Extrinsic and Intrinsic Sensor Calibration

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Faraz M. Mirzaei

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Stergios I. Roumeliotis, Adviser

December, 2013

Acknowledgements

First and foremost, my gratitude goes to Prof. Stergios Roumeliotis, whose constant care and guidance not only enabled me to complete my PhD and this dissertation, but also instilled in me the vision and attention to detail that is expected from an engineer. His stride towards excellence and contributions to the science is nothing but exemplary. I wish him the best in his future academic endeavors.

I would like to thank my first mentor in graduate school and great friend, Tassos Mourikis for taking the time to patiently answer all the questions that I had for him as a junior graduate student, and to teach me the basics of research in grad school. I would also like to thank my friend Nikolas Trawny for his invaluable advice and help. His great attention to detail was always an example for the group. He was always someone that one could rely on in times of need.

I am heavily indebted to Joel Hesch, with whom, I spent endless nights debugging code, writing papers, and having philosophical discussions, and to Gian Luca Mariottini, from whom I learned many lessons in kindness and companionship. I would never forget the cold nights of February, when we were tirelessly working day after night, each one keeping the other awake and motivated.

My further gratitude goes to my friends and former office-mates: Esha Nerurkar, Sam Zhou, Ke Zhou, Paul Huang, Dimitrios Kottas, Chao Guo, and Kejain Wu. Your companionship made the graduate school and the MARS lab a fun place to be, and I will miss you all. I would also like to thank my first roommates, Farshid Rafiee Rad, who helped me settle down in Minnesota, and Ehsan Elahi who proved to me that human kindness and compassion can be truly boundless. My deep gratitude goes to Kia Bazargan, my dearest friend, who helped me out in most difficult moments of my life. His constant support and advice – only one phone call away – is one of the reasons that I completed the doctoral program. Minnesota's cold winters would not have been tolerable for me without having company of my great friends: Shahrouz Takyar, Haleh Hagh-Shenas, Amir Radmehr, Samira Tahvildari, Amir Kashipazha, Shervin Shajiee,

Sharareh Noorbaloochi, Nikolas Gatsis, Navid Dadkhah, and many others with whom I have had the best memories of my life. Thank you.

Above and beyond everything, the unlimited and unconditional love and support that my parents, my sisters, and my brother gave me was the reason that I started graduate school in the first place. While thousands of miles away from me, talking with them, and keeping their thoughts in my heart gave me the ability to accomplish this task. I love you.

I would not have been able to complete this dissertation without constant support and encouragement from the love of my life, my beautiful wife, Mina. From the nights that you stayed up with me, and supported me to complete my research and write my papers, to all the advice you gave me in difficult moments, being there when I needed you, I am forever in your debt. I always knew that someone at home will listen all night to my stories and worries. I am so lucky to have you. I love you.

Dedication

To Mohammad M. and others,
who sacrificed their lives in the pursuit of freedom.
And to their parents,
whose tears will never bring their beloved back.

Abstract

Sensor Calibration is the process of determining the intrinsic (e.g., focal length) and extrinsic (i.e., position and orientation (pose) with respect to the world, or to another sensor) parameters of a sensor. This task is an essential prerequisite for many applications in robotics, computer vision, and augmented reality. For example, in the field of robotics, in order to fuse measurements from different sensors (e.g., camera, LIDAR, gyroscope, accelerometer, odometer, etc. for the purpose of Simultaneous Localization and Mapping or SLAM), all the sensors' measurements must be expressed with respect to a common frame of reference, which requires knowing the relative pose of the sensors. In augmented reality the pose of a sensor (camera in this case) with respect to the surrounding world along with its internal parameters (focal length, principal point, and distortion coefficients) have to be known in order to superimpose an object into the scene.

When designing calibration procedures and before selecting a particular estimation algorithm, there exist two main issues of concern than one needs to consider:

1. Whether the system is observable, meaning that the sensor's measurements contain sufficient information for estimating all degrees of freedom (d.o.f.) of the unknown calibration parameters;

2. Given an observable system, whether it is possible to find the globally optimal solution.

Addressing these issues is particularly challenging due to the nonlinearity of the sensors' measurement models. Specifically, classical methods for analyzing the observability of linear systems (e.g., the observability Gramian) are not directly applicable to nonlinear systems. Therefore, more advanced tools, such as Lie derivatives, must be employed to investigate these systems' observability. Furthermore, providing a guarantee of optimality for estimators applied to nonlinear systems is very difficult, if not impossible. This is due to the fact that commonly used (iterative) linearized estimators require initialization and may only converge to a local optimum. Even with accurate initialization, no guarantee can be made regarding the optimality of the solution computed by linearized estimators.

In this dissertation, we address some of these challenges for several common sensors, including cameras, 3D LIDARs, gyroscopes, Inertial Measurement Units (IMUs), and odometers. Specifically, in the first part of this dissertation we employ Lie-algebra techniques to study the observability of gyroscope-odometer and IMU-camera calibration systems. In addition, we prove the observability of the 3D LIDAR-camera calibration system by demonstrating that only a finite number of values for the calibration parameters produce a given set of measurements. Moreover, we provide the conditions on the control inputs and measurements under which these systems become observable. In the second part of this dissertation, we present a novel method for mitigating the initialization requirements of iterative estimators for the 3D LIDAR-camera and monocular camera calibration systems. Specifically, for each problem we formulate a nonlinear Least-Squares (LS) cost function whose optimality conditions comprise a system of polynomial equations. We subsequently exploit recent advances in algebraic geometry to analytically solve these multivariate polynomial systems and compute the LS critical points. Finally, the guaranteed LS-optimal solutions are directly found by evaluating the cost function at the critical points without requiring any initialization or iteration.

Together, our observability analysis and analytical LS methods provide a framework for accurate and reliable calibration of common sensors in robotics and computer vision.

Contents

List of Tables

List of Figures

Nomenclature and Abbreviations

$\mathbf{I}_n$ The $n \times n$ identity matrix.

$\mathbf{0}_{m\times n}$ The $m \times n$ matrix of zeros.

$\{A\}$ A coordinate frame of reference.

${}^{X}_{Y}\mathbf{C}$ Rotation matrix representing the relative orientation of $\{Y\}$ w.r.t. $\{X\}$.

${}^{X}\mathbf{t}_{Y}$ Relative position of $\{Y\}$ w.r.t. $\{X\}$.

■ End of example.

□ End of proof.

ANLS Analytical Nonlinear Least Squares

BLS Batch Least Squares

CAD Computer Aided Design

d.o.f. degrees of freedom

EKF Extended Kalman Filter

FLS Fixed-Lag Smoothing

f.o.v. field of view

GPS Global Positioning System

IEKF Iterative Extended Kalman Filter

IMU Inertial Measurement Unit

INS Inertial Navigation System

KF Kalman Filter

PBH Popov-Belevitch-Hautus

PF Particle Filter

PnP Perspective n-point Pose

UKF Unscented Kalman Filter

w.r.t. with respect to

Chapter 1

Introduction

1.1 Sensors in Robotics and Computer Vision

In today's world, cameras, laser scanners, gyroscopes, and accelerometers found on vehicles (e.g., cars and airplanes), personal electronic devices (e.g., cell phone and laptops), and robots, are increasingly being used to perform localization (e.g., for personal navigation, providing location-based services, etc.), or to automate tasks that used to be solely executed by humans (e.g., parallel parking, lawn mowing, window cleaning, etc.). These sensors are usually classified into two categories: (i) Proprioceptive sensors, which measure quantities related to their motion, such as linear and angular velocities and accelerations. Examples of this type of sensors are wheel encoders and Inertial Measurement Units (IMUs). (ii) Exteroceptive sensors, which provide information about the environment, such as the distance and bearing to a feature, or directly measure the sensor's position and orientation (pose) with respect to an external frame of reference. Examples of this type of sensors include cameras, laser scanners, Global Positioning System (GPS) receivers, compasses, etc.

In order to effectively use the information provided by one or several sensors onboard a device, a measurement model should be developed that relates the sensor's measurements to the states that need to be estimated (e.g., the position of a vehicle or a map of the area it navigates in). Measurement models often include two sets of parameters that have to be *known*, before the model can be used to process the sensors' measurements. In the following two sections we provide an overview of these two sets of calibration parameters, and then argue why it is essential to determine them accurately.

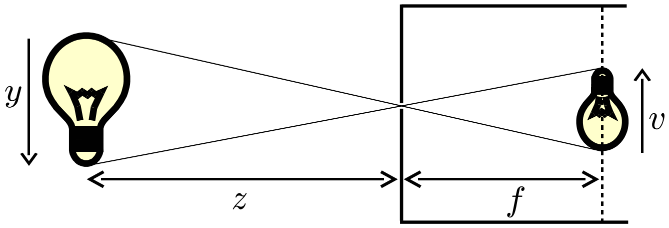


Figure 1.1: Perspective projection camera model: The image plane is shown by a dashed line. For simplicity this figure only shows the y and z coordinates of the model.

1.1.1 Intrinsic Parameters

Intrinsic parameters are those that do not depend on the outside world and how the sensor is placed in it. A well-studied case is the perspective projection camera shown in Fig. 1.1. The pin-hole camera model follows a simple mathematical formulation:

$$u = f\frac{x}{z} \quad , \quad v = f\frac{y}{z} \tag{1.1}$$

In this model, u and v represent the 2D projection of a feature point (e.g., a landmark) on the image plane, x, y, and z represent the 3D position of the corresponding point in the world coordinate frame with origin at the focal point of the camera, and f denotes the focal length of the camera. In this simple model, the focal length of the camera is an internal parameter which is usually unknown or only approximately known. The focal length should be estimated accurately before employing this camera model in any sensor fusion algorithm. This problem, which is called camera *intrinsic calibration*, has received considerable attention in the past and well-established solutions exist in the literature [129, 130, 50, 140]. Similar to a camera, many other sensors such as 2D laser scanners and 3D LIDARs [49, 44], IMUs [123], wheel encoders [39], etc., have internal parameters that must be calibrated before using them. Although the problem of intrinsic calibration is well studied for sensors such as IMUs [122, 57, 58], wheel odometer [1, 6, 14], and 2D laser scanners [136, 104], the development of new sensors such as the revolving-head 3D LIDAR (e.g., Velodyne [133]) whose model comprises hundreds of parameters, has raised the need for new calibration procedures. One of the contributions of this dissertation is introducing a novel algorithm for intrinsic calibration of the revolving-head 3D LIDAR.

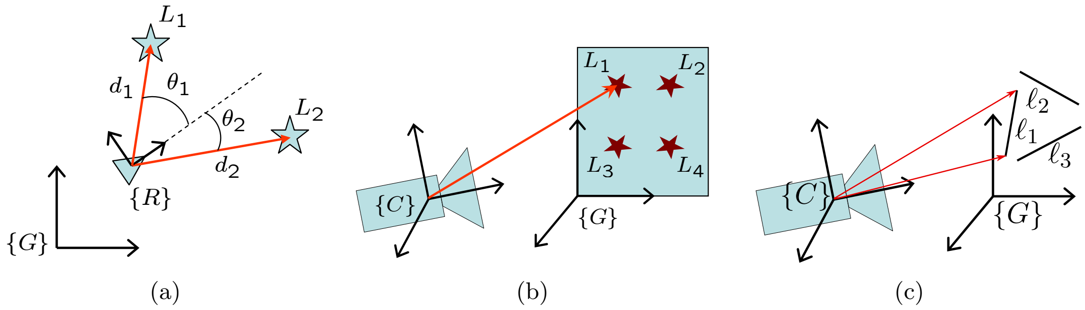


Figure 1.2: (a) A laser scanner can be localized in 2D space (3 d.o.f.), with respect to the global frame of reference, if it can measure its distance and bearing to two or more landmarks whose positions are known; (b) Localization of camera in 3D space (6 d.o.f.) requires observations of at least four non-collinear known landmarks; (c) A camera can also be localized using observations of three known lines whose directions are linearly independent.

1.1.2 Extrinsic Parameters

Extrinsic parameters are those that describe the pose (i.e., position and orientation) of a sensor with respect to an external frame of reference. When the the sensor's pose needs to be determined with respect to a global frame of reference (i.e., a frame not attached to the device or vehicle carrying the sensor), the problem of estimating these parameters is often called *global localization*, and it can be solved using efficient algorithms that exist for various sensors [115, 138, 7]. For example, the 2D pose of a laser scanner with respect to the global frame can be found easily if distance and bearing measurements to at least two *a priori* known landmarks are provided [see Fig. 1.2(a)]. A related, but more challenging problem is that of 3D camera localization, also known as *extrinsic camera calibration* [69, 106, 77, 3, 48]. In this case, the 6 d.o.f. camera pose can be computed from observations of at least four non-collinear landmarks whose positions are known in the global frame of reference [see Fig. 1.2(b)], or at least three known lines whose directions in the 3D space are linearly independent [see Fig. 1.2(c)]. Despite the extensive treatment of this problem, one of the most important aspects of it, i.e., the *optimality* of the solution has not yet been addressed. One of the main contributions of this dissertation is to provide a method for extrinsic calibration of a camera from line observations with guarantees of optimality in a least-squares sense.

Sensor-to-sensor Extrinsic Calibration

In many systems, multiple sensors are rigidly attached to the same device. Fusing measurements from multiple sensors may be necessary in order to ensure that the system is observable, or to increase robustness against single-sensor failure. The quantities that

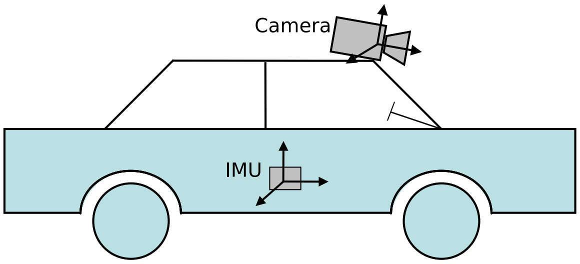


Figure 1.3: An example showing a car equipped with an Inertial Measurement Unit (IMU), which measures linear accelerations and angular velocities, installed inside the vehicle close to its center of rotation, and a camera that records images of the surroundings, installed on top of the car to provide a good field of view. Each of these sensors makes observations with respect to its own frame, and the transformation between these two frames must be known before fusing their measurements.

a sensor measures are expressed in its own frame of reference (see Fig. 1.3). Fusion algorithms, however, can process measurements corresponding to geometric quantities and provided from multiple sensors only if these are spatially related. This is the reason why we need to know the *sensor-to-sensor transformation*, i.e., so as to express all of the measurements with respect to a common frame of reference. To clarify this, consider a simple example where we want to estimate the position of a comet by averaging the position measurements ${}^{M}\mathbf{z}_1$ and ${}^{S}\mathbf{z}_2$, from sensors of equal accuracy located in Minnesota (represented by the superscript prefix M) and Spain (represented by the superscript prefix S), respectively. Since each sensor measures the position of the comet in its own frame of reference, we need to transform one of the measurements to the other sensor's frame of reference before combining them:

$$ {}^{M}\mathbf{z}_{avg} = \frac{1}{2}\left({}^{M}\mathbf{z}_1 + \mathbf{g}({}^{S}\mathbf{z}_2)\right) \quad , \quad \mathbf{g}({}^{S}\mathbf{z}_2) = {}^{M}\mathbf{z}_2 \tag{1.2} $$

In these equations, the function $\mathbf{g}$, which transforms the measurement $\mathbf{z}_2$ from frame $\{S\}$ to frame $\{M\}$, represents the sensor-to-sensor transformation, generally modeled as a 3 d.o.f. rotation and a 3 d.o.f. translation.

The process of estimating the sensor-to-sensor transformation is called *extrinsic sensor-to-sensor calibration.* Depending on the type of sensors used, there exist two cases of sensor-to-sensor extrinsic calibration:

- **Pairs of sensors whose spatial measurements can be correlated:** In this case, the sensors (typically both exteroceptive) are able to localize themselves with respect to a *common* frame of reference. A well-known example of this case is the stereo camera rig (see Fig. 1.4), where the pose of each camera with respect to

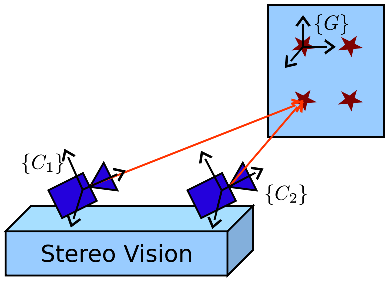


Figure 1.4: The transformation between two cameras rigidly attached to each other can be indirectly estimated by computing the pose of each of them with respect to a common frame of reference. This procedure, however, is not feasible for many other sensor pairs if they cannot directly estimate their pose with respect to an external frame of reference.

jointly observed landmarks is independently computed [3, 53]. Subsequently, the transformation between the two cameras can be readily obtained by combining the sensors' poses with respect to the common frame. Inspired by this principle, one of the main contributions of this dissertation is the development of a novel algorithm for extrinsic calibration of a 3D LIDAR and a camera.

- **Pairs of sensors whose spatial measurements cannot be directly correlated:** In this case, the pose of the two sensors[1] with respect to a *common* frame of reference cannot be obtained. Instead, we need to exploit the fact that they are rigidly connected and use the *perceived motion* by each sensor to deduce the transformation between them. This method has been used for extrinsic calibration of odometers with respect to a camera [24, 81, 5, 46] and 2D laser scanners [23]. In this work, we present two novel methods that employ this principle for extrinsic calibration of inertial sensors with respect to cameras and odometers.

1.1.3 Importance of Accurate Sensor Calibration

In this section, we provide a few examples to illustrate the importance of accurate sensor calibration. Initially consider the simple pinhole camera whose only calibration parameter is its focal length. If the estimate of the focal length is, for example, smaller than its actual value, the object will appear closer (or larger) than it is in reality (see Fig. 1.5). Note that this will result in a systematic error (bias) in the observations of the camera, and if unaccounted, may lead to incorrect results of the algorithm that uses the camera measurements.

[1] Pairs containing two proprioceptive sensors, or, an exteroceptive and a proprioceptive sensor, or, two exteroceptive sensors whose fields of view do not overlap.

