal solvers* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), 3945.

53. Byröd, M., Josephson, K. & Astrom, K. Fast and Stable Polynomial Equation Solving and Its Application to Computer Vision. *International Journal of Computer Vision* **84**, 237 (2009).

54. Santa Cruz, R., Fernando, B., Cherian, A. & Gould, S. Visual permutation learning. *IEEE transactions on pattern analysis and machine intelligence* **41**, 3100 (2018).

55. Klein, G. & Murray, D. *Parallel tracking and mapping for small AR workspaces* in *2007 6th IEEE and ACM international symposium on mixed and augmented reality* (2007), 225.

56. Mur-Artal, R., Montiel, J. M. M. & Tardos, J. D. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE transactions on robotics* **31**, 1147 (2015).

57. Mur-Artal, R. & Tardós, J. D. ORB-SLAM2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics* **33**, 1255 (2017).

58. Campos, C., Elvira, R., Rodríguez, J. J. G., Montiel, J. M. & Tardós, J. D. ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM. *arXiv preprint arXiv:2007.11898* (2020).

59. Qin, T., Li, P. & Shen, S. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Transactions on Robotics (T-RO)* **34**, 1004 (2018).

60. Engel, J., Schöps, T. & Cremers, D. *LSD-SLAM: Large-Scale Direct Monocular SLAM* in *Proceedings of the European Conference on Computer Vision (ECCV)* (2014).

61. Min, Z. & Dunn, E. *VOLDOR+SLAM: For the times when feature-based or direct methods are not good enough* in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2021).

62. Koestler, L., Yang, N., Zeller, N. & Cremers, D. *TANDEM: Tracking and Dense Mapping in Real-time using Deep Multi-view Stereo* in *In Conference on Robot Learning (CoRL)* (2021).

63. Teed, Z. & Deng, J. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. *Advances in neural information processing systems* (2021).

64. Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A. J., Conradt, J., Daniilidis, K., *et al.* Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence* **44**, 154 (2020).

65. 2022.

66. Longuet-Higgins, H. C. A computer algorithm for reconstructing a scene from two projections. *Nature* **293**, 133 (1981).

67. Hartley, R. & Zisserman, A. *Multiple View Geometry in Computer Vision* (Cambridge University Press, 2003).

68. Pizarro, O., Eustice, R. M. & Singh, H. *Relative Pose Estimation for Instrumented, Calibrated Imaging Platforms.* in *DICTA* (2003), 601.

69. Nistér, D. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**, 756 (2004).

70. Stewénius, H., Engels, C. & Nistér, D. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing* **60**, 284 (2006).

71. Fraundorfer, F., Tanskanen, P. & Pollefeys, M. *A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles* in *Proceedings of the European Conference on Computer Vision (ECCV)* (2010).

72. Guerrero, J., Murillo, A. & Sagues, C. Localization and matching using the planar trifocal tensor with bearing-only data. **24**, 494 (2008).

73. Scaramuzza, D., Fraundorfer, F. & Siegwart, R. *Real-time monocular visual odometry for on-road vehicles with 1-point ransac* in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2009), 4293.

74. Scaramuzza, D., Fraundorfer, F., Pollefeys, M. & Siegwart, R. *Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints* in *2009 IEEE 12th international conference on computer vision* (2009), 1413.

75. Scaramuzza, D. 1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. *International Journal of Computer Vision (IJCV)* **95**, 74 (2011).

76. Hee Lee, G., Faundorfer, F. & Pollefeys, M. *Motion estimation for self-driving cars with a generalized camera* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013), 2746.

77. Peng, X., Cui, J. & Kneip, L. *Articulated multi-perspective cameras and their application to truck motion estimation* in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2019), 2052.

78. Huang, K., Wang, Y. & Kneip, L. *Motion estimation of non-holonomic ground vehicles from a single feature correspondence measured over n views* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Long Beach, USA, 2019).