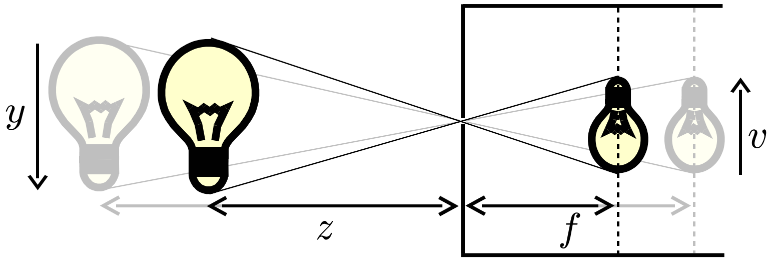
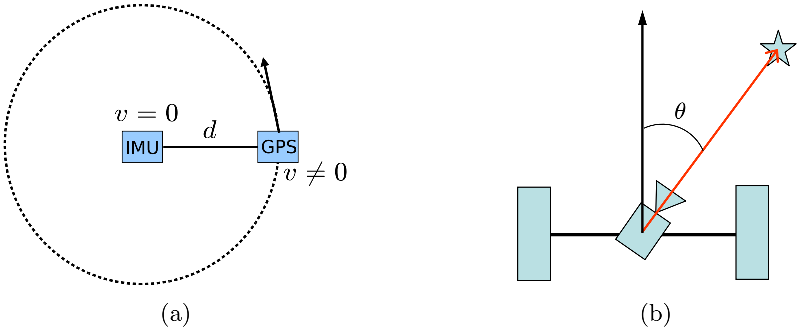


Figure 1.5: The impact of a bad estimate for the focal length on the estimation algorithm. In this figure the true object and focal length are grayed out. An estimate of the focal length that is shorter than the real focal length, shown in black, leads us to infer that the object is closer to the camera than it is in reality.

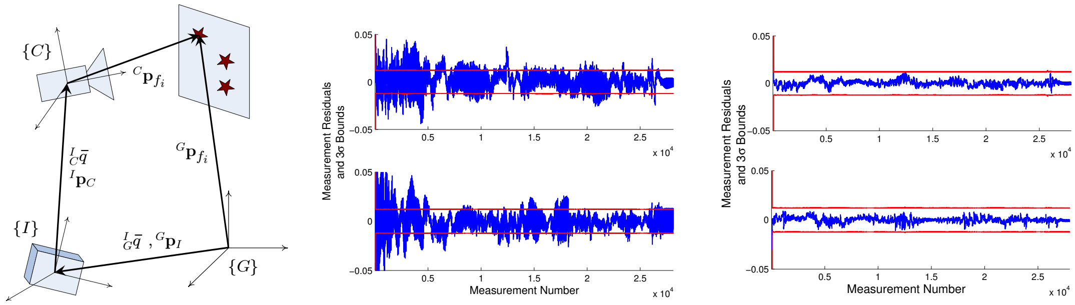


(a)

(b)

Figure 1.6: (a) An illustration of a rigidly connected IMU-GPS pair. If the distance d is not known, fusing measurements of the GPS and the IMU will lead to large errors. (b) Unknown angle between a camera and wheel encoders installed on a vehicle result in systematic errors in the fusion algorithm; however, if the angle is precisely known, it can be easily compensated by expressing both measurements in the same frame of reference.

As a second example, consider an IMU (a proprioceptive sensor that measures linear accelerations and angular velocities) which is commonly used in conjunction with a GPS receiver, in order to estimate the 6 d.o.f. pose of a holonomic vehicle. Often, the IMU is installed close to the center of rotation of the vehicle to avoid saturation, while the GPS antenna is mounted on the outer body of the vehicle, to guarantee high quality signal reception. This setup inevitably results in a large distance between the IMU and GPS. Now, consider an adverse scenario where the vehicle is standing still and then starts rotating around the IMU [see Fig. 1.6(a)]. In this case the GPS measurements indicate nonzero linear velocity, but the integration of the measured linear acceleration by the IMU implies zero velocity. If we do not know the distance between the IMU and the GPS (or more precisely, the transformation between them), there is no way to resolve this contradiction and any algorithm fusing measurements from these two sensors will most likely fail.



(a) (b) (c)

Figure 1.7: (a) Illustration of an IMU-Camera pair installed on a robot. The transformation between the IMU and the camera, represented by $^{I}_{C}\bar{q}^{T}$ and $^{I}\mathbf{p}_{C}^{T}$, must be determined before fusing their measurements. (b) When an extended Kalman filter is used to fuse the measurements from the IMU and the camera, measurement residuals and the predicted 3σ bounds indicate the consistency of the filter. When the IMU-camera transformation is only approximately known, the measurement residuals exceed the 3σ bounds, which suggests that the filter is not functioning optimally. (c) The filter is functioning consistently when an accurate estimate of the IMU-Camera transformation is used. In this case, 99.7% of the measurement residuals lie within the predicted 3σ bounds.

Fig. 1.6(b) shows another example where a mobile robot is equipped with a pair of wheel encoders that measure linear and angular velocity, and a camera observing static landmarks. We consider the case where the camera estimates its position and velocity by processing images of known landmarks [3]. If the miss-alignment θ between the camera and the heading of the robot is unknown, the velocity measurements from the camera and the wheel encoders will contradict each other and fusing them will introduce a systematic error in the motion estimates. However, precise knowledge of the angle between the optical axis of the camera and the heading of the robot will allow us to transform both measurements to the same frame of reference, and then fuse them to obtain a better estimate of the robot's velocity.

A more involved version of the last example is when a vehicle is equipped with an IMU, measuring linear accelerations and angular velocities, and a camera, observing static landmarks. A diagram of this system is depicted in Fig. 1.7(a). Similar to the case of IMU-GPS, the IMU is most likely installed close to vehicle's center of rotation while the camera is mounted on the body of the vehicle to provide a good field of view. In order to fuse measurements from these two sensors, both of them should be expressed with respect to the same frame of reference, requiring precise knowledge of the transformation between the sensors. When an *inaccurate* estimate of the transformation is used, the fusion algorithm does not operate optimally. Fig. 1.7(b) shows the measurement residuals (i.e., the difference between predicted and actual measurements) of an

Extended Kalman Filter (EKF) that is used for sensor fusion in the latter case. As evident, the measurement residuals exceed the predicted 3σ bounds, indicating that the filter is not operating optimally. In this case, we also expect that the estimated pose of the vehicle will diverge from its true value, invalidating the linearization approximation of the EKF, and hence, causing total failure of the fusion algorithm. This situation should be compared and contrasted to the case of precisely known sensor-to-sensor transformation where 99.7% of the measurement residuals, are within the predicted 3σ bounds [see Fig. 1.7(c)].

1.2 Research Objectives

In order to design algorithms that estimate the calibration parameters accurately and reliably, two essential questions must be answered:

- Is the system observable? In other words, do the sensor measurements provide sufficient information for estimating the calibration parameters?
- If the system is observable, is it possible to find the optimal estimate for the calibration parameters given the measurements?

The main objective of this dissertation is to answer these questions for certain sensors and sensor pairs commonly used in robotics and computer vision. In the next two sections, we provide an overview of the key results of this thesis.

1.2.1 System Observability

Intuitively, the observability of a system guarantees that the sensor measurements provide sufficient information for estimating the unknown states. Various tools are available for observability analysis. In particular, if the system is linear, one can exploit the Observability Gramian [16] or the Popov-Belevitch-Hautus (PBH) test [110] to prove (un-)observability. Most sensor-calibration systems, however, are nonlinear and their observability properties may not be proved using the aforementioned methods. Instead, in this work we employ *Lie-derivative*-based analysis [51, 101, 59] to prove observability of gyroscope-odometer and IMU-camera calibration systems. The Lie-derivative-based observability analysis directly takes into account the impact of various control inputs on the observability of the nonlinear system. This, in turn, enables us to determine the conditions that if the control inputs satisfy, we can guarantee the calibration system's observability.

While the Lie-derivative-based observability analysis is suitable for sensor pairs that involve proprioceptive sensors and hence dynamic states (e.g., IMU velocity and biases), it is not as effective for systems that only involve static parameters. An example of such system is the case of 3D LIDAR-camera calibration. In this case, we prove the observability of a 3D LIDAR-camera calibration system using an alternative technique.[2] Specifically, we show that under certain conditions, there exist only a finite number of calibration parameters that can produce a given set of measurements.

We make several assumptions for proving the observability of each of the above-mentioned sensor-calibration systems. For the case of IMU-camera calibration, we assume that the camera observes at least four landmarks whose locations are *a priori* known in the global frame of reference. For both gyroscopes-odometer calibration and IMU-camera calibration systems, we assume precise time-synchronization between the sensor measurements, and neglect the (possibly time-varying) time delays. Finally, for the case of 3D LIDAR-camera calibration we assume that a subset of the intrinsic parameters of the 3D LIDARs are known, in order to prove the observability of the system for estimating the remaining intrinsic and extrinsic calibration parameters.

The direct impact of the provided analysis is to describe the conditions (e.g., control input, number of measurements) under which the observability of the calibration systems is guaranteed. The indirect benefit of the presented analysis is to provide an insight as how to investigate the observability of other challenging sensor calibration problems. To this end, an algorithm has been proposed in [65] to extend our IMU-camera calibration approach to the case of Simultaneous Localization and Mapping (SLAM), when no known landmarks are available.

1.2.2 Optimality of the Estimator

As important as the observability analysis is, it does not provide all the information required to efficiently estimate the unknown calibration parameters. In particular, the observability analysis does not say how we can estimate the unknowns, even if the system is observable. In the absence of noise, we can attempt to directly solve the geometric constraints relating the unknowns and the measurements. The difficulty of this *deterministic approach* is that the geometric constraints are almost always nonlinear, and solving them is often nontrivial. In these situations, we can use iterative

[2] Static systems whose parameters can be estimated from their measurements are more precisely called *identifiable* instead of observable. To simplify the presentation, however, in this work we call any system whose measurements contain sufficient information to estimate their calibration parameters as observable, regardless of their static or dynamic nature.

solvers, such as Newton-Raphson [105], to find the solutions to the geometric constraints. These iterative methods, however, require initialization, and may not find all the solutions if more than one exist. A common technique to address this issue is to convert the geometric constraints to a system of polynomial equations, and then solve the system by employing techniques from algebraic geometry (see for example [3, 102, 119, 19, 142, 21, 20, 127, 126, 143, 144]).

In practice the sensor measurements are always noisy and their corresponding geometric constraints are not exactly satisfied. Solving such constraints without accounting for noise leads to inaccurate or even infeasible solutions and does not provide any measure of optimality. This issue can be addressed by directly taking the effect of noise into account and following a *stochastic approach*. In particular, acknowledging that the geometric constraints are not exactly satisfied, one can attempt to minimize their residuals in a least-squares framework. Due to the nonlinearity of the geometric constraints, the consequent least-squares problem is often nonconvex and its solution is nontrivial. Iterative techniques such as Gauss-Newton [63] are usually employed to solve these nonlinear least-squares problems. However, the accuracy and performance of these iterative methods depends on their initialization. Moreover, they provide no guarantees of convergence to the global optimum. In practice, iterative solvers are often initialized with the estimates provided by a deterministic approach. In this way, however, the least-squares refinement inherits the deficiencies of the deterministic method and may still converge to a local minimum far from the global one.

Inspired by [125], we follow and extend a new paradigm, called *Analytical Nonlinear Least Squares (ANLS)*, to obtain the guaranteed optimal estimates for the unknown parameters. In particular, we we first convert the geometric constraints into polynomial equations and form a polynomial least-squares cost function whose optimality conditions comprise a system of *multivariate* polynomial equations. We then solve this polynomial system using techniques from algebraic geometry to find the critical points of the least-squares cost function, and among them, select as guaranteed global optimum the critical point that minimizes the cost function. In this thesis, we show the outstanding performance of this method for extrinsic calibration of cameras from line-segment observations.

Despite the effectiveness of the ANLS method for extrinsic camera calibration, it cannot be applied to problems with large number of unknown parameters. One such problem is the 3D LIDAR-camera calibration which requires estimating hundreds of unknown parameters. In this case, we relax the problem and divide it into smaller ones each of which can be solved using the ANLS technique. The solution to the relaxed problem is then used to initialize an iterative least-square refinement. Although in this case the optimality of the estimated solutions cannot be guaranteed anymore, through experimental evaluation we have demonstrated that the achieved accuracy outperforms that of competing methods.

A key assumption that we make in order to be able to develop the above-mentioned estimators is that the measurements do not contain outliers. When outliers do exist in the measurements, we need to employ a solver with minimum number of required measurements (so-called minimal solver) in the RANdom SAmple Consensus (RANSAC) framework [41, 48] to identify and reject the outliers at the pre-processing stage.

The geometric nature of most problems in computer vision and robotics means that they often can be expressed using polynomial constraints. Thus, the methodology and techniques developed in this thesis can be leveraged to address the issue of optimality in such problems.

1.3 Structure of the Manuscript

The rest of this manuscript is structured as follows: Chapter 2 describes the extrinsic gyroscope-odometer calibration problem, and provides an analysis of the observability of the system. An efficient estimator that takes into account different sampling mechanisms of odometer and gyroscopes is developed and validated in real experiments. Chapter 3 discusses the problem of extrinsic IMU-camera calibration, proves its observability under certain conditions, and describes the estimators that have to be implemented for performing the calibration. Extensive simulations and experimental validation are provided to demonstrate the performance of the proposed method. In Chapter 4 the problem of intrinsic and extrinsic calibration of a 3D LIDAR-camera pair is investigated, and the conditions under which the system is observable are studied. Then, a relaxation of this problem is presented and solved using the ANLS technique, followed by a batch least-squares refinement. The accuracy of the estimated calibration parameters in real experiments are compared to those obtained from alternative methods. Chapter 5 presents an algorithm based on the ANLS methodology for extrinsically calibrating a camera using observations of known line segments. The performance of this method is compared to competing approaches in simulation and experiments. Subsequently, in Chapter 6 the focal length and the rotational component of the extrinsic calibration of a camera, corresponding to the camera's vanishing points, is estimated using the ANLS technique in an urban environment. In this case, the only assumption used is that most of the lines detected in the image are along the three cardinal directions. The developed method in this chapter is extensively tested using online image datasets. Finally, in Chapter 7 concluding remarks and directions for future work are provided.

Chapter 2

Gyroscope-Odometer Calibration

2.1 Introduction

Odometers are among the most widely used proprioceptive sensors for measuring ego-motion in mobile robotics. Often consisting of two wheel encoders, they measure the average velocities of the robot's right and left wheels, based on which, the average rotational and linear velocities of the robot are computed. Open-loop integration of these measurements (i.e., dead-reckoning) yields an estimate for the position and heading (pose) of the robot. The accuracy of these estimates, however, quickly deteriorates with time due to integration of noise in the encoder measurements. Additionally, odometers are highly susceptible to faults such as wheel slippage or stalling; thus without appropriate safeguards, their measurements can be unreliable.

To tackle these issues, additional *auxiliary* sensors are often used to improve the accuracy of the pose estimates, and they provide redundancy to allow odometry fault detection. Gyroscopes are among the most promising auxiliary sensors and have received significant attention over the past several years [79, 37, 28, 109, 70, 95]. As a proprioceptive sensor, the main advantage of gyroscopes is their independence from the environment they operate in. This is in contrast with GPS receivers, cameras, and laser scanners, which work only outdoors, require good lighting conditions, or depend on surrounding static obstacles, respectively. Nevertheless, all exteroceptive sensors can be used in conjunction with gyroscopes and odometers when the robot operates in appropriate environments.

Fusing measurements of gyroscopes and odometers, while compelling, requires addressing the following challenges:

- *Temperature-dependent scale factor and time-varying biases of gyroscopes*: While the former issue (i.e., scale factor) is mostly addressed in commercially-available temperature-compensated products, the latter persists even in tactical-grade gyroscopes. Therefore the biases need to be estimated in real-time, in order to make the best use of gyroscope measurements.

- *The gyroscope-odometer extrinsic calibration*: The prerequisite for optimally fusing two sensors' measurements is precise knowledge of the misalignment between them (see Fig. 2.1). This misalignment can be due to imperfect manufacturing, or environmental changes such as temperature changes. Manual measurement of the transformation between the two sensors is often impractical or imprecise. Employing calibration equipments such as 3D laser scanners can be prohibitively expensive or time-consuming.

- *Ensuring observability of the system*: Similar to any other sensor fusion algorithm, the most important challenge is to ensure that the gyroscope-odometer data fusion system is observable at all times, thus allowing accurate estimation of the unknown parameters (e.g., extrinsic calibration, gyroscope biases, etc.).

- *Difference in the sampling mechanism and frequency*: While odometers measure the *average* velocity by counting the number of encoder ticks in constant periods of time (e.g., 100 ms), gyroscopes measure *instantaneous* rotational velocity at a much higher rate (e.g., 100 Hz or 10 ms). Clearly, except for constant velocity motions, these two measurements are not equal even in the absence of noise and biases; thus, combining them requires additional care.

While the first and last challenges are reasonably addressed in the literature, the other two are widely neglected. Specifically, the gyroscope-odometer transformation is often roughly calculated from technical drawings, leading to sub-optimality of the fusion algorithm. The system observability is also commonly overlooked, even though the lack of observability can lead to inaccurate estimation of the unknowns, or even divergence of the fusion algorithm. In this chapter, we address all these four issues simultaneously. Specifically, we describe an Extended Kalman Filter (EKF)-based algorithm that estimates the gyroscope biases and extrinsic calibration parameters, and appropriately accounts for different sampling mechanisms of the sensors. Furthermore, we analytically prove that the proposed data fusion and calibration system is *locally observable* [51], thus allowing accurate estimation of the unknown parameters. Moreover, while the proposed approach already allows efficient statistical fault detection for

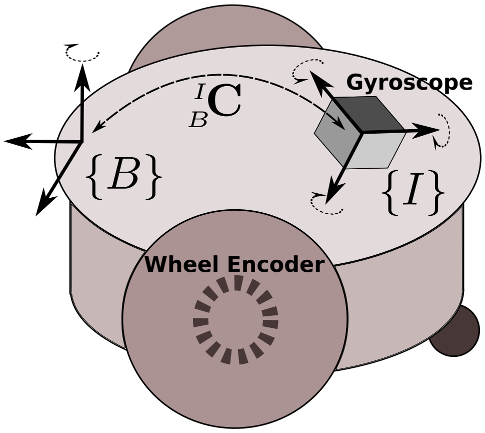


Figure 2.1: A differential-drive robot equipped with a gyroscope and an odometer. The odometer, consisting of two wheel encoders, measures the average rotational velocity around the z-axis of frame $\{B\}$ (marked by dotted arc). The triaxial gyroscope, rigidly mounted on the robot, measures instantaneous rotational velocities around three cardinal axes of frame $\{I\}$, whose orientation with respect to $\{B\}$ is denoted by the rotational matrix ${}^{I}_{B}\mathbf{C}$.

the odometer, it can be easily augmented to include measurements from additional exteroceptive sensors such as laser scanners and cameras.