79. Hartley, R. *Projective reconstruction from line correspondences* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (1994).

80. Huang, K., Wang, Y. & Kneip, L. *B-splines for Purely Vision-based Localization and Mapping on Non-holonomic Ground Vehicles* in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2021).

81. Weikersdorfer, D., Hoffmann, R. & Conradt, J. *Simultaneous localization and mapping for event-based vision systems* in *International Conference on Computer Vision Systems* (2013), 133.

82. Weikersdorfer, D., Adrian, D. B., Cremers, D. & Conradt, J. *Event-based 3D SLAM with a depth-augmented dynamic vision sensor* in *2014 IEEE international conference on robotics and automation (ICRA)* (2014), 359.

83. Censi, A. & Scaramuzza, D. *Low-latency event-based visual odometry* in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (2014), 703.

84. Mueggler, E., Huber, B. & Scaramuzza, D. *Event-based, 6-DOF pose tracking for high-speed maneuvers* in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2014), 2761.

85. Gallego, G., Lund, J. E., Mueggler, E., Rebecq, H., Delbruck, T. & Scaramuzza, D. Event-based, 6-dof camera tracking for high-speed applications. *arXiv preprint arXiv:1607.03468* **2** (2016).

86. Chamorro Hernández, W. O., Andrade-Cetto, J. & Solà Ortega, J. *High-speed event camera tracking* in *Proceedings of the The 31st British Machine Vision Virtual Conference* (2020), 1.

87. Bryner, S., Gallego, G., Rebecq, H. & Scaramuzza, D. *Event-based, direct camera tracking from a photometric 3D map using nonlinear optimization* in *2019 International Conference on Robotics and Automation (ICRA)* (2019), 325.

88. Gallego, G., Rebecq, H. & Scaramuzza, D. *A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), 3867.

89. Gallego, G., Gehrig, M. & Scaramuzza, D. *Focus is all you need: Loss functions for event-based vision* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), 12280.

90. Kueng, B., Mueggler, E., Gallego, G. & Scaramuzza, D. *Low-latency visual odometry using event-based feature tracks* in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2016), 16.

91. Liu, D., Parra, A. & Chin, T.-J. *Globally optimal contrast maximisation for event-based motion estimation* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), 6349.

92. Peng, X., Gao, L., Wang, Y. & Kneip, L. Globally-optimal contrast maximisation for event cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

93. Kim, H., Leutenegger, S. & Davison, A. J. *Real-time 3D reconstruction and 6-DoF tracking with an event camera* in *European conference on computer vision* (2016), 349.

94. Rebecq, H., Horstschäfer, T., Gallego, G. & Scaramuzza, D. Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time. *IEEE Robotics and Automation Letters* **2**, 593 (2016).

95. Zhu, A., Atanasov, N. & Daniilidis, K. *Event-based visual inertial odometry* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).

96. Rebecq, H., Horstschaefer, T. & Scaramuzza, D. *Real-time Visual-Inertial Odometry for Event Cameras using Keyframe-based Nonlinear Optimization* in *Proceedings of the British Machine Vision Conference (BMVC)* (2017).

97. Mueggler, E., Gallego, G., Rebecq, H. & Scaramuzza, D. Continuous-time visual-inertial odometry for event cameras. *IEEE Transactions on Robotics* **34**, 1425 (2018).

98. Vidal, A. R., Rebecq, H., Horstschaefer, T. & Scaramuzza, D. Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios. *IEEE Robotics and Automation Letters* **3**, 994 (2018).

99. Zhou, Y., Gallego, G. & Shen, S. Event-based stereo visual odometry. *IEEE Transactions on Robotics* **37**, 1433 (2021).

100. Zuo, Y.-F., Yang, J., Chen, J., Wang, X., Wang, Y. & Kneip, L. DEVO: Depth-Event Camera Visual Odometry in Challenging Conditions. *arXiv preprint arXiv:2202.02556* (2022).