The remainder of this chapter is organized as follows. Section 2.2 provides an overview of the related literature and Section 2.3 presents the problem formulation. A brief introduction to nonlinear observability and the observability analysis of the gyroscope-odometer calibration and data fusion system is presented in Section 2.4. Our EKF-based discrete-time estimator is described in Section 2.5, and validated in simulation and experiments in Section 2.6. A summary of this chapter is provided in Section 2.7.

2.2 Related Work

Improving the odometer's accuracy by using a gyroscope has received significant attention over the past several years. The early work by Maeyama *et al.* [79] compares odometry and gyroscope measurements, and fuses them if they do not disagree significantly. In [37], Dissanayake *et al.* propose a method for fusing the vehicle's velocity, measured by an odometer, with gyroscope measurements using an information filter. In [28], two estimates for the robot's heading based on gyroscope and odometer measurements are tracked in a Kalman filter. The equality between these two estimates is used as an inferred measurement to update the filter. Various modifications and improvements of this method are proposed in [109, 70, 95]. Specifically, in [109] the state vector is augmented to include intrinsic odometer parameters, and in [70] a Gauss-Markov model is used to propagate the gyroscope biases. More recently in [95], the observability Grammian of the data fusion system is numerically computed and an alternative measurement update is proposed to improve the estimation accuracy.

The main limitation of the aforementioned methods is that the transformation between the gyroscope and the odometer is assumed to be *a priori* known. This is a very restrictive assumption, since there is usually some error in the alignment of the gyroscope and the odometer due to imperfect manufacturing. Furthermore, the alignment between the gyroscope and the robot's body may change due to environmental conditions, such as temperature. Manual measurement of this misalignment is often impractical or imprecise, and a special purpose calibration procedure is required. One solution is to augment the state vector of the existing estimators with the gyroscope-odometer extrinsic transformation. However, without necessary considerations, this may result in an unobservable system, whose state vector cannot be accurately estimated from the sensor measurements.

In this chapter, we address these issues and propose an EKF-based algorithm for simultaneous data fusion and extrinsic calibration of gyroscopes and odometers. Using an approach similar to [81], we prove that the underlying system is *locally observable* and the extrinsic calibration parameters can be accurately estimated. Additionally, while taking the different sampling mechanisms in gyroscopes and odometers into account, the proposed algorithm provides a statistical test for detection of odometric faults.

2.3 Problem Formulation

The objective of this work is to find an efficient way of obtaining the robot's heading by fusing measurements from the robot's odometer and a triaxial rate gyroscope[1] that is mounted rigidly on the robot. This requires precise estimates of the *a priori* unknown rotational transformation between the gyroscope and the odometer as well as the unknown and time-varying biases affecting the gyroscope measurements. In order to employ an estimator (e.g., EKF, Maximum A Posteriori (MAP) estimator, etc.) to determine these unknowns along with the robot's heading, we need to formulate a dynamical system relating them to the sensor measurements. While it is possible to

[1] In this chapter we describe the most general case where the gyroscope is triaxial. Single- and double-axes gyroscopes can be easily considered as special instances of this general case.

design several such dynamical models, not all of them are guaranteed to be observable [110]. Observability of a dynamical system is of paramount importance to ensure the possibility of accurately estimating the unknown state vector given the measurements. In the following, we propose one such formulation that is guaranteed to be locally observable [51].

We assume that the robot moves on a 2D plane, and its rotational velocity is represented by the scalar $\omega(t)$, whose *noisy* and *time-averaged* measurements are provided by the on-board odometer (see Fig. 2.1). If we attach the frame of reference $\{B\}$ to the robot's body such that its z-axis is perpendicular to the plane of motion, then the 3D rotational velocity of the robot expressed in $\{B\}$ is ${}^{B}\boldsymbol{\omega}(t) = [0 \quad 0 \quad \omega(t)]^{T}$. Note that the x and y components of ${}^{B}\boldsymbol{\omega}(t)$ are zero since the robot motion is planar. On the other hand, the gyroscope measurements are provided in its own frame of reference, $\{I\}$, whose rotational transformation to $\{B\}$, ${}^{I}_{B}\mathbf{C}$, is not accurately known. However, since the robot motion is confined to a 2D plane, we do not need all the components of ${}^{I}_{B}\mathbf{C}$ to fuse measurements of the gyroscope and the odometer. This can be seen by transforming ${}^{B}\boldsymbol{\omega}(t)$ to frame $\{I\}$:

$$ {}^{I}\boldsymbol{\omega}(t) = {}^{I}_{B}\mathbf{C}\, {}^{B}\boldsymbol{\omega}(t) \triangleq \mathbf{c}\, \omega(t). \tag{2.1}$$

In this equation $\mathbf{c}$ is the 3×1 unit vector comprising the third column of ${}^{I}_{B}\mathbf{C}$.

Based on this discussion, we compose the following 7×1 state vector,

$$\mathbf{x}^{T}(t) = [\omega(t) \quad \mathbf{c}^{T}(t) \quad \mathbf{b}^{T}(t)] \tag{2.2}$$

which, in addition to the already introduced quantities of interest, contains $\mathbf{b}$, the 3×1 vector of time-varying biases affecting the gyroscope measurements. The dynamical system describing the time evolution of this state vector is:

$$\dot{\omega}(t) = n_u(t), \quad \dot{\mathbf{c}} = \mathbf{0}_{3\times 1}, \quad \dot{\mathbf{b}} = \mathbf{n}_b(t). \tag{2.3}$$

In this model, $n_u(t)$ and $\mathbf{n}_b(t)$ can be considered as the *time-varying* control inputs driving the rotational velocity and the gyroscope's biases. The essential difference between these two is that we cannot control, or in any way modify the driving input of the gyroscope's biases, while the driving control input of the rotational velocity (i.e., the rotational acceleration) is under our control, since we can command the robot to accelerate, decelerate, or stop. However, we do not have a precise knowledge of the values of neither $n_u(t)$ nor $\mathbf{n}_b(t)$ (and that is not needed for proving observability as it will be discussed in the next section). Therefore in the estimator design, we model both of them random variables drawn from white zero-mean Gaussian distributions with standard deviations σ_u and σ_b, respectively. Finally, note that the time-derivative of $\mathbf{c}$ is zero, since the gyroscope is rigidly mounted on the robot.

Employing (2.1), the 3×1 vector of measurements from the triaxial gyroscope is expressed as a *nonlinear* function of the state vector:

$$\mathbf{h}_g(\mathbf{x}) = {}^I\boldsymbol{\omega}_m(t) = \mathbf{c}\,\omega(t) + \mathbf{b}(t) + \mathbf{n}_g(t) \tag{2.4}$$

where $\mathbf{n}_g(t)$ is the zero-mean Gaussian noise with covariance $\sigma_g^2\mathbf{I}_3$ affecting the gyroscope measurements. The odometer, on the other hand, measures the robot's average rotational velocity between two sampling time instants t_j and t_{j+1}. However, for the purpose of observability analysis, we can assume $t_{j+1} - t_j$ is infinitesimally small such that the odometer measures instantaneous rotational velocity. Then, these measurements can be expressed as:

$$h_o(\mathbf{x}) = \omega(t) + n_o(t) \tag{2.5}$$

where n_o is the zero-mean Gaussian measurement noise with standard deviation $\sigma_o(t)$.

2.4 Nonlinear Observability Analysis

A *linear* dynamical system is observable if its state at a certain time instant can be uniquely determined given a finite sequence of its outputs [110]. Intuitively this means that the measurements of an observable system provide sufficient information for estimating its state. In particular, the constant, but otherwise unknown components of the state vector of an observable system can be estimated with arbitrarily small uncertainty given sufficient number of measurements. Moreover, the time-varying components of the state vector of an observable system can be estimated with bounded uncertainty. In contrast, the state vector of unobservable systems cannot be recovered with bounded uncertainty regardless of the duration of the estimation process [83]. The observability of linear systems can be investigated by employing any of the well-known observability tests such as the rank of the Observability Gramian [83] or the Popov-Belevitch-Hautus (PBH) test [110] (the latter is only applicable for time-invariant systems).

The concept of observability for *nonlinear* dynamical systems is more involved. In particular, the nonlinear observability is often considered *locally*, in a neighborhood of a state, since this can be examined by a simple algebraic test called the observability rank condition [51]. Contrary to the linear observability, the nonlinear local observability analysis does not provide any guarantee of "estimatability" of the associated state vector. Instead, it only proves whether there exist any control inputs that make a state distinguishable from its neighbors based on the output measurements.[2] However, this does not mean that one can always estimate the state of a locally observable system, since appropriate control inputs are required in order to differentiate a state from its neighbors.

To formalize our discussion, we hereafter provide a brief overview of local observability and associated concepts. The interested reader is referred to [101, 59] for detailed treatments of the subject. Consider the state-space representation of the following infinitely-smooth affine nonlinear system:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}_0(\mathbf{x}) + \mathbf{f}_1(\mathbf{x})u_1 + \ldots + \mathbf{f}_l(\mathbf{x})u_l \\ \mathbf{y} = \mathbf{h}(\mathbf{x}) \end{cases} \tag{2.6}$$

where $\mathbf{x} \in M \subset \mathbb{R}^n$ is the state vector, $\mathbf{u} = [u_1 \;, \ldots, \; u_l]^T \in \mathbb{R}^l$ is the vector of control inputs, and $\mathbf{y} = [y_1 \;, \ldots, \; y_m]^T \in \mathbb{R}^m$ is the output measurement vector, with $y_k = h_k(\mathbf{x})$, $k = 1 \ldots m$. Given an initial state $\mathbf{x}_0$ at $t = 0$, the state vector at any time instant $t > 0$ can be computed by integrating (2.6). Let us denote the corresponding output function as $\mathbf{y}(t, \mathbf{x}_0, \mathbf{u})$.

Definition 2.1. We call two states $\mathbf{x}_0$ and $\mathbf{x}_1$ *indistinguishable* (denoted as $\mathbf{x}_0\mathcal{I}\mathbf{x}_1$) if for every admissible function $\mathbf{u}$ the output functions $\mathbf{y}(t, \mathbf{x}_0, \mathbf{u})$ and $\mathbf{y}(t, \mathbf{x}_1, \mathbf{u})$ are *identical* for any $t \geq 0$. A system is called *locally observable at* $\mathbf{x}_0$ if there exists a neighborhood of $\mathbf{x}_0$ such that the relation $\mathbf{x}_0\mathcal{I}\mathbf{x}_1$ implies $\mathbf{x}_0 = \mathbf{x}_1$. If this property is satisfied at any $\mathbf{x}_0$, the system is called *locally observable* [101].

Intuitively, for a locally observable system, every state $\mathbf{x}_0$ can be distinguished from its neighbors by looking at output measurements when the system trajectories remain close to $\mathbf{x}_0$. The importance of local observability is due to its close relationship with the *observation space*, $\mathcal{O}$, which is obtained from the Lie derivatives of the system.

The zeroth-order Lie derivative of any (scalar) function is the function itself, i.e., $\mathfrak{L}^0 h_k(\mathbf{x}) = h_k(\mathbf{x})$. The first-order Lie derivative of function $h_k(\mathbf{x})$ with respect to $\mathbf{f}_i$ is

[2] Note that if a nonlinear system is *not* locally observable, there is no control input that can make a state distinguishable from its neighbors, and thus it is impossible to estimate the state from the outputs.

defined as:

$$\begin{aligned}\mathfrak{L}^1_{\mathbf{f}_i} h_k(\mathbf{x}) &= \frac{\partial h_k(\mathbf{x})}{\partial x_1} f_{i1}(\mathbf{x}) + \cdots + \frac{\partial h_k(\mathbf{x})}{\partial x_n} f_{in}(\mathbf{x}) \\ &= \nabla h_k(\mathbf{x}) \cdot \mathbf{f}_i(\mathbf{x}) \end{aligned} \tag{2.7}$$

where $\mathbf{f}_i(\mathbf{x}) = [f_{i1}(\mathbf{x})\ , \ldots ,\ f_{in}(\mathbf{x})]^T$, ∇ represents the gradient operator, and '$\cdot$' denotes the vector inner product. Considering that $\mathfrak{L}^1_{\mathbf{f}_i} h_k(\mathbf{x})$ is a scalar function itself, the second-order Lie derivative of $h_k(\mathbf{x})$ with respect to $\mathbf{f}_j$ is:

$$\mathfrak{L}^2_{\mathbf{f}_i} h_k(\mathbf{x}) = \mathfrak{L}^1_{\mathbf{f}_j} \mathfrak{L}^1_{\mathbf{f}_i} h_k(\mathbf{x}) = \nabla \mathfrak{L}^1_{\mathbf{f}_i} h_k(\mathbf{x}) \cdot \mathbf{f}_j(\mathbf{x}). \tag{2.8}$$

Other higher-order Lie derivatives are defined similarly. Additionally, it is possible to define mixed Lie derivatives, i.e., with respect to different functions of the process model. For example, the second-order Lie derivative of h_k with respect to $\mathbf{f}_j$ and $\mathbf{f}_i$, given its first derivative with respect to $\mathbf{f}_i$, is:

$$\mathfrak{L}^2_{\mathbf{f}_j \mathbf{f}_i} h_k(\mathbf{x}) = \mathfrak{L}^1_{\mathbf{f}_j} \mathfrak{L}^1_{\mathbf{f}_i} h_k(\mathbf{x}) = \nabla \mathfrak{L}^1_{\mathbf{f}_i} h_k(\mathbf{x}) \cdot \mathbf{f}_j(\mathbf{x}) \tag{2.9}$$

Based on the preceding expressions for the Lie derivatives, the observation space, $\mathcal{O}$, is defined as the linear space over $\mathbb{R}$ containing $h_k(\mathbf{x})$ and all possible Lie derivatives

$$\mathfrak{L}^1_{\mathbf{f}_i} \cdots \mathfrak{L}^1_{\mathbf{f}_j} h_k(\mathbf{x}) \quad i, \ldots, j \in \{0, \ldots, l\}; k = 1, \ldots, m \tag{2.10}$$

Alternatively, the observation space can be defined as the linear space of functions containing $h_k(\mathbf{x}),\ k = 1, \ldots, m$ and all the following Lie derivatives:

$$\mathfrak{L}^1_{\mathbf{g}_i} \cdots \mathfrak{L}^1_{\mathbf{g}_j} h_k(\mathbf{x}) \quad i, \ldots, j \in \{0, \ldots, s\}; k = 1, \ldots, m \tag{2.11}$$

where $\mathbf{g}_\ell(\mathbf{x}) = \mathbf{f}_0(\mathbf{x}) + \mathbf{f}_1(\mathbf{x}) u_1^\ell + \ldots + \mathbf{f}_l(\mathbf{x}) u_l^\ell$ for s different points $\mathbf{u}^\ell = [u_1^\ell\ , \ldots ,\ u_l^\ell]^T$, $\ell \in \{1, \ldots, s\}$. The equivalence of these two definitions is proved in [101].

Theorem 2.1. *Consider the system* (2.6)*. If* rank$(\nabla\mathcal{O}(\mathbf{x}_0)) = n$*, where* $\nabla\mathcal{O}(\mathbf{x}_0)$ *(known as Observability Matrix) is a matrix whose rows are the gradients of the members of* $\mathcal{O}$ *at* $\mathbf{x}_0$*, then the system is locally observable at* $\mathbf{x}_0$*.*

Proof. Since rank$(\nabla\mathcal{O}(\mathbf{x}_0)) = n$, there exist n smooth functions $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_n \in \mathcal{O}$, such that $\nabla\boldsymbol{\phi}_1(\mathbf{x}_0), \ldots, \nabla\boldsymbol{\phi}_n(\mathbf{x}_0)$ are linearly independent. Then the differentiable map

$\mathbf{\Phi}\colon M \to \mathbb{R}^n$ defined as $\mathbf{\Phi}(\mathbf{x}) = [\boldsymbol{\phi}_1(\mathbf{x}) \ , \dots , \ \boldsymbol{\phi}_n(\mathbf{x})]^T$ is bijective[3] in a neighborhood W of $\mathbf{x}_0$. Now suppose $\mathbf{x}_0 \mathcal{I} \mathbf{x}_1$ for some $\mathbf{x}_1 \in W$, then for small $t_1, \dots, t_\ell$ we have

$$h_k(\xi_\ell^{t_\ell} \circ \xi_{\ell-1}^{t_{\ell-1}} \circ \cdots \circ \xi_1^{t_1}(\mathbf{x}_0)) = h_k(\xi_\ell^{t_\ell} \circ \xi_{\ell-1}^{t_{\ell-1}} \circ \cdots \circ \xi_1^{t_1}(\mathbf{x}_1))$$

for any ℓ, where $\circ$ denotes composite functions, and $\xi_\gamma^{t_\beta}(\mathbf{x}_\alpha)$ corresponds to integration of $\mathbf{g}_\gamma$ from t_α to t_β with $\mathbf{x}_\alpha$ as initial condition. Differentiating both sides with respect to $t_1, \dots, t_\ell$ at $t_1 = 0, \dots, t_\ell = 0$ yields

$$\mathfrak{L}_{\mathbf{g}_1} \cdots \mathfrak{L}_{\mathbf{g}_\ell} h_k(\mathbf{x}_0) = \mathfrak{L}_{\mathbf{g}_1} \cdots \mathfrak{L}_{\mathbf{g}_\ell} h_k(\mathbf{x}_1) \tag{2.12}$$

for all ℓ. From the definition of the observation space it follows that $\boldsymbol{\phi}(\mathbf{x}_0) = \boldsymbol{\phi}(\mathbf{x}_1)$ for all $\phi \in \mathcal{O}$. Thus, $\mathbf{\Phi}(\mathbf{x}_0) = \mathbf{\Phi}(\mathbf{x}_1)$, and from the bijectivity of $\mathbf{\Phi}$ on W, we conclude that $\mathbf{x}_0 = \mathbf{x}_1$. Thus, based on Definition 2.1, the system is locally observable at $\mathbf{x}_0$. □

Remark 2.1. As it can be seen from the proof of Theorem 2.1, the exact value of the control input $\mathbf{u}$ is irrelevant in the study of local observability of a nonlinear dynamical system.[4] Instead, we need to change the control input sufficiently to generate different $\mathbf{g}_\ell(\mathbf{x})$ (and corresponding Lie derivatives) with linearly independent gradients so as to guarantee bijectivity (diffeomorphism) of $\mathbf{\Phi}$.

Remark 2.2. Since the process and measurement functions [see (2.6)] are infinitely-smooth, the observability matrix $\nabla\mathcal{O}(\mathbf{x})$ can have infinite number of rows. However, to prove that $\nabla\mathcal{O}(\mathbf{x})$ is full rank, it suffices to show that a subset of its rows are linearly independent.

In general, there exists no systematic method for selecting the suitable Lie derivatives and corresponding rows of $\nabla\mathcal{O}(\mathbf{x})$ when examining the observability of a system. Instead, this selection is performed by sequentially considering the directions of the state space along which the gradient of each of the candidate Lie derivatives provides information.

Bonnifait and Garcia were the first to employ Theorem 2.1 for examining the observability of map-based bearing-only single-robot localization in 2D [13]. Later on,

[3] More precisely, this map is a diffeomorphism in a neighborhood W of $\mathbf{x}_0$.

[4] Although the local observability of a system does not require knowledge of the exact values of the control inputs, it is impossible to estimate the state vector if no information is available about them. In particular, in the gyroscope-odometer calibration problem, we assume that limited information about the control inputs is provided in the form of a finite-covariance probability distribution function from which the control inputs take their values [see (2.3)].

Martinelli and Siegwart [82] used Lie derivatives to analyze the observability of cooperative localization for pairs of mobile robots navigating in 2D. In a related problem, Mariottini *et al.* [80] investigated the observability of 2D leader-follower formations based on Lie derivatives and the observability rank condition. Recently, Lie derivatives were also used for examining the observability of the single-robot simultaneous localization and mapping (SLAM) in 2D [72], and of the camera-odometer extrinsic calibration process in 2D [81].

2.4.1 Observability of the Gyroscope-Odometer System

Let us re-write the process equations describing the time evolution of the state vector for the gyroscope-odometer system [see (2.3)] as:

$$\begin{bmatrix} \dot{\omega} \\ \dot{\mathbf{c}} \\ \dot{\mathbf{b}} \end{bmatrix} = \mathbf{f}_1 \, n_u + \underline{\mathbf{f}}_2 \, \mathbf{n}_b \quad , \quad \mathbf{f}_1 \triangleq \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \; , \quad \underline{\mathbf{f}}_2 \triangleq \begin{bmatrix} \mathbf{0}_{4\times 3} \\ \mathbf{I}_3 \end{bmatrix} . \tag{2.13}$$

As mentioned before, the exact value of the control inputs n_u and $\mathbf{n}_b$ are irrelevant to the observability analysis. Instead, we only need to excite them so as to ensure the gradients of the observation space $\mathcal{O}(\mathbf{x})$ are linearly independent for any $\mathbf{x}$. It is easy to show that this condition is always satisfied by considering the following members of $\mathcal{O}(\mathbf{x})$:

$$\mathcal{O}' = \{h_o \; , \; \mathbf{h}_g \; , \; \mathfrak{L}_{\mathbf{f}_1} \mathbf{h}_g = \mathbf{c}\} \subset \mathcal{O}(\mathbf{x}). \tag{2.14}$$

The inclusion of the last element, i.e., the first-order Lie derivative of $\mathbf{h}_g$ with respect to $\mathbf{f}_1$, is based on the assumption that n_u can be excited (hence $\omega(t)$ is time-varying). On the other hand, the exclusion of the Lie derivatives with respect to $\underline{\mathbf{f}}_2$ reflects no commitment to excite $\mathbf{n}_b$. This is important since we have no practical way of exciting, or in any way changing, the gyroscope's biases. To show that $\nabla\mathcal{O}(\mathbf{x})$ is full rank, we compute the gradient of the elements of $\mathcal{O}'$:

$$\begin{bmatrix} \nabla h_o \\ \nabla \mathbf{h}_g \\ \nabla \mathfrak{L}_{\mathbf{f}_1} \mathbf{h}_g \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} \\ \mathbf{c} & \omega \mathbf{I}_3 & \mathbf{I}_3 \\ \mathbf{0}_{3\times 1} & \mathbf{I}_3 & \mathbf{0}_{3\times 3} \end{bmatrix} . \tag{2.15}$$

This matrix has always rank 7, regardless of the values of $\omega(t)$ and $\mathbf{c}$. Therefore,

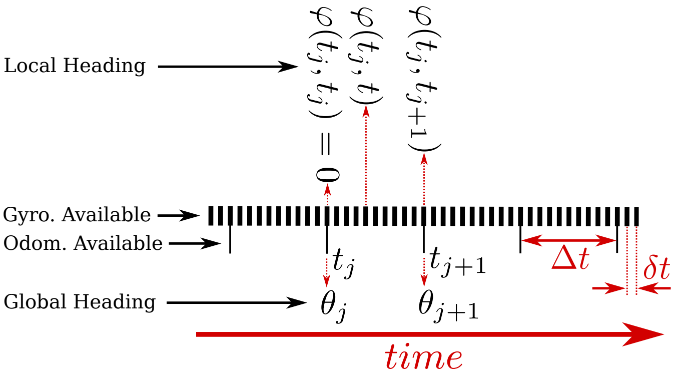


Figure 2.2: The timeline demonstrating the availability of synchronized gyroscope and odometer measurements. The long narrow ticks with period Δt represent time instants when odometer measurements become available, and short ticks with period δt indicate when gyroscope measurements are provided. Global and local robot's headings are also shown for interval $[t_j, t_{j+1}]$.

the dynamical system described in (2.3) is locally observable.[5]

Observability of the above gyroscope-odometer system implies the *possibility* of estimating the state vector (2.2) (in particular the extrinsic calibration parameters and the gyroscope biases) from the output measurements. In the next section, we introduce a novel algorithm that achieves this goal.

2.5 Estimator Design

As mentioned before, the sampling mechanism and frequency of gyroscopes and odometers are different. Specifically, gyroscopes provide high-frequency measurements of *instantaneous* rotational velocity, $\omega(t)$, which can be directly used to update a Kalman filter based on (2.3) and (2.4). On the other hand, practical odometers report low-frequency measurements of the *time-averaged* rotational velocity between two consecutive sampling time instants, t_j and t_{j+1}. This average velocity is obtained by dividing the change in the robot's heading between t_j and t_{j+1} by $\Delta t \triangleq t_{j+1} - t_j$. Clearly, for large values of Δt (e.g., 0.1 s) these measurements are not described by the model in (2.5), and cannot be directly used to update a Kalman filter based on the system model of (2.3). To overcome this issue, we hereafter describe a method for directly fusing the odometer measurements of *incremental change* in orientation with gyroscope measurements.

[5] In the case of a single- or double-axis gyroscope, the system's observability is preserved by removing the components of $\mathbf{b}$ and $\mathbf{c}$ corresponding to the missing axes of the gyroscope from the state vector.

Let us consider the time interval $[t_j, t_{j+1}]$ between two consecutive odometer sampling time instants (see Fig. 2.2). We introduce the variable θ_j, the robot's *global heading* at time t_j, which remains constant over the interval $[t_j, t_{j+1}]$. Additionally, we denote with $\varphi(t_j, t)$, $t_j \leq t \leq t_{j+1}$, the robot's *local heading* relative to its heading at time t_j. Note that $\varphi(t_j, t)$ can be determined by integrating the following differential equation:

$$\dot{\varphi}(t_j, t) = \omega(t), \quad t \in [t_j, t_{j+1}], \quad \varphi(t_j, t_j) = 0. \tag{2.16}$$

It is clear that using this notation, the robot's global heading at any time $t \in [t_j, t_{j+1}]$ is simply $\theta_j + \varphi(t_j, t)$.

We proceed by augmenting the state vector in (2.2) to include these two variables:

$$\mathbf{x}^j(t) = [\varphi(t_j, t) \;\; \omega(t) \;\; \mathbf{c}^T(t) \;\; \mathbf{b}^T(t) \;\; \theta_j]^T. \tag{2.17}$$

Next, we describe how to estimate this state vector using gyroscope and odometer measurements. For simplicity, we assume that the gyroscope and odometer sampling clocks are synchronized, although working at different rates (see Fig. 2.2). In other words, the gyroscope takes measurements at δt periods, where $K\delta t = \Delta t$, $K \in \mathbb{N}$, and at t_j both gyroscope and odometer measurements are available. To estimate (2.17), we employ a "*double-clock*" Kalman filter which utilizes different propagation and update equations when only the gyroscope clock ticks (e.g., at time instant t_k when $t_j < t_k < t_{j+1}$) compared to when both gyroscope and odometer clocks tick simultaneously (e.g., at time instant $t_k = t_{j+1}$). In the following we describe each of these two instances.

- **Only gyroscope's clock ticks ($t_j < t_k < t_{j+1}$):** Considering that θ_j is constant during this time interval and applying the expectation operator to (2.3) and (2.16), we obtain the following state estimate propagation equation:

$$\hat{\mathbf{x}}^j_{k+1|k} = \mathbf{F}\hat{\mathbf{x}}^j_{k|k}, \quad \mathbf{F} \triangleq \begin{bmatrix} 1 & \delta t & \mathbf{0}_{1\times 7} \\ 0 & 1 & \mathbf{0}_{1\times 7} \\ \mathbf{0}_{7\times 1} & \mathbf{0}_{7\times 1} & \mathbf{I}_7 \end{bmatrix}. \tag{2.18}$$

The propagation equation for the covariance of the estimated state is:

$$\mathbf{P}^j_{k+1|k} = \mathbf{F}\mathbf{P}^j_{k|k}\mathbf{F}^T + \mathbf{Q} \tag{2.19a}$$

$$\mathbf{Q} = \mathbf{Diag}(0, \sigma_u^2, \mathbf{0}_{3\times 3}, \sigma_b^2\mathbf{I}_3, 0) \tag{2.19b}$$

where $\mathbf{P}^{j}_{\ell|k}$ is the covariance of the estimated state $\hat{\mathbf{x}}^{j}_{\ell|k}$. The propagated state estimate $\hat{\mathbf{x}}^{j}_{k+1|k}$ and its covariance $\mathbf{P}^{j}_{k+1|k}$, are updated at each step using the gyroscope measurements. This is readily done by applying the standard EKF update equations, using the gyroscope's measurement model [see (2.4)] and its Jacobian [see (2.15)] after padding it with zeros to reflect the state augmentation:

$$\mathbf{h}_g(\mathbf{x}) = {}^{I}\boldsymbol{\omega}_m(t) = \mathbf{c}\,\omega(t) + \mathbf{b}(t) + \mathbf{n}_g(t), \tag{2.20a}$$

$$\mathbf{H}_g = \nabla_{\mathbf{x}}\mathbf{h}_g = [\mathbf{0}_{3\times 1} \quad \mathbf{c} \quad \omega\mathbf{I}_3 \quad \mathbf{I}_3 \quad \mathbf{0}_{3\times 1}]. \tag{2.20b}$$

Additionally, to ensure that the vector $\mathbf{c}$ has unit norm, the following inferred measurement is employed to update the state and covariance estimate:

$$h_1(\mathbf{x}) = \mathbf{c}^T\mathbf{c} - 1 = 0, \tag{2.21}$$

$$\mathbf{h}_1 = \nabla_{\mathbf{x}} h_1 = [\mathbf{0}_{1\times 2} \quad 2\mathbf{c}^T \quad \mathbf{0}_{1\times 4}]. \tag{2.22}$$

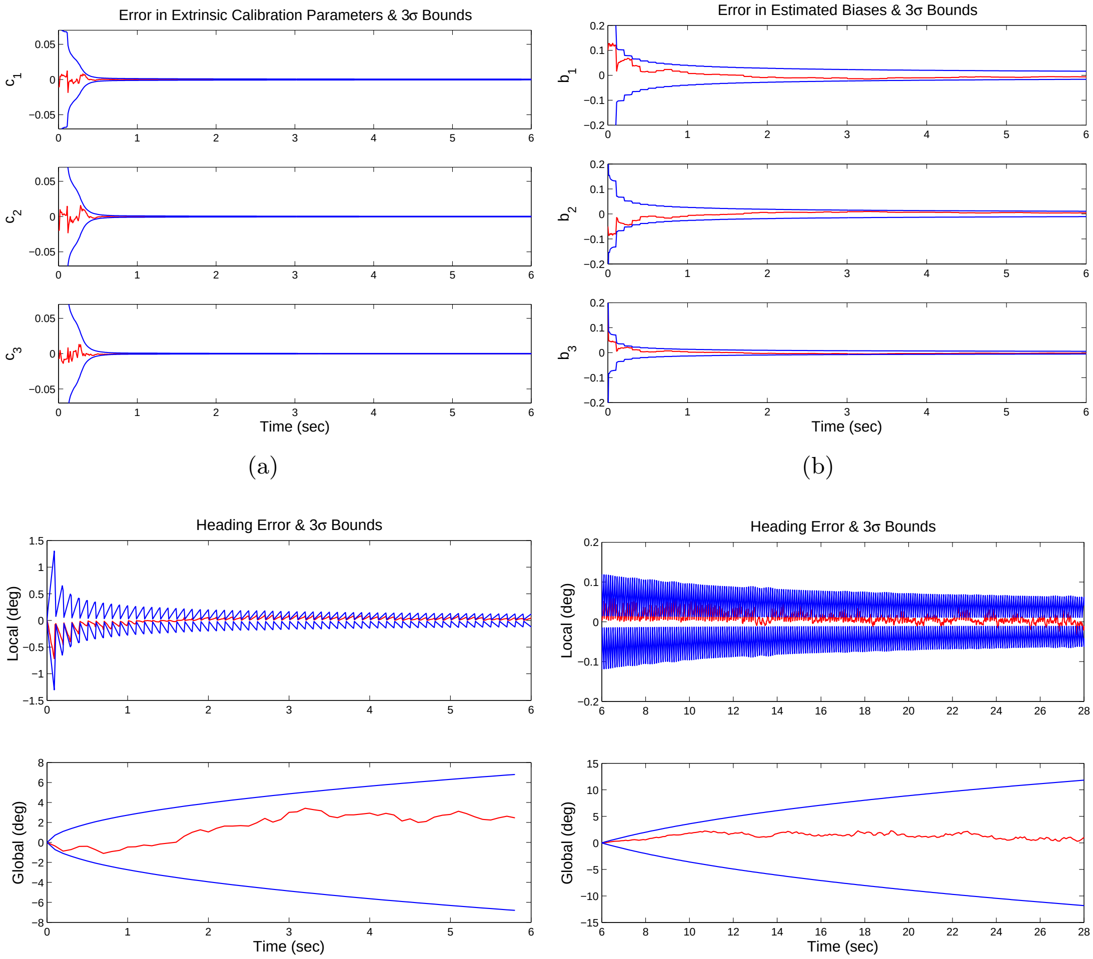
Similar to $\mathbf{h}_g$, this inferred measurement is applied to the filter using the standard EKF update equations.

- **Both gyroscope's and odometer's clocks tick ($t_k = t_{j+1}$):** In this case, we first propagate the state vector, and update it using the gyroscope measurements as described above. Next, we employ the standard Kalman filtering equations to update the state vector using the odometer's measurement of the change in the robot's orientation. The measurement model in this case is:

$$h_o(\mathbf{x}) = \varphi(t_j, t_{j+1}) + n_\varphi \tag{2.23}$$

where n_φ is white Gaussian noise with standard deviation σ_φ. In fact, this is the time instant when the gyroscope and odometer measurements are actually fused, resulting in corrections of the state vector [see (2.2)], and in particular of the extrinsic calibration parameters and the gyroscope biases.

In the next step, we need to marginalize out $\varphi(t_j, t_{j+1})$, and start a new variable $\varphi(t_{j+1}, t)$ to track the robot's local heading during the following time interval $[t_{j+1}, t_{j+2}]$. However, before doing so, we need to update the global heading of the robot as $\theta_{j+1} = \theta_j + \varphi(t_j, t_{j+1})$. These two steps are performed by the following



(a)

(b)

(c)

(d)

Figure 2.3: Time evolution of the 3σ uncertainty bounds and errors in the estimated (a) calibration parameters, (b) gyroscope biases, (c) heading in calibration phase, and (b) heading in localization phase.

linear transformation of the state estimate and its covariance matrix:

$$\hat{\mathbf{x}}^{j+1} = \mathbf{T}\hat{\mathbf{x}}^{j}, \quad \mathbf{T} \triangleq \begin{bmatrix} 0 & \mathbf{0}_{1\times 7} & 0 \\ \mathbf{0}_{7\times 1} & \mathbf{I}_{7\times 7} & \mathbf{0}_{7\times 1} \\ 1 & \mathbf{0}_{1\times 7} & 1 \end{bmatrix} \tag{2.24}$$

$$\mathbf{P}^{j+1} = \mathbf{T}\mathbf{P}^{j}\mathbf{T}^{T}. \tag{2.25}$$

The state vector $\hat{\mathbf{x}}^{j+1}$ is then used to fuse gyroscope and odometer measurements during the next time interval, $[t_{j+1}, t_{j+2}]$.

In summary, this double-clock Kalman filter uses the gyroscope and odometer measurements to provide an estimate for the robot's heading, as well as the extrinsic calibration parameters and gyroscope biases. Note, however, that the analysis of Section 2.4 does not imply observability of the robot's heading. In fact, it is well-known that the robot's global pose (and thus heading) is unobservable in the absence of appropriate exteroceptive measurements [82].