101. Zhou, Y., Li, H. & Kneip, L. Canny-vo: Visual odometry with rgb-d cameras based on geometric 3-d–2-d edge alignment. *IEEE Transactions on Robotics* **35**, 184 (2018).

102. Hidalgo-Carrió, J., Gallego, G. & Scaramuzza, D. *Event-aided Direct Sparse Odometry* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), 5781.

103. Le Gentil, C., Tschopp, F., Alzugaray, I., Vidal-Calleja, T., Siegwart, R. & Nieto, J. *IDOL: A framework for IMU-DVS odometry using lines* in *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)* (2020), 5863.

104. Zihao Zhu, A., Atanasov, N. & Daniilidis, K. *Event-based visual inertial odometry* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), 5391.

105. Maqueda, A. I., Loquercio, A., Gallego, G., García, N. & Scaramuzza, D. *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), 5419.

106. Gehrig, M., Shrestha, S. B., Mouritzen, D. & Scaramuzza, D. *Event-based angular velocity regression with spiking networks* in *2020 IEEE International Conference on Robotics and Automation (ICRA)* (2020), 4195.

107. Peng, X., Xu, W., Yang, J. & Kneip, L. *Continuous Event-Line Constraint for Closed-Form Velocity Initialization* in *Proceedings of the British Machine Vision Conference (BMVC)* (2021).

108. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I. & Leonard, J. J. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Transactions on Robotics (T-RO)* **32** (2016).

109. Sterlow, D. & Singh, S. Motion estimation from image and intertial measurements. *International Journal of Robotics Research (IJRR)* **23**, 1157 (2004).

110. Dong-Si, T.-C. & Mourikis, A. I. *Motion tracking with fixed-lag smoothing: Algorithm and consistency analysis* in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2011).

111. Mourikis, A. I., Roumeliotis, S. I., *et al. A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation.* in *ICRA* **2** (2007), 6.

112. Strasdat, H., Montiel, J. M. M. & Davison, A. J. *Realtime monocular SLAM: Why filter?* in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2010).

113. Lupton, T. & Sukkarieh, S. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. *IEEE Transactions on Robotics* **28**, 61 (2011).

114. Forster, C., Carlone, L., Dellaert, F. & Scaramuzza, D. On-Manifold Preintegration for Real-Time Visual-Inertial Odometry. *IEEE Transactions on Robotics (T-RO)* **33**, 1 (2017).

115. Campos, C., Elvira, R., Rodríguez, J. J. G., Montiel, J. M. & Tardós, J. D. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics* **37**, 1874 (2021).

116. Qin, T., Li, P. & Shen, S. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics* **34**, 1004 (2018).

117. Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R. & Furgale, P. Keyframe-based visual–inertial odometry using nonlinear optimization. *International Journal of Robotics Research (IJRR)* **34**, 314 (2015).

118. Von Stumberg, L. & Cremers, D. *DM-VIO: Delayed Marginalization Visual-Inertial Odometry* in. **7** (2022), 1408.

119. Song, X., Seneviratne, L. D., Althoefer, K. & Song, Z. Vision-based velocity estimation for unmanned ground vehicles. *International Journal of Information Acquisition* **4**, 303 (2007).

120. Honegger, D., Greisen, P., Meier, L., Tanskanen, P. & Pollefeys, M. *Real-time velocity estimation based on optical flow and disparity matching* in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2012), 5177.

121. Honegger, D., Meier, L., Tanskanen, P. & Pollefeys, M. *An open source and open hardware embedded metric optical flow cmos camera for indoor and outdoor applications* in *2013 IEEE International Conference on Robotics and Automation* (2013), 1736.

122. Weiss, S., Achtelik, M. W., Lynen, S., Chli, M. & Siegwart, R. *Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments* in *2012 IEEE international conference on robotics and automation* (2012), 957.

123. Weiss, S., Brockers, R. & Matthies, L. *4dof drift free navigation using inertial cues and optical flow* in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2013), 4180.

124. Ma, Y., Soatto, S., Košecká, J. & Sastry, S. *An invitation to 3-d vision: from images to geometric models* (Springer, 2004).

125. Deng, H., Arif, U., Fu, Q., Xi, Z., Quan, Q. & Cai, K.-Y. Visual–inertial estimation of velocity for multicopters based on vision motion constraint. *Robotics and Autonomous Systems* **107**, 262 (2018).

126. Gao, Z., Ramesh, B., Lin, W.-Y., Wang, P., Yan, X. & Zhai, R. Efficient velocity estimation for MAVs by fusing motion from two frontally parallel cameras. *Journal of Real-Time Image Processing* **16**, 2367 (2019).

127. Weng, J., Huang, T. S. & Ahuja, N. Motion and structure from line correspondences; closed-form solution, uniqueness, and optimization. *IEEE Transactions on Pattern Analysis & Machine Intelligence* **14**, 318 (1992).

128. Hartley, R. I. Lines and points in three views and the trifocal tensor. *International Journal of Computer Vision* **22**, 125 (1997).

129. Von Gioi, R. G., Jakubowicz, J., Morel, J.-M. & Randall, G. LSD: A fast line segment detector with a false detection control. *IEEE transactions on pattern analysis and machine intelligence* **32**, 722 (2008).

130. Pumarola, A., Vakhitov, A., Agudo, A., Sanfeliu, A. & Moreno-Noguer, F. *PL-SLAM: Real-time monocular visual SLAM with points and lines* in *2017 IEEE international conference on robotics and automation (ICRA)* (2017), 4503.

131. He, Y., Zhao, J., Guo, Y., He, W. & Yuan, K. PL-VIO: Tightly-coupled monocular visual–inertial odometry using point and line features. *Sensors* **18**, 1159 (2018).

132. Pless, R. *Using many cameras as one* in *IEEE Conference on Computer Vision and Pattern Recognition* (2003).

133. Helmke, U., Hüper, K., Lee, P. Y. & Moore, J. Essential matrix estimation using Gauss-Newton iterations on a manifold. *International Journal of Computer Vision* **74**, 117 (2007).

134. Kneip, L., Siegwart, R. & Pollefeys, M. *Finding the exact rotation between two images independently of the translation* in *European Conference on Computer Vision* (Springer, 2012), 696.

135. Briales, J. & Gonzalez-Jimenez, J. *Convex global 3D registration with Lagrangian duality* in *IEEE Conference on Computer Vision and Pattern Recognition* (2017), 4960.

136. Anstreicher, K. & Wolkowicz, H. On Lagrangian relaxation of quadratic matrix constraints. *SIAM Journal on Matrix Analysis and Applications* **22**, 41 (2000).

137. Yang, H. & Carlone, L. *A Quaternion-based Certifiably Optimal Solution to the Wahba Problem with Outliers* in *International Conference on Computer Vision* (2019), 1665.

138. Luo, Z.-Q., Ma, W.-K., So, A. M.-C., Ye, Y. & Zhang, S. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine* **27**, 20 (2010).

139. Vandenberghe, L. & Boyd, S. Semidefinite programming. *SIAM Review* **38**, 49 (1996).

140. Ye, Y. *Interior Point Algorithms: Theory and Analysis* (Wiley & Sons, 1997).

141. Boyd, S. & Vandenberghe, L. *Convex Optimization* (Cambridge University Press, 2004).

142. Sanyal, R., Sottile, F. & Sturmfels, B. Orbitopes. *Mathematika* **57**, 275 (2011).

143. Cifuentes, D., Agarwal, S., Parrilo, P. A. & Thomas, R. R. On the local stability of semidefinite relaxations. *arXiv:1710.04287v2* (2018).