2.5.1 Estimating the Robot's Position

Since the gyroscope does not provide any measurement of the robot's linear motion, we solely rely on the odometer measurements to estimate the robot's position. For this purpose, we model the robot's position at time $t \in [t_j, t_{j+1}]$ as:

$$\dot{x}(t) = v(t)\cos(\theta_j + \varphi(t_j, t)) \tag{2.26a}$$

$$\dot{y}(t) = v(t)\sin(\theta_j + \varphi(t_j, t)) \tag{2.26b}$$

where $v(t)$ is the linear velocity of the robot. Assuming the noise in the two wheel encoders of the odometer are i.i.d., it can be easily verified that the measurement of linear and rotational velocities are uncorrelated. Therefore, the position of the robot can be estimated (through dead-reckoning) without impacting the robot's heading estimates. Specifically, the position estimates can be easily obtained by integrating (2.26) using odometer measurements of $v(t)$, and the latest robot's global heading estimate $\theta_j + \varphi(t_j, t)$. Clearly, in the absence of exteroceptive measurements, the robot's position is unobservable, and the uncertainty of its estimates will grow unbounded unless exteroceptive measurements of landmarks are used.

2.5.2 Fault Detection

In addition to providing improved accuracy, using a gyroscope in conjunction with the robot's odometer enables us to detect odometry faults such as slippage and stalling. Furthermore, while avoiding the integration of faulty odometer measurements, the gyroscope continues to provide a reliable source for updating heading estimates until the odometer functions properly. The detection of odometer faults is incorporated into the algorithm by examining the Mahalanobis distance of the acquired vs. expected measurements based on (2.23). Specifically, before updating the state vector using odometer measurements, we compute the following Mahalanobis distance:

$$\chi^2 = \frac{(\varphi_m(t_j, t_{j+1}) - h_o(\hat{\mathbf{x}}^j))^2}{\mathbf{h}_o \mathbf{P}^j \mathbf{h}_o^T + \sigma_\varphi^2} \tag{2.27}$$

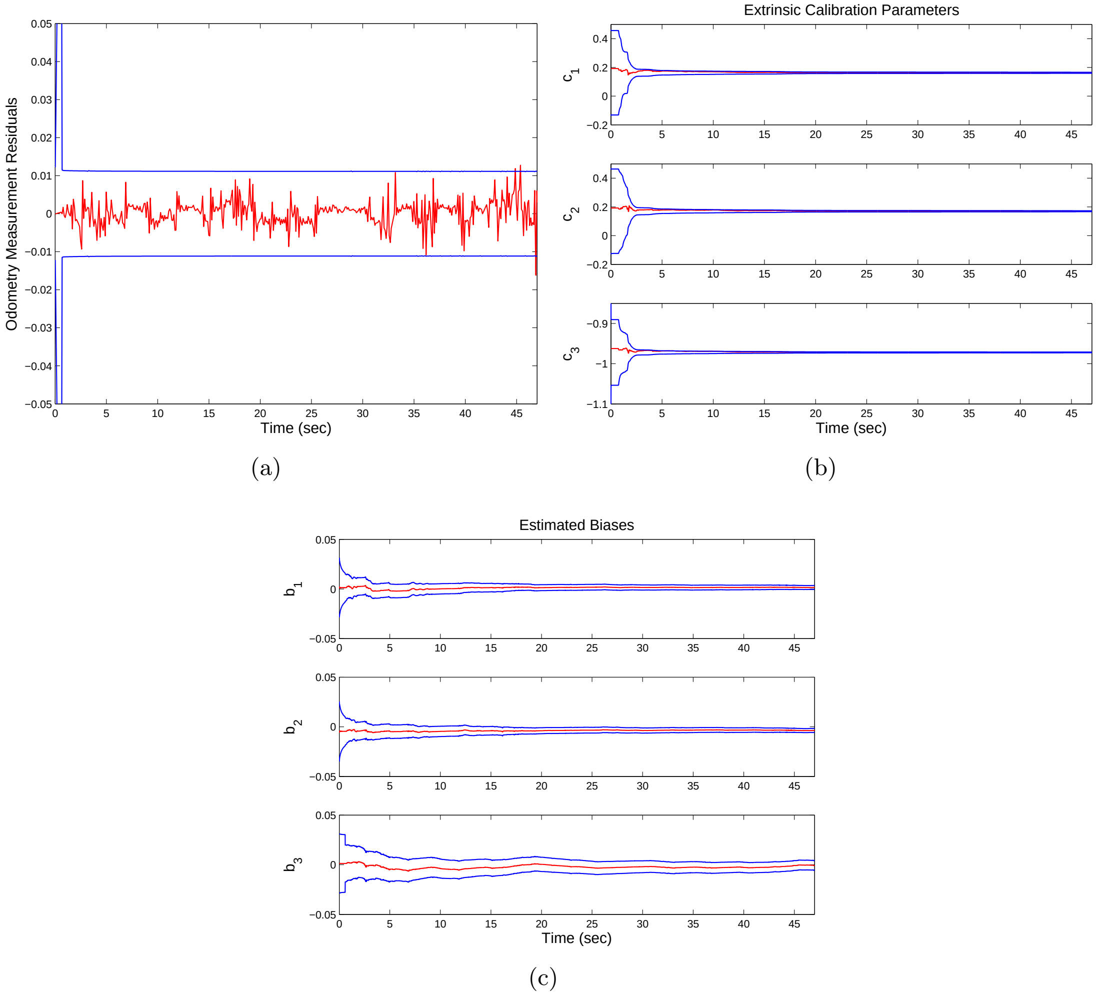
where $\varphi_m(t_j, t_{j+1})$ is the odometer measurement at time instant k, $\mathbf{h}_o = [1 \;\; \mathbf{0}_{1\times 8}]$, and $h_o(\hat{\mathbf{x}}^j) = \hat{\varphi}(t_j, t_{j+1})$ is the filter's estimate of the change in the robot's heading from t_j to t_{j+1}. Comparing this distance with a statistically selected threshold using the first-order chi-square distribution enables the algorithm to detect and discard outliers and faulty odometer measurements.

2.6 Simulations and Experiments

2.6.1 Simulation Results

In order to validate the proposed algorithm for simultaneous calibration and data fusion of gyroscope-odometer sensor pairs when ground truth is available, we have performed a number of simulation tests. In our simulation setup, a mobile robot moves randomly while the odometer and triaxial gyroscope sense the motion at 10 Hz and 100 Hz, respectively. Odometer measurements are perturbed with white Gaussian noise with standard deviation of $3 \frac{\circ}{\text{sec}}$. The measurement and bias driving noise of the gyroscope are selected as white Gaussian with standard deviation $5 \times 10^{-4} \; \frac{\text{rad/sec}}{\sqrt{\text{Hz}}}$ and $1 \times 10^{-5} \; \frac{\text{rad/sec}^2}{\sqrt{\text{Hz}}}$. The calibration parameter, $\mathbf{c}$ is set to $[0.80 \quad -0.53 \quad 0.27]^T$ (normalized to one), while the initial gyroscope biases are chosen randomly.

In the beginning, the state vector and its covariance are initialized with zero for the robot's heading and rotational velocity. The initial calibration estimates and gyroscope biases are selected based on their true values perturbed by Gaussian noise with $\sigma = 0.1$. The corresponding values in the state covariance matrix are set accordingly. Note that this large initial perturbation and uncertainty reflects our lack of confidence when we use hand-measurements to initialize the algorithm in practice. For the results presented in this section, we have run the algorithm for a total time of 28 s. The time evolution of the error in the estimated calibration parameters and biases along with their 3σ bounds for the first six seconds of the run are shown in Fig. 2.3. In addition to confirming the filter's consistency, theses figures corroborate the observability analysis of Section 2.4. In particular, the calibration parameter, $\mathbf{c}$, is estimated extremely accurately less than one second after the algorithm starts [Fig. 2.3(a)]. After six seconds, the uncertainty (1σ) in the estimated calibration parameters is $[2 \times 10^{-4} \;\; 1 \times 10^{-4} \;\; 7 \times 10^{-5}]^T$. Similarly, the



(a)

(b)

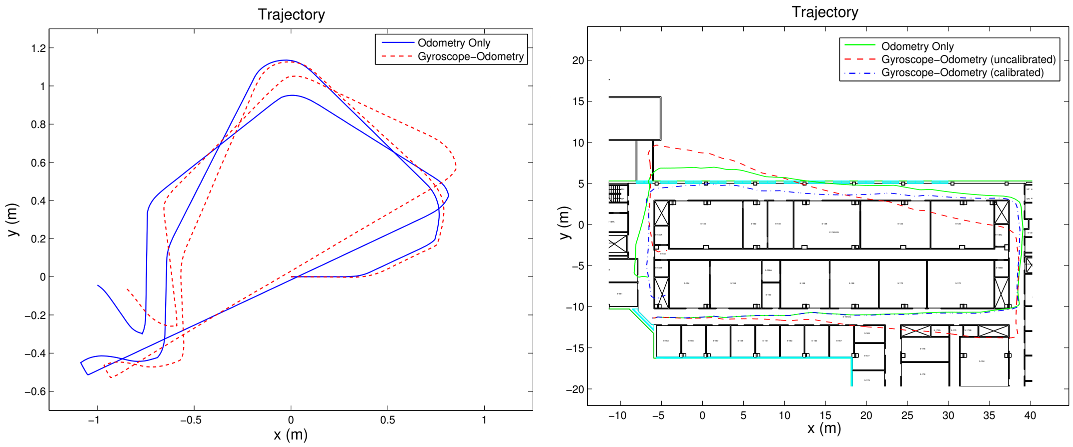
(c)

Figure 2.4: (a) In the absence of ground truth, the odometer measurement residuals and their estimated 3σ bounds were employed as indicators of consistency of the EKF; (b) Time evolution of the the estimated calibration parameters (red) and the 3σ uncertainty bounds centered around the final estimate (blue); (c) Time evolution of the estimated gyroscope biases (red) and the 3σ uncertainty bounds at each time instant (blue).

gyroscope biases are estimated with bounded accuracy [Fig. 2.3(b)]. Specifically, after six seconds the uncertainty in the estimated gyroscope biases is $[0.005 \;\; 0.004 \;\; 0.002]^T \, \frac{\text{rad}}{\text{sec}}$.

The estimated local and global heading for the first six seconds are shown in Fig. 2.3(c). The sawtooth pattern in the local orientation is due to the marginalization of $\phi(t_j, t_{j+1})$ after each odometry update. It can be seen that the uncertainty in the local robot's heading is fairly large in the beginning when the estimates of $\mathbf{c}$ and $\mathbf{b}$ are inaccurate. However, as the estimates of $\mathbf{c}$ and $\mathbf{b}$ become more accurate, the overall accuracy of the local heading estimates is improved. On the other hand, the global orientation, $\theta(t)$, is unobservable; thus its uncertainty grows unbounded with time, although its growth rate slows as the estimates of the calibration parameters and gyroscope biases become more accurate.

Due to the initial inaccuracy for the values of the calibration parameters and gyroscope biases, the estimate of the global heading quickly accumulates uncertainty. To address this issue, before actual deployment of the robot for localization, we perform a short calibration phase where the global orientation of the robot is not of importance. Once the calibration parameters and the gyroscope biases are estimated with sufficient accuracy, we reset the robot's orientation and start the main localization task, where the robot's heading and gyroscope biases are tracked with higher accuracy, while the estimates of the calibration parameters are refined. This is depicted in Fig. 2.3(d) which shows the last 22 s of the above simulation run after the global orientation is reset. As expected, the uncertainty in the robot's global heading increases at a much slower pace compared to the calibration phase due to the availability of high-precision estimates for the gyroscope biases and calibration parameters.



(a) (b)

Figure 2.5: (a) The robot's trajectory during the initial calibration phase; (b) The robot's estimated trajectory during the main localization phase.

2.6.2 Experimental Results

In order to demonstrate the validity of our calibration and data fusion algorithm in a real situation, we have conducted an experiment using a Pioneer 3 robot with odometer sampling rate of 10 Hz, and an ISIS IMU with sampling rate of 100 Hz. The IMU was rigidly mounted on the robot in a tilted configuration. We initialized the calibration parameters to $[0.1 \;\; 0.1 \;\; -0.8]^T$ (normalized to one), obtained from the technical drawings. The initial values for the gyroscope biases were obtained by averaging the gyroscope measurements when the robot was static. The initial uncertainty (1σ) of the calibration parameters and the biases were set to 0.1 and 0.01, respectively. We then ran the algorithm for 47 s while the robot followed a random trajectory in a 2×2 m arena [see Fig. 2.5(a)]. In the absence of ground truth, we employed the odometer measurement residuals as an indicator of the filter's performance and consistency [see Fig. 2.4(a)].

The time evolution of the estimated calibration parameters as well as their 3σ confidence bounds are shown in Fig. 2.4(b). Note that since the ground truth for these parameters is not available, we have centered the bounds around the final estimated values. This figure clearly demonstrates that the estimates of the calibration parameters quickly converge and remain constant. In particular, note that the estimated calibration parameters and their 3σ uncertainty bounds at the end of the experiment were $[0.16 \quad 0.17 \quad -0.97] \pm [0.0033 \quad 0.0033 \quad 0.008]^T$, which indicates about two orders of magnitude improvement in accuracy. The time evolution of the gyroscope bias estimates and their uncertainties are depicted in Fig. 2.4(c). Note that, unlike the calibration parameters, the gyroscope biases are time-varying; thus their uncertainty bounds are centered around the estimates at each time instant.

Finally, using the estimated calibration parameters, we performed a localization experiment where the robot traveled more than 120 meters in about 6 minutes. The estimated trajectory of the robot in this experiment is shown in Fig. 2.5(b) along with the odometry-only trajectory. This figure also shows the uncalibrated gyroscope-odometer trajectory when an inaccurate transformation between the gyroscope and odometer is used. Clearly, the estimated trajectory when using calibrated gyroscope-odometer is more accurate than the others. However, due to the unobservability of the robot's pose in the absence of exteroceptive measurements, the error in the estimated trajectory grows unbounded with time. The remedy for this issue is to use appropriate exteroceptive sensors in conjunction with the proposed gyroscope-odometer system so as to guarantee observability of the robot's pose and bound its uncertainty.

2.7 Summary

In this chapter, we presented a new EKF-based method for simultaneous data fusion and extrinsic calibration of gyroscope-odometer pairs used to track the heading of a

2D robot. The proposed method accounts for gyroscope biases and properly handles the different sampling mechanisms of the gyroscope and odometer. We analytically proved that the gyroscope-odometer system is locally observable, thus allowing accurate estimation of the unknown extrinsic calibration parameters and gyroscope biases. We presented results from simulations and real experiments, confirming the validity of the described algorithm.

Chapter 3

IMU-Camera Calibration[†]

3.1 Introduction

In recent years, Inertial Navigation Systems (INS) have been widely used for estimating the motion of vehicles moving in a 3-dimensional space such as airplanes, helicopters, automobiles, etc [25]. At the core of most INS lies an Inertial Measurement Unit (IMU) that measures linear accelerations and rotational velocities. By integrating these signals in real time, an INS is capable of tracking the position, velocity, and attitude of a vehicle. This deadreckoning process, however, cannot be used over extended periods of time because the errors in the computed estimates continuously increase. This is due to the noise and biases present in the inertial measurements. For this reason, current INS rely on the Global Positioning System (GPS) in order to receive periodic corrections. In most cases, a Kalman filter estimator is used for optimally combining the IMU and GPS measurements [123]. One of the main limitations of the *GPS-aided INS* configuration is that it cannot be used when the GPS signals are not available (e.g., indoors, underground, underwater, in space, etc), or their reliability is limited (e.g., in the vicinity of tall buildings and structures due to specular reflections and multipath error). Furthermore, high-precision GPS receivers are prohibitively expensive, and often the acquired level of accuracy is not sufficient for certain applications (e.g., parallel parking a car within a tight space).

An alternative approach to provide corrections to an INS is via the use of visual sensors such as cameras. Cameras are small-size, light-weight, passive sensors that provide rich information for the surroundings of a vehicle at low cost. When observing

[†] This work is partially appeared at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Diego, CA, 2007 [90], and at the IEEE Transaction on Robotics, 2008 [93].

a known scene, both the position and attitude of the camera can be computed [3]. Furthermore, by tracking visual features through sequences of images, the motion of the camera can be estimated [9, 94]. Cameras and IMUs are complementary in terms of accuracy and frequency response. An IMU is ideal for tracking the state of a vehicle over short periods of time when it undergoes motions with high dynamic profile. On the other hand, a camera is best suited for state estimation over longer periods of time and for smoother motion profiles. Combining these two sensors to form a *Vision-aided INS* (V-INS) has recently become a popular topic of research [36].

In order to fuse measurements from an IMU and a camera in a V-INS, the 6 degrees of freedom (6-d.o.f.) *transformation* between these two devices must be known precisely (see Fig. 3.1). Inaccuracies in the values of the IMU-camera relative pose (position and attitude), will appear as biases that will reduce the accuracy of the estimation process or even cause the estimator to diverge. In most cases in practice, this unknown transformation is computed manually (e.g., from CAD plots) or through the use of additional sensors. For example, for the Mars Exploration Rover (MER) mission [62], a high precision 3D laser scanner was employed to measure the location of the 4 corners of the IMU housing with respect to a checker-board placed in front of the camera for calibration purposes. Although this method achieved sub-degree relative attitude accuracy and less than 1 cm relative position error [61], it is prohibitive for many applications due to the high cost of the equipment (3D laser scanner) involved. Additionally, every time one of the two sensors is removed (e.g., for service) and repositioned, the same process needs to be repeated, which requires significant time and effort. Automating this procedure will reduce the cost of deploying a V-INS, increase the accuracy of the computed state estimates during regular operation, and minimize the probability of failure due to bias-induced divergence.

In this chapter, we present an Extended Kalman Filter (EKF)-based algorithm for determining the 6 d.o.f. transformation between a single camera and an IMU using measurements only from these two sensors [90]. Contrary to existing approaches [75, 71] that rely on modified hand-eye calibration processes (e.g., [131, 27, 31]), our method takes into account the *time correlation* of the IMU measurements by explicitly modeling them using an augmented-state EKF [83, 43]. Additionally, our algorithm computes the uncertainty in the estimated quantities, or equivalently, the *covariance*. Furthermore, we do *not* separate the task of translation estimation from rotation estimation which prevents potential error propagation. Moreover, unlike existing approaches, the described method does *not* require any special testbed except a calibration pattern which is also needed for estimating the intrinsic parameters of the camera. Therefore it offers the inherent capability of re-calibrating the IMU-camera system as frequently as needed. Finally, a comprehensive observability analysis based on Lie derivatives [51, 111] is performed to ensure that the sensor measurements provide sufficient information for accurately estimating the IMU-camera transformation.