144. Yamashita, M., Fujisawa, K., Fukuda, M., Kobayashi, K., Nakata, K. & Nakata, M. in *Handbook on semidefinite, conic and polynomial optimization* 687 (Springer, 2012).

145. Park, J. & Boyd, S. General Heuristics for Nonconvex Quadratically Constrained Quadratic Programming. *arXiv preprint arXiv:1703.07870* (2017).

146. Kneip, L. & Furgale, P. *OpenGV: A unified and generalized approach to real-time calibrated geometric vision* in *IEEE International Conference on Robotics and Automation* (2014), 1.

147. Fischler, M. A. & Bolles, R. C. Random Sample Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography. *Communications of the ACM* **24**, 381 (1981).

148. Geiger, A., Lenz, P. & Urtasun, R. *Are we ready for autonomous driving? the KITTI vision benchmark suite* in *IEEE Conference on Computer Vision and Pattern Recognition* (2012), 3354.

149. Schonberger, J. L. & Frahm, J.-M. *Structure-from-motion revisited* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), 4104.

150. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I. & Leonard, J. J. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics* **32**, 1309 (2016).

151. Kneip, L., Scaramuzza, D. & Siegwart, R. *A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation* in *CVPR 2011* (2011), 2969.

152. Zhao, J., Xu, W. & Kneip, L. *A certifiably globally optimal solution to generalized essential matrix estimation* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), 12034.

153. Heng, L., Choi, B., Cui, Z., Geppert, M., Hu, S., Kuan, B., Liu, P., Nguyen, R., Yeo, Y. C., Geiger, A., *et al. Project autovision: Localization and 3d scene perception for an autonomous vehicle with a multi-camera system* in *2019 International Conference on Robotics and Automation (ICRA)* (2019), 4695.

154. Wang, Y., Huang, K., Peng, X., Li, H. & Kneip, L. *Reliable frame-to-frame motion estimation for vehicle-mounted surround-view camera systems* in *2020 IEEE International conference on robotics and automation (ICRA)* (2020), 1660.

155. Snavely, N., Seitz, S. M. & Szeliski, R. in *ACM siggraph 2006 papers* 835 (2006).

156. Frahm, J.-M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y.-H., Dunn, E., Clipp, B., Lazebnik, S., *et al. Building rome on a cloudless day* in *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11* (2010), 368.

157. Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S. M. & Szeliski, R. Building rome in a day. *Communications of the ACM* **54**, 105 (2011).

158. Moulon, P., Monasse, P., Perrot, R. & Marlet, R. *Openmvg: Open multiple view geometry* in *Reproducible Research in Pattern Recognition: First International Workshop, RRPR 2016, Cancún, Mexico, December 4, 2016, Revised Selected Papers 1* (2017), 60.

159. Konolige, K. & Agrawal, M. FrameSLAM: From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics* **24**, 1066 (2008).

160. Chum, O., Matas, J. & Kittler, J. *Locally optimized RANSAC* in *Pattern Recognitio* (2003), 236.

161. Morel, J.-M. & Yu, G. ASIFT: A new framework for fully affine invariant image comparison. *SIAM journal on imaging sciences* **2**, 438 (2009).

162. Mishkin, D., Matas, J. & Perdoch, M. MODS: Fast and robust method for two-view matching. *Computer vision and image understanding* **141**, 81 (2015).

163. Mishchuk, A., Mishkin, D., Radenovic, F. & Matas, J. Working hard to know your neighbor's margins: Local descriptor learning loss. *Advances in neural information processing systems* **30** (2017).

164. Mishkin, D., Radenovic, F. & Matas, J. *Repeatability is not enough: Learning affine regions via discriminability* in *Proceedings of the European Conference on Computer Vision (ECCV)* (2018), 284.

165. Fischler, M. A. & Bolles, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* **24**, 381 (1981).

166. Molnár, J. & Chetverikov, D. Quadratic Transformation for Planar Mapping of Implicit Surfaces. *Journal of Mathematical Imaging and Vision* (2014).