The rest of this chapter is structured as follows: Section 3.2 provides an overview of the related literature. Section 3.3 presents the proposed EKF-based algorithm, and Section 3.4 investigates the observability of the nonlinear system describing the IMU-camera calibration process. Simulation and experimental results are provided in Section 3.5, and finally, Section 3.6 summarizes the chapter.

3.2 Related Work

A well-known related process is the hand-eye calibration [131], whose objective is to estimate the 6 d.o.f. transformation between a camera and a robot manipulator. Recently, there have been some attempts to modify existing hand-eye calibration algorithms to determine the IMU-camera alignment [75, 71]. Specifically, in [71] the rotation part of the hand-eye equation is solved using nonlinear optimization software under the assumption that the translation between the IMU and the camera is negligible. However, in most realistic situations this assumption is *not* valid and ignoring the translation introduces biases in estimation algorithms using these alignment parameters.

A different approach to this problem is proposed by Lobo and Dias in [75, 76]. First, they obtain the vertical direction of the IMU and the camera frames by measuring the direction of gravity while viewing a vertically-installed calibration pattern. Then, using Horn's method [55], they estimate the rotation between the IMU and the camera. Finally, they use a spin table to rotate the system about the IMU's center and zero-out the linear acceleration of the IMU due to rotation. This process allows one to compute the translation between the camera and the IMU based only on the camera measurements. The main drawback of this approach is that it ignores the *time correlation* between the inertial measurements due to the IMU biases. Additionally, it does *not* provide any figure of merit of the achieved level of accuracy (e.g., covariance of the estimated quantities). Furthermore this two-stage process *decouples* the computation of rotation and translation, and hence allows error propagation from the rotation estimates to the translation estimates. Finally this method requires fine adjustment of the IMU-camera system on a *spin table* which limits its applicability when re-calibration is frequently

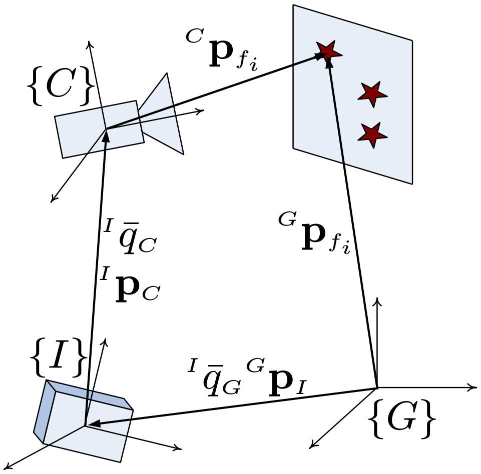


Figure 3.1: The geometric relation between the known landmarks f_i and the camera, $\{C\}$, IMU, $\{I\}$, and global, $\{G\}$, frames of reference. The unknown IMU-camera transformation is denoted by the position and quaternion pair $({}^{I}\mathbf{p}_{C},\ {}^{I}\bar{q}_{C})$. This transformation is determined using estimates of the IMU motion, $({}^{G}\mathbf{p}_{I},\ {}^{I}\bar{q}_{G})$, the projections of the landmarks' positions, ${}^{C}\mathbf{p}_{f_i}$, on the camera frame (image observations), and the known positions of the landmarks, ${}^{G}\mathbf{p}_{f_i}$, expressed in the global frame of reference.

needed.

The IMU-camera and hand-eye calibration problems require separate treatments due to the different *noise characteristics* of the IMU and shaft-encoder signals. Specifically, while the shaft-encoder measurements at different time instants are uncorrelated, consecutive IMU measurements are not. This is due to the IMU *biases*. Ignoring the time correlation of the inertial measurements limits the accuracy of the IMU-camera calibration process and can lead to inconsistent estimates.

To the best of our knowledge the EKF-based algorithm presented in this chapter is the first approach to the IMU-camera calibration problem that does *not* ignore the correlations between the IMU measurements and requires no specialized hardware. Furthermore, the uncertainty in the estimated alignment parameters is provided at every time step by computing their covariance. Finally, it is shown that it suffices to rotate the camera in place in order for these parameters to become observable.

Recently, Kelly and Sukhatme have proposed an extension to the work presented here that relaxes the assumption of having globally known landmarks, by performing Simultaneous Localization and Mapping (SLAM) [65]. Furthermore, a deterministic nonlinear observer for the IMU-camera relative rotation is proposed in [112] that is guaranteed to be exponentially stable under certain observability conditions. This is in contrast with the EKF that does not provide any stability guarantees.

3.3 Description of the Algorithm

The IMU-camera calibration is achieved through a two-step process. First, camera images are processed in a *batch* algorithm to compute an *initial estimate* for the camera pose. Additionally, the approximate value of the unknown transformation (e.g., hand-measured or from CAD plots) is combined with the camera-pose estimate to compute an *initial* estimate for the IMU pose (Section 3.3.1). In the next step, both these estimates are used to initialize the corresponding variables in the EKF estimator. By *sequentially* processing additional measurements from the camera and the IMU, the EKF is able to *refine* the initial estimate for the unknown transformation, while simultaneously tracking the position, velocity, and attitude of the two sensors (Sections 3.3.2 – 3.3.5).

3.3.1 Filter Initialization

The purpose of this process is to determine the initial estimate for the *IMU pose* (${}^G\mathbf{p}_I,\ {}^I\bar{q}_G$) where ${}^G\mathbf{p}_I$ denotes the position of the IMU with respect to the global frame of reference, and ${}^I\bar{q}_G$ is the rotation quaternion between the IMU and the global frames.

We first compute an estimate for the *camera pose* (${}^G\mathbf{p}_C,\ {}^C\bar{q}_G$) using visual features (corners of the squares in the calibration pattern) whose positions, ${}^G\mathbf{p}_{f_i}$, are known in global coordinates. Specifically, the initial estimates of the depth to these features are computed using Ansar's method [3], while the initial estimate for the camera pose is determined by employing Horn's method [55]. Finally, a least-squares algorithm refines the camera-pose estimate and additionally computes its covariance [47].

In the next step of the initialization process, we use an approximate estimate for the unknown IMU-camera transformation (${}^I\mathbf{p}_C,\ {}^I\bar{q}_C$). This was determined manually in our case but it can also be found using the CAD plots showing the IMU-camera placement. We should note, that the requirement for an approximate estimate for the initial IMU-camera transformation is not limiting, since it can also be determined by employing any hand-eye calibration algorithm. An initial estimate for the IMU pose is then computed from the following relations (see Fig. 3.1):

$$\begin{aligned} {}^G\mathbf{p}_I &= {}^G\mathbf{p}_C - \mathbf{C}^T({}^C\bar{q}_G)\mathbf{C}^T({}^I\bar{q}_C){}^I\mathbf{p}_C \\ {}^I\bar{q}_G &= {}^I\bar{q}_C \otimes {}^C\bar{q}_G \end{aligned} \tag{3.1}$$

where $\mathbf{C}(\bar{q})$ is the rotational matrix corresponding to quaternion $\bar{q}$, and $\otimes$ denotes quaternion multiplication. Finally, after computing the corresponding Jacobians [by linearizing (3.1)] and considering the uncertainty (covariance) in the estimates of (${}^I\mathbf{p}_C,\ {}^I\bar{q}_C$) and (${}^G\mathbf{p}_C,\ {}^C\bar{q}_G$), the covariance of the initial IMU pose estimate is readily found.

3.3.2 Filter Propagation

The EKF estimates the IMU pose and linear velocity as well as the unknown transformation (rotation and translation) between the camera and the IMU. Additionally, the filter estimates the biases in the IMU signals.

Continuous-time system model

We first derive the linearized continuous-time system model that describes the time evolution of the errors in the state estimates. Discretization of this model will allow us to employ the sampled measurements of the IMU for state propagation. The filter state is described by the vector:

$$\mathbf{x} = \begin{bmatrix} {}^I\bar{q}_G^T & \mathbf{b}_g^T & {}^G\mathbf{v}_I^T & \mathbf{b}_a^T & {}^G\mathbf{p}_I^T & {}^I\bar{q}_C^T & {}^I\mathbf{p}_C^T \end{bmatrix}^T \tag{3.2}$$

where ${}^I\bar{q}_G(t)$ and ${}^I\bar{q}_C(t)$ are the quaternions which represent the orientation of the *global frame* and the *camera frame* in the IMU frame, respectively. The position and velocity of the IMU in the global frame are denoted by ${}^G\mathbf{p}_I(t)$ and ${}^G\mathbf{v}_I(t)$. ${}^I\mathbf{p}_C(t)$ is the position of the camera in the IMU frame, and $\mathbf{b}_g$, $\mathbf{b}_a$ are the 3×1 bias vectors affecting the gyroscope and accelerometer measurements, respectively. These biases are typically present in the signals of inertial sensors, and need to be modeled and estimated, in order to attain accurate state estimates. In our work the IMU biases are modeled as random walk processes driven by the zero-mean white Gaussian noise vectors $\mathbf{n}_{wg}$ and $\mathbf{n}_{wa}$, respectively.

The system model describing the time evolution of the IMU state and of the IMU-camera transformation is given by the following equations [73, 124]:

$$\begin{aligned}
{}^I\dot{\bar{q}}_G(t) &= \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}(t))\,{}^I\bar{q}_G(t) && (3.3)\\
{}^G\dot{\mathbf{p}}_I(t) &= {}^G\mathbf{v}_I(t) \quad , \quad {}^G\dot{\mathbf{v}}_I(t) = {}^G\mathbf{a}(t) && (3.4)\\
\dot{\mathbf{b}}_g(t) &= \mathbf{n}_{wg}(t) \quad , \quad \dot{\mathbf{b}}_a(t) = \mathbf{n}_{wa}(t) && (3.5)\\
{}^I\dot{\bar{q}}_C(t) &= \mathbf{0}_{3\times 1} \quad , \quad {}^I\dot{\mathbf{p}}_C(t) = \mathbf{0}_{3\times 1} && (3.6)
\end{aligned}$$

In these expressions $\boldsymbol{\omega}(t) = [\omega_x \ \ \omega_y \ \ \omega_z]^T$ is the rotational velocity of the IMU, expressed in the IMU frame, and

$$\mathbf{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -\lfloor \boldsymbol{\omega} \times \rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix}, \lfloor \boldsymbol{\omega} \times \rfloor = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

Finally, ${}^G\mathbf{a}$ is the acceleration of the IMU, expressed in the global frame.

The gyroscope and accelerometer measurements, $\boldsymbol{\omega}_m$, and $\mathbf{a}_m$ respectively, which are employed for state propagation, are modeled as

$$\boldsymbol{\omega}_m(t) = \boldsymbol{\omega}(t) + \mathbf{b}_g(t) + \mathbf{n}_g(t) \tag{3.7}$$

$$\mathbf{a}_m(t) = \mathbf{C}({}^I\bar{q}_G(t))({}^G\mathbf{a}(t) - {}^G\mathbf{g}) + \mathbf{b}_a(t) + \mathbf{n}_a(t) \tag{3.8}$$

where $\mathbf{n}_g$ and $\mathbf{n}_a$ are zero-mean, white Gaussian noise processes, and ${}^G\mathbf{g}$ is the gravitational acceleration.

By applying the expectation operator on both sides of (3.3)-(3.6), we obtain the state estimates' propagation equations:

$${}^I\dot{\hat{\bar{q}}}_G(t) = \frac{1}{2}\mathbf{\Omega}(\hat{\boldsymbol{\omega}}(t))\,{}^I\hat{\bar{q}}_G(t) \tag{3.9}$$

$${}^G\dot{\hat{\mathbf{p}}}_I(t) = {}^G\hat{\mathbf{v}}_I(t), \quad {}^G\dot{\hat{\mathbf{v}}}_I(t) = \mathbf{C}^T({}^I\hat{\bar{q}}_G(t))\hat{\mathbf{a}}(t) + {}^G\mathbf{g} \tag{3.10}$$

$$\dot{\hat{\mathbf{b}}}_g(t) = \mathbf{0}_{3\times 1}\ , \quad \dot{\hat{\mathbf{b}}}_a(t) = \mathbf{0}_{3\times 1} \tag{3.11}$$

$${}^I\dot{\hat{\bar{q}}}_C(t) = \mathbf{0}_{3\times 1}\ , \quad {}^I\dot{\hat{\mathbf{p}}}_C(t) = \mathbf{0}_{3\times 1} \tag{3.12}$$

with

$$\hat{\mathbf{a}}(t) = \mathbf{a}_m(t) - \hat{\mathbf{b}}_a(t), \quad \text{and} \quad \hat{\boldsymbol{\omega}}(t) = \boldsymbol{\omega}_m(t) - \hat{\mathbf{b}}_g(t) \tag{3.13}$$

The 21×1 filter error-state vector is defined as:

$$\widetilde{\mathbf{x}} = \begin{bmatrix} {}^I\boldsymbol{\delta\theta}_G^T & \widetilde{\mathbf{b}}_g^T & {}^G\widetilde{\mathbf{v}}_I^T & \widetilde{\mathbf{b}}_a^T & {}^G\widetilde{\mathbf{p}}_I^T & {}^I\boldsymbol{\delta\theta}_C^T & {}^I\widetilde{\mathbf{p}}_C^T \end{bmatrix}^T \tag{3.14}$$

For the IMU and camera positions, and the IMU velocity and biases, the standard additive error definition is used (i.e., the error in the estimate $\hat{x}$ of a quantity x is $\widetilde{x} = x - \hat{x}$). However, for the quaternions a different error definition is employed. In particular, if $\hat{\bar{q}}$ is is the estimated value of the quaternion $\bar{q}$, then the attitude error is described by the *error quaternion*:

$$\delta\bar{q} = \bar{q} \otimes \hat{\bar{q}}^{-1} \simeq \begin{bmatrix} \frac{1}{2}\boldsymbol{\delta\theta}^T & 1 \end{bmatrix}^T \tag{3.15}$$

Intuitively, the quaternion $\delta\bar{q}$ describes the (small) rotation that causes the true and estimated attitude to coincide. The main advantage of this error definition is that it allows us to represent the attitude uncertainty by the 3×3 covariance matrix $E\{\boldsymbol{\delta\theta\delta\theta}^T\}$. Since the attitude corresponds to 3 degrees of freedom, this is a minimal representation.

The linearized continuous-time error-state equation is:

$$\dot{\widetilde{\mathbf{x}}} \;=\; \mathbf{F}_c\widetilde{\mathbf{x}} + \mathbf{G}_c\mathbf{n}, \tag{3.16}$$

where

$$\mathbf{F}_c = \begin{bmatrix} -\lfloor\hat{\boldsymbol{\omega}}\times\rfloor & -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times9} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times9} \\ -\mathbf{C}^T({}^I\hat{\bar{q}}_G)\lfloor\hat{\mathbf{a}}\times\rfloor & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{C}^T({}^I\hat{\bar{q}}_G) & \mathbf{0}_{3\times9} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times9} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times9} \\ \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} & \mathbf{0}_{6\times9} \end{bmatrix}$$

$$\mathbf{G}_c = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{C}^T({}^I\hat{\bar{q}}_G) & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_3 \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} \end{bmatrix}, \quad \mathbf{n} = \begin{bmatrix} \mathbf{n}_g \\ \mathbf{n}_{wg} \\ \mathbf{n}_a \\ \mathbf{n}_{wa} \end{bmatrix}$$

and $\mathbf{I}_3$ is the 3×3 identity matrix. The covariance, $\mathbf{Q}_c$, of the system noise depends on the IMU noise characteristics and is computed off-line according to [58, 57].

Discrete-time implementation

The IMU signals $\boldsymbol{\omega}_m$ and $\mathbf{a}_m$ are sampled at 100 Hz (i.e., T=0.01 sec). Every time a new IMU measurement is received, the state estimate is propagated using 4th-order Runge-Kutta numerical integration of (3.9)-(3.12). In order to derive the covariance propagation equation, we evaluate the discrete-time state transition matrix:

$$\boldsymbol{\Phi}_k = \boldsymbol{\Phi}(t_k + T, t_k) = \exp\left(\int_{t_k}^{t_k+T} \mathbf{F}_c(\tau)\mathrm{d}\tau\right) \tag{3.17}$$

and the discrete-time system noise covariance matrix:

$$\mathbf{Q}_d = \int_{t_k}^{t_k+T} \boldsymbol{\Phi}(t_{k+1}, \tau)\mathbf{G}_c\mathbf{Q}_c\mathbf{G}_c^T\boldsymbol{\Phi}^T(t_{k+1}, \tau)\mathrm{d}\tau \tag{3.18}$$

The propagated covariance is then computed as:

$$\mathbf{P}_{k+1|k} = \boldsymbol{\Phi}_k\mathbf{P}_{k|k}\boldsymbol{\Phi}_k^T + \mathbf{Q}_d$$

3.3.3 Measurement Model

The IMU-camera moves continuously and records images of a calibration pattern. These are then processed to detect and identify point features whose positions, ${}^G\mathbf{p}_{f_i}$, are known with respect to the global frame of reference (centered and aligned with the checkerboard pattern of the calibration target). Once this process is completed for each image, a list of point features along with their measured image coordinates, $(u_i,\ v_i)$, is provided to the EKF, which uses them to update the state estimates.

The projective camera measurement model employed is:

$$\mathbf{z}_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} + \boldsymbol{\eta}_i = \begin{bmatrix} x_i/z_i \\ y_i/z_i \end{bmatrix} + \boldsymbol{\eta}_i = \mathbf{h}_i(\mathbf{x}, {}^G\mathbf{p}_{f_i}) + \boldsymbol{\eta}_i \tag{3.19}$$

where (see Fig. 3.1),

$${}^C\mathbf{p}_{f_i} = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \mathbf{C}({}^C\bar{q}_I)\mathbf{C}({}^I\bar{q}_G)\left({}^G\mathbf{p}_{f_i} - {}^G\mathbf{p}_I\right) - \mathbf{C}({}^C\bar{q}_I){}^I\mathbf{p}_C$$

and $\boldsymbol{\eta}_i$ is the feature-measurement noise with covariance $\mathbf{R}_i = \sigma_i^2\mathbf{I}_2$.

The measurement Jacobian matrix $\mathbf{H}_i$ is:

$$\mathbf{H}_i = \mathbf{J}_{cam}^i \begin{bmatrix} \mathbf{J}_{\theta_G}^i & \mathbf{0}_{3\times 9} & \mathbf{J}_{\mathbf{p}_I}^i & \mathbf{J}_{\theta_C}^i & \mathbf{J}_{\mathbf{p}_c}^i \end{bmatrix} \tag{3.20}$$

with

$$\mathbf{J}^i_{cam} = \frac{1}{\hat{z}_i^2}\begin{bmatrix}\hat{z}_i & 0 & -\hat{x}_i \\ 0 & \hat{z}_i & -\hat{y}_i\end{bmatrix} \tag{3.21}$$

$$\mathbf{J}^i_{\theta_G} = \mathbf{C}({}^C\hat{\bar{q}}_I)\lfloor \mathbf{C}({}^I\hat{\bar{q}}_G)({}^G\mathbf{p}_{f_i} - {}^G\hat{\mathbf{p}}_I)\times\rfloor$$

$$\mathbf{J}^i_{\theta_C} = -\mathbf{C}({}^C\hat{\bar{q}}_I)\lfloor \mathbf{C}({}^I\hat{\bar{q}}_G)({}^G\mathbf{p}_{f_i} - {}^G\hat{\mathbf{p}}_I) - {}^I\hat{\mathbf{p}}_C\times\rfloor$$

$$\mathbf{J}^i_{\mathbf{p}_I} = -\mathbf{C}({}^C\hat{\bar{q}}_I)\mathbf{C}({}^I\hat{\bar{q}}_G)\ ,\ \ \mathbf{J}^i_{\mathbf{p}_c} = -\mathbf{C}({}^C\hat{\bar{q}}_I)$$

$$\begin{bmatrix}\hat{x}_i \\ \hat{y}_i \\ \hat{z}_i\end{bmatrix} = \mathbf{C}({}^C\hat{\bar{q}}_I)\mathbf{C}({}^I\hat{\bar{q}}_G)\left({}^G\mathbf{p}_{f_i} - {}^G\hat{\mathbf{p}}_I\right) - \mathbf{C}({}^C\hat{\bar{q}}_I){}^I\hat{\mathbf{p}}_C$$

When observations to N features are available concurrently, we stack these in one measurement vector $\mathbf{z} = [\mathbf{z}_1^T \ \cdots \ \mathbf{z}_N^T]^T$ to form a single batch-form update equation. Similarly, the batch measurement Jacobian matrix is defined as $\mathbf{H} = [\mathbf{H}_1^T \ \cdots \ \mathbf{H}_N^T]^T$. Finally, the measurement residual is computed as:

$$\mathbf{r} \triangleq \mathbf{z} - \hat{\mathbf{z}} \simeq \mathbf{H}\widetilde{\mathbf{x}} + \boldsymbol{\eta}, \tag{3.22}$$

where $\boldsymbol{\eta} = [\boldsymbol{\eta}_1^T \ \cdots \ \boldsymbol{\eta}_N^T]^T$ is the measurement noise with covariance $\mathbf{R} = \mathbf{Diag}(\mathbf{R}_i), i = 1, \ldots, N$.

3.3.4 Iterated Extended Kalman Filter Update

In order to increase the accuracy and numerical stability in the face of the highly nonlinear measurement model, we employ the *Iterated Extended Kalman Filter* [84, 60] to update the state. The iterative scheme proceeds as follows:

At each iteration step j

1. Compute $\hat{\mathbf{z}}^j = E\{\mathbf{z}\}$ as a function of the current j-th iterate $\hat{\mathbf{x}}^j_{k+1|k+1}$ using the measurement function (3.19).

2. Evaluate the measurement Jacobian matrix $\mathbf{H}^j$ [see (3.20)] using the current iterate $\hat{\mathbf{x}}^j_{k+1|k+1}$.

3. Form the residual $\mathbf{r}^j = \mathbf{z} - \hat{\mathbf{z}}^j$, and compute its covariance $\mathbf{S}^j = \mathbf{H}^j\mathbf{P}_{k+1|k}\mathbf{H}^{jT} + \mathbf{R}$.

4. Using the Kalman gain $\mathbf{K}^j = \mathbf{P}_{k+1|k}\mathbf{H}^{j^T}(\mathbf{S}^j)^{-1}$ compute the correction

$$\Delta\mathbf{x}^j = \mathbf{K}^j(\mathbf{r}^j + \mathbf{H}^j\ \Delta\mathbf{x}^{j-1}) \tag{3.23}$$

with $\Delta\mathbf{x}^0 = \mathbf{0}_{21\times1}$, necessary for determining the next iterate of the updated state estimate $\hat{\mathbf{x}}^{j+1}_{k+1|k+1}$.

The iteration begins using the propagated state estimate $\hat{\mathbf{x}}^0_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k}$ as the zeroth iterate, which makes the first iteration equivalent to a regular EKF update. This process is repeated till the reduction in the cost function

$$J^j = \widetilde{\mathbf{x}}^{j^T}\mathbf{P}^{-1}_{k+1|k}\widetilde{\mathbf{x}}^j + \mathbf{r}^{j^T}\mathbf{R}^{-1}\mathbf{r}^j \tag{3.24}$$

with $\widetilde{\mathbf{x}}^j \triangleq \hat{\mathbf{x}}_{k+1|k} - \hat{\mathbf{x}}^j_{k+1|k+1}$, falls below the threshold $\tau = \max(0.01, 0.001 \times J^{j-1})$, or, when a maximum number of iterations is reached [105]. Finally, the covariance matrix for the current state is updated using the values for $\mathbf{K}$ and $\mathbf{S}$ from the last iteration:

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{K}\mathbf{S}\mathbf{K}^T \tag{3.25}$$

3.3.5 Outlier Rejection

Before using the detected features in the measurement update, we employ a Mahalanobis distance test to detect and reject mismatches or very noisy observations. Every time a new measurement becomes available, we compute the Mahalanobis distance:

$$\chi^2 = (\mathbf{z}_{i_k} - \hat{\mathbf{z}}_{i_k})^T\,\mathbf{S}_i^{-1}\,(\mathbf{z}_{i_k} - \hat{\mathbf{z}}_{i_k}) \tag{3.26}$$

In this equation, $\mathbf{z}_{i_k}$ is the measurement of i-th landmark at time-step k, $\hat{\mathbf{z}}_{i_k}$ is the expected measurement of the same landmark based on the latest state estimate, and $\mathbf{S}_i = \mathbf{H}_i\mathbf{P}_{k+1|k}\mathbf{H}_i + \mathbf{R}_i$ is the covariance of the corresponding measurement residual. A probabilistic threshold on χ^2 is used to specify whether the measurement is reliable or not. Measurements which pass this test are processed by the iterative update procedure as described above.

3.4 Observability Analysis

In this work for the first time, we study the observability of the nonlinear system describing the IMU-camera calibration process, and prove that the IMU-camera calibration system is *locally observable* when at least two rotations about different axes are performed. For a brief review of the Lie derivatives and the observability of nonlinear systems, we refer the reader to Section 2.4. First and in order to simplify the notation, we retain only few of the subscripts describing the variables in the system state vector [see (3.2)]:

$$\mathbf{x}(t) = \begin{bmatrix} \bar{q}_I^T & \mathbf{b}_g^T & \mathbf{v}^T & \mathbf{b}_a^T & \mathbf{p}_I^T & \bar{q}_C^T & \mathbf{p}_C^T \end{bmatrix}^T \tag{3.27}$$

Then, we rearrange the nonlinear kinematic equations (3.3)-(3.6) in a suitable format for computing the Lie derivatives:

$$\begin{bmatrix} \dot{\bar{q}}_I \\ \dot{\mathbf{b}}_g \\ \dot{\mathbf{v}} \\ \dot{\mathbf{b}}_a \\ \dot{\mathbf{p}}_I \\ \dot{\bar{q}}_C \\ \dot{\mathbf{p}}_C \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{1}{2}\boldsymbol{\Xi}(\bar{q}_I)\mathbf{b}_g \\ \mathbf{0}_{3\times 1} \\ \mathbf{g} - \mathbf{C}^T(\bar{q}_I)\mathbf{b}_a \\ \mathbf{0}_{3\times 1} \\ \mathbf{v} \\ \mathbf{0}_{3\times 1} \\ \mathbf{0}_{3\times 1} \end{bmatrix}}_{\mathbf{f}_0} + \underbrace{\begin{bmatrix} \frac{1}{2}\boldsymbol{\Xi}(\bar{q}_I) \\ \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} \end{bmatrix}}_{\underline{\mathbf{f}}_1} \boldsymbol{\omega}_m + \underbrace{\begin{bmatrix} \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} \\ \mathbf{C}^T(\bar{q}_I) \\ \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} \end{bmatrix}}_{\underline{\mathbf{f}}_2} \mathbf{a}_m \tag{3.28}$$

where $\boldsymbol{\omega}_m$ and $\mathbf{a}_m$ are considered the control inputs, and

$$\boldsymbol{\Xi}(\bar{q}) = \begin{bmatrix} q_4\mathbf{I}_{3\times 3} + \lfloor \mathbf{q}\times \rfloor \\ -\mathbf{q}^T \end{bmatrix} \quad \text{with } \bar{q} = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix} \tag{3.29}$$

Note also that $\mathbf{f}_0$ is a 23×1 vector, while $\underline{\mathbf{f}}_1$ and $\underline{\mathbf{f}}_2$ are both compact representations of 3 vectors of dimension 23×1, i.e.,

$$\underline{\mathbf{f}}_1\boldsymbol{\omega}_m = \mathbf{f}_{11}\omega_{m1} + \mathbf{f}_{12}\omega_{m2} + \mathbf{f}_{13}\omega_{m3}$$

where, for $i = 1 \ldots 3$, $\mathbf{f}_{1i}$ denotes the i-th column vector comprising $\underline{\mathbf{f}}_1$, and ω_{mi} is the i-th scalar component of the rotational velocity vector.

A well-known result that we will use in the observability analysis of (3.28) is the following: When 4 or more[1] known features are detected in each calibration image processed by the filter, the *camera pose* is observable [120] and can be computed in closed-form [3]. Based on this fact, we replace the measurement equation [see (3.19)] with the following pair of inferred measurements of the camera pose expressed with respect to the global frame of reference:

$$ {}^{G}\bar{q}_{C} = \boldsymbol{\xi}_1(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4) = \mathbf{h}_1^*(\mathbf{x}) = \mathbf{J}\bar{q}_{I} \otimes \bar{q}_{C} \tag{3.30} $$

$$ {}^{G}\mathbf{p}_{C} = \boldsymbol{\xi}_2(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4) = \mathbf{h}_2^*(\mathbf{x}) = \mathbf{p}_{I} + \mathbf{C}^T(\bar{q}_{I})\mathbf{p}_{C} \tag{3.31} $$

where $\mathbf{C}(\bar{q}_I)$ is the rotational matrix corresponding to the quaternion $\bar{q}_I$, $\otimes$ denotes quaternion multiplication, and,

$$ \mathbf{J}\bar{q}_I = \bar{q}_I^{-1}, \quad \mathbf{J} \triangleq \begin{bmatrix} -\mathbf{I}_{3\times 3} & 0 \\ 0 & 1 \end{bmatrix} \tag{3.32} $$

At this point, we should note that the functions $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ in (3.30) and (3.31) need not to be known explicitly. Instead what is required for the observability analysis is their functional relation with the random variables, $\bar{q}_I$ and $\mathbf{p}_I$, and the unknown parameters, $\bar{q}_C$ and $\mathbf{p}_C$, appearing in the system's state vector.

Furthermore, we enforce the unit-quaternion constraints by employing the following additional measurement equations:

$$ h_3^*(\mathbf{x}) = \bar{q}_I^T \bar{q}_I - 1 = 0 \tag{3.33} $$

$$ h_4^*(\mathbf{x}) = \bar{q}_C^T \bar{q}_C - 1 = 0 \tag{3.34} $$

According to Remark 2.2, it suffices to show that a subset of the rows of the observability matrix $\nabla\mathcal{O}$ [see Theorem 2.1] are linearly independent. In the remaining of this section, we prove that the system described by (3.28) and (3.30)-(3.34) is observable by computing among the candidate zeroth, first, and second-order Lie derivatives of $\mathbf{h}_1^*$, $\mathbf{h}_2^*$, and h_3^*, the ones whose gradients ensure that $\nabla\mathcal{O}$ is full rank.

- *Zeroth-order Lie derivatives* ($\mathfrak{L}^0\mathbf{h}_1^*, \mathfrak{L}^0\mathbf{h}_2^*, \mathfrak{L}^0 h_3^*$)

[1] If an initial estimate of the pose is available, then observation of only 3 known features is sufficient for uniquely determining the camera pose [120].

By definition, the zeroth-order Lie derivative of a function is the function itself, i.e.,

$$\mathfrak{L}^0\mathbf{h}_1^* = \mathbf{h}_1^* = \bar{q}_I^{-1} \otimes \bar{q}_C \tag{3.35}$$

$$\mathfrak{L}^0\mathbf{h}_2^* = \mathbf{h}_2^* = \mathbf{p}_I + \mathbf{C}^T(\bar{q}_I)\mathbf{p}_C \tag{3.36}$$

$$\mathfrak{L}^0 h_3^* = h_3^* = \bar{q}_I^T\bar{q}_I - 1 \tag{3.37}$$

Therefore the gradients of the zeroth-order Lie derivatives are exactly the same as the Jacobians of the corresponding measurement functions:

$$\nabla\mathfrak{L}^0\mathbf{h}_1^* = \begin{bmatrix} \mathcal{R}(\bar{q}_C)\mathbf{J} & \mathbf{0}_{4\times 12} & \mathcal{L}(\mathbf{J}\bar{q}_I) & \mathbf{0}_{4\times 3} \end{bmatrix} \tag{3.38}$$

$$\nabla\mathfrak{L}^0\mathbf{h}_2^* = \begin{bmatrix} \boldsymbol{\Psi}(\bar{q}_I, \mathbf{p}_C) & \mathbf{0}_{3\times 9} & \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 4} & \mathbf{C}^T(\bar{q}_I) \end{bmatrix} \tag{3.39}$$

$$\nabla\mathfrak{L}^0 h_3^* = \begin{bmatrix} 2\bar{q}_I^T & \mathbf{0}_{1\times 19} \end{bmatrix} \tag{3.40}$$

where, for a quaternion $\bar{q}$ and a vector $\mathbf{p}$, we define

$$\mathcal{L}(\bar{q}) \triangleq \begin{bmatrix} q_4\mathbf{I}_{3\times 3} - \lfloor \mathbf{q}\times\rfloor & \mathbf{q} \\ -\mathbf{q}^T & q_4 \end{bmatrix} \tag{3.41}$$

$$\mathcal{R}(\bar{q}) \triangleq \begin{bmatrix} q_4\mathbf{I}_{3\times 3} + \lfloor \mathbf{q}\times\rfloor & \mathbf{q} \\ -\mathbf{q}^T & q_4 \end{bmatrix} \tag{3.42}$$

and

$$\boldsymbol{\Psi}(\bar{q}, \mathbf{p}) \triangleq \frac{\partial}{\partial \bar{q}}\mathbf{C}^T(\bar{q})\mathbf{p} \tag{3.43}$$

Note also that for deriving (3.38), we have used the following identities [124]:

$$\begin{aligned} \bar{q}_I^{-1} \otimes \bar{q}_C &= \mathcal{R}(\bar{q}_C)\bar{q}_I^{-1} = \mathcal{R}(\bar{q}_C)\mathbf{J}\bar{q}_I \\ &= \mathcal{L}(\bar{q}_I^{-1})\bar{q}_C = \mathcal{L}(\mathbf{J}\bar{q}_I)\bar{q}_C \end{aligned}$$

- *First-order Lie derivatives* ($\mathfrak{L}^1_{\mathbf{f}_0}\mathbf{h}_1^*, \mathfrak{L}^1_{\mathbf{f}_0}\mathbf{h}_2^*, \mathfrak{L}^1_{\underline{\mathbf{f}}_1}\mathbf{h}_2^*$)

The first-order Lie derivatives of $\mathbf{h}_1^*$ and $\mathbf{h}_2^*$ with respect to $\mathbf{f}_0$ are computed as:

$$\mathfrak{L}^1_{\mathbf{f}_0}\mathbf{h}_1^* = \nabla\mathfrak{L}^0\mathbf{h}_1^* \cdot \mathbf{f}_0 = -\tfrac{1}{2}\mathcal{R}(\bar{q}_C)\mathbf{J}\boldsymbol{\Xi}(\bar{q}_I)\mathbf{b}_g \tag{3.44}$$

$$\mathfrak{L}^1_{\mathbf{f}_0}\mathbf{h}_2^* = \nabla\mathfrak{L}^0\mathbf{h}_2^* \cdot \mathbf{f}_0 = -\tfrac{1}{2}\boldsymbol{\Psi}(\bar{q}_I, \mathbf{p}_C)\boldsymbol{\Xi}(\bar{q}_I)\mathbf{b}_g + \mathbf{v} \tag{3.45}$$

while their gradients are given by:

$$
\begin{aligned}
\nabla \mathfrak{L}^1_{\mathbf{f}_0} \mathbf{h}^*_1 &= \begin{bmatrix} \mathbf{X}_1 & -\frac{1}{2}\mathcal{R}(\bar{q}_C)\mathbf{J}\boldsymbol{\Xi}(\bar{q}_I) & \mathbf{0}_{4\times 9} & \mathbf{X}_2 & \mathbf{0}_{4\times 3} \end{bmatrix} \\
\nabla \mathfrak{L}^1_{\mathbf{f}_0} \mathbf{h}^*_2 &= \begin{bmatrix} \mathbf{X}_3 & \mathbf{X}_4 & \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 10} & \mathbf{X}_5 \end{bmatrix}
\end{aligned}
\tag{3.46}
$$

In these last expressions, $\mathbf{X}_i,\ i = 1 \ldots 5$, are matrices of appropriate dimensions (4×4 the first two, 3×4 the third one, and 3×3 the last two) which, regardless of their values, will be eliminated in the following derivations; hence, they need not be computed explicitly.