167. Barath, D. & Matas, J. *Graph-cut RANSAC* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), 6733.

168. Barath, D., Noskova, J., Ivashechkin, M. & Matas, J. *MAGSAC++, a fast, reliable and accurate robust estimator* in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), 1304.

169. Barath, D., Pritts, J., Mishkin, D. & Hajder, L. *Affine Correspondences and their Applications* in *Conference on Computer Vision and Pattern Recognition Tutorial* (2022).

170. Li, H. & Hartley, R. *Five-Point Motion Estimation Made Easy* in *International Conference on Pattern Recognition* (2006), 630.

171. Stewénius, H., Engels, C. & Nistér, D. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing* **60**, 284 (2006).

172. Kukelova, Z., Bujnak, M. & Pajdla, T. *Polynomial Eigenvalue Solutions to the 5-pt and 6-pt Relative Pose Problems.* in *BMVC* **2** (2008), 2008.

173. Ventura, J., Kukelova, Z., Sattler, T. & Baráth, D. *P1AC: Revisiting absolute pose from a single affine correspondence* in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), 19751.

174. Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M. W. & Siegwart, R. The EuRoC micro aerial vehicle datasets. *International Journal of Robotics Research* **35**, 1157 (2016).

175. Sturm, J., Engelhard, N., Endres, F., Burgard, W. & Cremers, D. *A Benchmark for the Evaluation of RGB-D SLAM Systems* in *Proc. of the International Conference on Intelligent Robot Systems (IROS)* (2012).

176. He, K., Zhang, X., Ren, S. & Sun, J. *Deep residual learning for image recognition* in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), 770.

177. Mora, T. & Robbiano, L. The Gröbner Fan of an Ideal. *Journal of Symbolic Computation* **6**, 183 (1988).

178. Tsakiris, M. C., Peng, L., Conca, A., Kneip, L., Shi, Y. & Choi, H. An Algebraic-Geometric Approach to Shuffled Linear Regression. *ArXiv e-prints* (2018).

179. Peng, L., Song, X., Tsakiris, M. C., Choi, H., Kneip, L. & Shi, Y. *Algebraically-Initialized Expectation Maximization for Header-Free Communication* in *Proceedings of the IEEE International Conference on Accoustics, Speech and Signal Processing (ICASSP)* (2019).

180. Scaramuzza, D. & Fraundorfer, F. Visual odometry [tutorial]. *IEEE robotics & automation magazine* **18**, 80 (2011).

181. Nistér, D., Naroditsky, O. & Bergen, J. *Visual odometry* in *IEEE Conference on Computer Vision and Pattern Recognition* (2004).

182. Klein, G. & Murray, D. *Parallel tracking and mapping on a camera phone* in *2009 8th IEEE International Symposium on Mixed and Augmented Reality* (2009), 83.

183. Forster, C., Pizzoli, M. & Scaramuzza, D. *SVO: Fast Semi-Direct Monocular Visual Odometry* in *IEEE International Conference on Robotics and Automation (ICRA)* (2014).

184. Engel, J., Koltun, V. & Cremers, D. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence* **40**, 611 (2017).

185. Xu, W., Peng, X. & Kneip, L. Tight Fusion of Events and Inertial Measurements for Direct Velocity Estimation. *IEEE Transactions on Robotics*, 1 (2023).

186. Geiger, A., Lenz, P., Stiller, C. & Urtasun, R. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research* **32**, 1231 (2013).

187. Gehrig, D., Gehrig, M., Hidalgo-Carrió, J. & Scaramuzza, D. *Video to events: Recycling video datasets for event cameras* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), 3586.

188. Alzugaray, I. & Chli, M. Asynchronous corner detection and tracking for event cameras in real time. *IEEE Robotics and Automation Letters* **3**, 3177 (2018).

189. Shan, T. & Englot, B. *LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain* in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2018), 4758.

190. Brändli, C., Strubel, J., Keller, S., Scaramuzza, D. & Delbruck, T. *ELiSeD—An event-based line segment detector* in *2016 Second International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)* (2016), 1.

191. Bartoli, A. & Sturm, P. Structure-from-motion using lines: Representation, triangulation, and bundle adjustment. *Computer Vision and Image Understanding (CVIU)* **100**, 416 (2005).

192. Zhang, G., Lee, J. H., Lim, J. & Suh, I. H. Building a 3-D line-based map using stereo SLAM. *IEEE Transactions on Robotics (T-RO)* **31**, 1364 (2015).

193. Bartoli, A. & Sturm, P. *The 3D line motion matrix and alignment of line reconstructions* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* **1** (2001), I.

194. Stoffregen, T. & Kleeman, L. *Event cameras, contrast maximization and reward functions: an analysis* in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), 12300.

195. Cox, D., Little, J. & OShea, D. *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra* (Springer Science & Business Media, 2013).

196. Huber, P. J. in *Breakthroughs in statistics* 492 (Springer, 1992).

197. Agarwal, S., Mierle, K. & Team, T. C. S. *Ceres Solver* version 2.1. 2022.

198. Qin, T., Li, P. & Shen, S. VINS-MONO: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics (T-RO)* **34**, 1004 (2018).

199. Yang, J., Li, H., Campbell, D. & Jia, Y. Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE transactions on pattern analysis and machine intelligence* **38**, 2241 (2015).

200. Delmerico, J., Cieslewski, T., Rebecq, H., Faessler, M. & Scaramuzza, D. *Are we ready for autonomous drone racing? the UZH-FPV drone racing dataset* in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2019), 6713.

201. Rosten, E. & Drummond, T. *Machine learning for high-speed corner detection* in *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9* (2006), 430.

202. Lucas, B. D. & Kanade, T. *An iterative image registration technique with an application to stereo vision* in *IJCAI'81: 7th international joint conference on Artificial intelligence* **2** (1981), 674.

CURRICULUM VITAE

PERSONAL DATA

Name	Wanting Xu
Date of Birth	July 10, 1997
Place of Birth	Shandong, China
Citizen of	China

EDUCATION

2018 – 2024	ShanghaiTech University, Shanghai, China *Final degree:* Doctorate
2014 – 2018	Xinjiang University Xinjiang, China *Final degree:* Bachelor

PUBLICATIONS

Articles in peer-reviewed journals:

1. Xu, W., Peng, X. & Kneip, L. Tight Fusion of Events and Inertial Measurements for Direct Velocity Estimation. *IEEE Transactions on Robotics*, 1 (2023).
4. Zuo, Y.-.-F., Xu, W., Wang, X., Wang, Y. & Kneip, L. Cross-Modal Semi-Dense 6-DoF Tracking of an Event Camera in Challenging Conditions. *IEEE Transactions on Robotics*, 1 (2024).
7. Zhang, X., Peng, L., Xu, W. & Kneip, L. Accelerating Globally Optimal Consensus Maximization in Geometric Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

Conference contributions:

2. Xu, W., Hu, L., Tsakiris, M. C. & Kneip, L. *Online stability improvement of Gröbner basis solvers using deep learning* in *2019 International Conference on 3D Vision (3DV)* (2019), 544.
3. Xu, W., Zhang, S., Cui, L., Peng, X. & Kneip, L. *Event-Based Visual Odometry on Non-Holonomic Ground Vehicles* in *2024 International Conference on 3D Vision (3DV)* (2024).
5. Zhao, J., Xu, W. & Kneip, L. *A certifiably globally optimal solution to generalized essential matrix estimation* in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), 12034.
6. Peng, X., Xu, W., Yang, J. & Kneip, L. *Continuous Event-Line Constraint for Closed-Form Velocity Initialization* in *Proceedings of the British Machine Vision Conference (BMVC)* (2021).