The next first-order Lie derivative of interest is that of $\mathbf{h}^*_2$ with respect to $\underline{\mathbf{f}}_1$, i.e., $\mathfrak{L}^1_{\underline{\mathbf{f}}_1} \mathbf{h}^*_2$. At this point, we remind the reader that $\underline{\mathbf{f}}_1$ as defined in (3.28) is a compact representation of 3 column vectors. Similarly, we can also write the resulting Lie derivative in a compact form (i.e., a 3×3 matrix):

$$
\mathfrak{L}^1_{\underline{\mathbf{f}}_1} \mathbf{h}^*_2 = \nabla \mathfrak{L}^0 \mathbf{h}^*_2 \cdot \underline{\mathbf{f}}_1 = \frac{1}{2}\boldsymbol{\Psi}(\bar{q}_I, \mathbf{p}_C)\boldsymbol{\Xi}(\bar{q}_I) \tag{3.47}
$$

The gradients of the 3 columns of $\mathfrak{L}^1_{\underline{\mathbf{f}}_1} \mathbf{h}^*_2$ stacked together give:

$$
\nabla \mathfrak{L}^1_{\underline{\mathbf{f}}_1} \mathbf{h}^*_2 = \begin{bmatrix} \boldsymbol{\Gamma}(\bar{q}_I, \mathbf{p}_C) & \mathbf{0}_{9\times 16} & \boldsymbol{\Upsilon}(\bar{q}_I) \end{bmatrix} \tag{3.48}
$$

where the matrices

$$
\boldsymbol{\Gamma}(\bar{q}_I, \mathbf{p}_C) = \begin{bmatrix} \boldsymbol{\Gamma}_1(\bar{q}_I, \mathbf{p}_C) \\ \boldsymbol{\Gamma}_2(\bar{q}_I, \mathbf{p}_C) \\ \boldsymbol{\Gamma}_3(\bar{q}_I, \mathbf{p}_C) \end{bmatrix}, \ \boldsymbol{\Upsilon}(\bar{q}_I) = \begin{bmatrix} \boldsymbol{\Upsilon}_1(\bar{q}_I) \\ \boldsymbol{\Upsilon}_2(\bar{q}_I) \\ \boldsymbol{\Upsilon}_3(\bar{q}_I) \end{bmatrix} \tag{3.49}
$$

of dimensions 9×4 and 9×3, respectively, have block-row elements (for $i = 1 \ldots 3$)

$$
\boldsymbol{\Gamma}_i(\bar{q}_I, \mathbf{p}_C) = \frac{\partial}{\partial \bar{q}_I}\left[\left(\mathfrak{L}^1_{\underline{\mathbf{f}}_1} \mathbf{h}^*_2\right)\mathbf{e}_i\right], \ \boldsymbol{\Upsilon}_i(\bar{q}_I) = \frac{\partial}{\partial \mathbf{p}_C}\left[\left(\mathfrak{L}^1_{\underline{\mathbf{f}}_1} \mathbf{h}^*_2\right)\mathbf{e}_i\right]
$$

with $\mathbf{e}_1 = [1\ \ 0\ \ 0]^T$, $\mathbf{e}_2 = [0\ \ 1\ \ 0]^T$, and $\mathbf{e}_3 = [0\ \ 0\ \ 1]^T$.

Note that inclusion of all the block-row elements of the gradient (3.48) in the observability matrix, $\mathcal{O}$ [see (3.53)], implies that all components of $\boldsymbol{\omega}_m$ are nonzero. However, as it will become evident later on, in order to prove observability *only two* of the elements of $\boldsymbol{\omega}_m$ need to be nonzero. In such case, matrix $\nabla\mathcal{O}$ will contain the block matrices:

$$\boldsymbol{\Gamma}_{ij}(\bar{q}_I, \mathbf{p}_C) = \begin{bmatrix} \boldsymbol{\Gamma}_i(\bar{q}_I, \mathbf{p}_C) \\ \boldsymbol{\Gamma}_j(\bar{q}_I, \mathbf{p}_C) \end{bmatrix}, \, \boldsymbol{\Upsilon}_{ij}(\bar{q}_I) = \begin{bmatrix} \boldsymbol{\Upsilon}_i(\bar{q}_I) \\ \boldsymbol{\Upsilon}_j(\bar{q}_I) \end{bmatrix} \tag{3.50}$$

with $i, j = 1 \dots 3$, $i \neq j$, instead of $\boldsymbol{\Gamma}(\bar{q}_I, \mathbf{p}_C)$ and $\boldsymbol{\Upsilon}(\bar{q}_I)$.

- *Second-order Lie derivative* ($\mathfrak{L}^2_{\mathbf{f}_0}\mathbf{h}^*_2$)

Finally, we compute the second-order Lie derivative of $\mathbf{h}^*_2$ with respect to $\mathbf{f}_0$:

$$\begin{aligned} \mathfrak{L}^2_{\mathbf{f}_0}\mathbf{h}^*_2 &= \mathfrak{L}^1_{\mathbf{f}_0}\mathfrak{L}^1_{\mathbf{f}_0}\mathbf{h}^*_2 = \nabla\mathfrak{L}^1_{\mathbf{f}_0}\mathbf{h}^*_2 \cdot \mathbf{f}_0 \\ &= -\frac{1}{2}\mathbf{X}_3\,\boldsymbol{\Xi}(\bar{q}_I)\mathbf{b}_g + \mathbf{g} - \mathbf{C}^T(\bar{q}_I)\mathbf{b}_a \end{aligned} \tag{3.51}$$

and its gradient:

$$\nabla\mathfrak{L}^2_{\mathbf{f}_0}\mathbf{h}^*_2 = \begin{bmatrix} \mathbf{X}_6 & \mathbf{X}_7 & \mathbf{0}_{3\times3} & -\mathbf{C}^T(\bar{q}_I) & \mathbf{0}_{3\times7} & \mathbf{X}_8 \end{bmatrix} \tag{3.52}$$

where the matrices $\mathbf{X}_5$, $\mathbf{X}_6$, and $\mathbf{X}_7$ (of dimensions 3×4 the first one and 3×3 the last two) will be eliminated in the ensuing derivations and therefore, we do not need to compute them explicitly.

Stacking together all the previously computed gradients of the Lie derivatives, we form the observability matrix, $\nabla\mathcal{O}$ (see Theorem 2.1):

$$\nabla\mathcal{O} = \begin{bmatrix} \nabla\mathfrak{L}^0\mathbf{h}^*_1 \\ \nabla\mathfrak{L}^0\mathbf{h}^*_2 \\ \nabla\mathfrak{L}^1_{\mathbf{f}_0}\mathbf{h}^*_1 \\ \nabla\mathfrak{L}^1_{\mathbf{f}_0}\mathbf{h}^*_2 \\ \nabla\mathfrak{L}^1_{\underline{\mathbf{f}}_{1ij}}\mathbf{h}^*_2 \\ \nabla\mathfrak{L}^0 h^*_3 \\ \nabla\mathfrak{L}^2_{\mathbf{f}_0}\mathbf{h}^*_2 \end{bmatrix} = \begin{bmatrix} \mathcal{R}(\bar{q}_C)\mathbf{J} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathcal{L}(\mathbf{J}\bar{q}_I) & \mathbf{0}_{4\times3} \\ \boldsymbol{\Psi}(\bar{q}_I, \mathbf{p}_C) & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{C}^T(\bar{q}_I) \\ \mathbf{X}_1 & -\frac{1}{2}\mathcal{R}(\bar{q}_C)\mathbf{J}\boldsymbol{\Xi}(\bar{q}_I) & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{X}_2 & \mathbf{0}_{4\times3} \\ \mathbf{X}_3 & \mathbf{X}_4 & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{X}_5 \\ \boldsymbol{\Gamma}_{ij}(\bar{q}_I, \mathbf{p}_C) & \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} & \mathbf{0}_{6\times4} & \boldsymbol{\Upsilon}_{ij}(\bar{q}_I) \\ 2\bar{q}_I^T & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times4} & \mathbf{0}_{1\times3} \\ \mathbf{X}_6 & \mathbf{X}_7 & \mathbf{0}_{3\times3} & -\mathbf{C}^T(\bar{q}_I) & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{X}_8 \end{bmatrix} \tag{3.53}$$

In order to prove that the system described by (3.28) and (3.30)-(3.34) is observable, we employ the result of Theorem 2.1 and show that matrix $\nabla\mathcal{O}$ is full rank (i.e., the state space of the system is spanned by the gradients of the Lie derivatives of the measurement functions [51, 111]). Before presenting the main result of this section (see Lemma 3.3), we first state the following two lemmas:

Lemma 3.1. *The matrix*

$$\mathbf{A}_{ij} = \begin{bmatrix} \boldsymbol{\Gamma}_{ij} & \boldsymbol{\Upsilon}_{ij} \\ 2\bar{q}_I^T & \mathbf{0}_{1\times 3} \end{bmatrix} \tag{3.54}$$

formed by the 5-th and 6-th block-row elements of the first and last block-columns of the observability matrix $\nabla\mathcal{O}$ [see (3.53)], with $\boldsymbol{\Gamma}_{ij}$ and $\boldsymbol{\Upsilon}_{ij}$ defined in (3.50), is full rank.

Proof. We prove this lemma for the case of practical interest when the elements of $\mathbf{p}_C = [p_1 \;\; p_2 \;\; p_3]^T$ (i.e., the vector denoting the position of the camera expressed with respect to the IMU frame) are nonzero.[2]

For $i, j = 1 \ldots 3$, $i \neq j$, we expand $\mathbf{A}_{ij}$ as:

$$\mathbf{A}_{ij} = \begin{bmatrix} \boldsymbol{\Gamma}_{i} & \boldsymbol{\Upsilon}_{i} \\ \boldsymbol{\Gamma}_{j} & \boldsymbol{\Upsilon}_{j} \\ 2\bar{q}_I^T & \mathbf{0}_{1\times 3} \end{bmatrix} \begin{array}{l} \}\,(1:3) \leftrightarrow \omega_{mi} \\ \}\,(4:6) \leftrightarrow \omega_{mj} \\ \}\,7 \end{array} \tag{3.55}$$

The variables on the right side of the matrix next to the row numbers, specify the component of $\boldsymbol{\omega}_m = [\omega_{m1} \;\; \omega_{m2} \;\; \omega_{m3}]^T$ that are excited in order for these rows to be included in the observability matrix $\nabla\mathcal{O}$ [see (3.53)]. After considerable algebra, it can be shown that [89]:

$$\det\left(\mathbf{A}_{ij}\right) = 8(-1)^k p_k \left(p_j^2 + p_i^2\right) \tag{3.56}$$

where $i, j, k = 1 \ldots 3$, $k \neq i$, $k \neq j$, and $i \neq j$. We conclude the proof by noting that since all elements of $\mathbf{p}_C$ are nonzero, the determinant of $\mathbf{A}_{ij}$ in (3.56) is nonzero; hence $\mathbf{A}_{ij}$ is full rank. □

Corollary 3.1. *The matrix described by (3.54) is full rank if the IMU-camera rig is rotated about at least two different axes.*

Note that only two block rows of $\boldsymbol{\Gamma}(\bar{q}_I, \mathbf{p}_C)$ and $\boldsymbol{\Upsilon}(\bar{q}_I)$ [see (3.49)] – the ones corresponding to two nonzero components of $\boldsymbol{\omega}_m$ – are included in $\mathbf{A}_{ij}$ [see (3.54)]. Therefore, the third component of $\boldsymbol{\omega}_m$ can be zero (i.e., no rotation around the corresponding axis) without affecting the observability properties of the system.

[2] Note that $\mathbf{p}_C = \mathbf{0}_{3\times 1}$ is not physically realizable since it means that the centers of the IMU and the camera coincide. Also the case when one or more elements of $\mathbf{p}_C$ are zero, is extremely rare in practice since it requires perfect position alignment of the camera and the IMU. However, the latter case is addressed in [89] where it is shown that the system is still observable when all three degrees of rotational freedom are excited.

Lemma 3.2. *For any unit-quaternions $\bar{q}$ and $\bar{s}$, matrix $\mathbf{B} = \mathcal{R}(\bar{q})\mathbf{J}\boldsymbol{\Xi}(\bar{s})$ is full rank.*

Proof. This can be readily proved by computing $\mathbf{B}^T\mathbf{B}$, which is a 3×3 matrix:

$$\mathbf{B}^T\mathbf{B} = \boldsymbol{\Xi}^T(\bar{s})\mathbf{J}^T\mathcal{R}^T(\bar{q})\mathcal{R}(\bar{q})\mathbf{J}\boldsymbol{\Xi}(\bar{s}) = \mathbf{I}_{3\times3} \tag{3.57}$$

Therefore, matrix $\mathbf{B}$ is full rank. For computing (3.57), we used the identities $\mathcal{R}^T(\bar{q})\mathcal{R}(\bar{q}) = \mathbf{I}_{4\times4}$, and $\boldsymbol{\Xi}^T(\bar{s})\boldsymbol{\Xi}(\bar{s}) = \mathbf{I}_{3\times3}$ [124]. □

Lemma 3.3. *The observability matrix, $\nabla\mathcal{O}$ [see (3.53)], is full rank when the IMU-camera rig is rotated about at least two different axes.*

Proof. Here we provide a sketch of the proof based on block Gaussian elimination (for details please see [89]). We start by employing Lemma 3.1 and Corollary 3.1 to eliminate all the matrices in the first and last columns of $\nabla\mathcal{O}$. The next step is to eliminate $\mathbf{X}_2$ using $\mathcal{L}(\mathbf{J}\bar{q}_I)$, i.e., the (1,6) block element of $\nabla\mathcal{O}$ in (3.53). Note that $\bar{q}$ is unit quaternion and $\det(\mathcal{L}(\mathbf{J}\bar{q})) = ||\mathbf{J}\bar{q}|| = ||\bar{q}|| = 1$ [see (3.32), (3.34), and (3.41)]. Finally, since $-\frac{1}{2}\mathcal{R}(\bar{q}_C)\mathbf{J}\boldsymbol{\Xi}(\bar{q}_I)$ is full rank [see Lemma 3.2], it can be used to eliminate $\mathbf{X}_4$ and $\mathbf{X}_7$. Following these steps, $\nabla\mathcal{O}$ reduces to:

$$\begin{bmatrix}
\mathbf{0}_{4\times4} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{I}_{4\times4} & \mathbf{0}_{4\times3} \\
\mathbf{0}_{3\times4} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times3} \\
\mathbf{0}_{3\times4} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times3} \\
\mathbf{0}_{1\times4} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times4} & \mathbf{0}_{1\times3} \\
\mathbf{0}_{3\times4} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times3} \\
\mathbf{I}_{4\times4} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times3} & \mathbf{0}_{4\times4} & \mathbf{0}_{4\times3} \\
\mathbf{0}_{3\times4} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{I}_{3\times3} \\
\mathbf{0}_{3\times4} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{C}^T(\bar{q}_I) & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times3}
\end{bmatrix} \tag{3.58}$$

Considering that a property of a rotation matrix is that it is full rank ($\forall\bar{q},\ \det(\mathbf{C}(\bar{q})) = 1$), it is easy to see that (3.58) is full rank, indicating that $\nabla\mathcal{O}$ is also full rank. □

Corollary 3.2. *The system described by (3.28) and (3.30)-(3.34) is observable regardless of the linear motion of the IMU-camera rig.*

This is evident from the fact that for proving Lemma 3.3, we did *not* use any Lie derivatives with respect to $\underline{\mathbf{f}}_2$ [see (3.28)]. Therefore, $\mathbf{a}_m$, the measured linear acceleration can take arbitrary values without compromising the observability of the

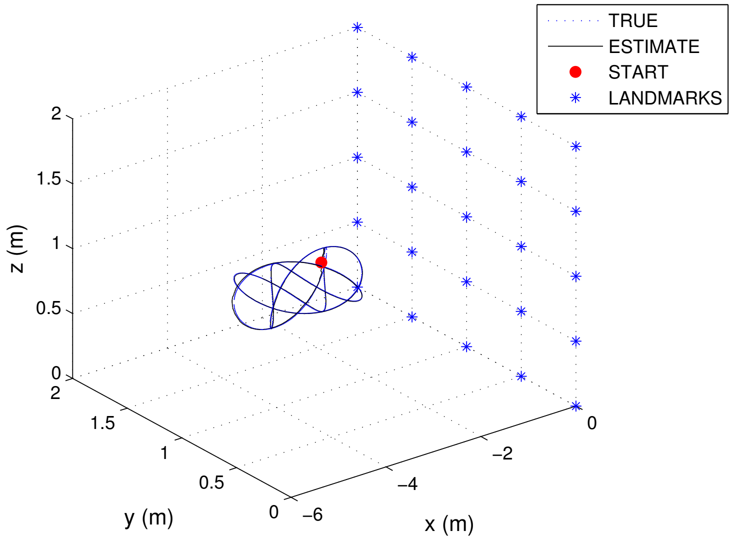


Figure 3.2: Trajectory of the IMU-camera system for 15 sec.

system. This observation has important practical implications when no significant linear motion is possible due to physical constraints (e.g., calibration of an IMU-camera rig in an indoor laboratory): the IMU-camera transformation can be accurately estimated even if no linear acceleration is exerted.

Remark 3.1. Since no noise is injected into the system along the directions of the IMU-camera transformation [see (3.6)], regardless of the observability of the system, the uncertainty of the IMU-camera transformation will never increase.

When the linearization of the IMU-camera calibration system is sufficiently accurate, this remark has the following important implication: running the estimation algorithm during periods when the observability conditions are not met (e.g., as a result of stopping the IMU-camera rig), will not decrease the accuracy of the IMU-camera estimates; although it might not improve their quality either. However, it is advisable to excite at least two degrees of rotational freedom for sufficiently long time at the beginning of the calibration process, so as to significantly reduce the error in the IMU-camera transformation and ensure the validity of the linear approximation.

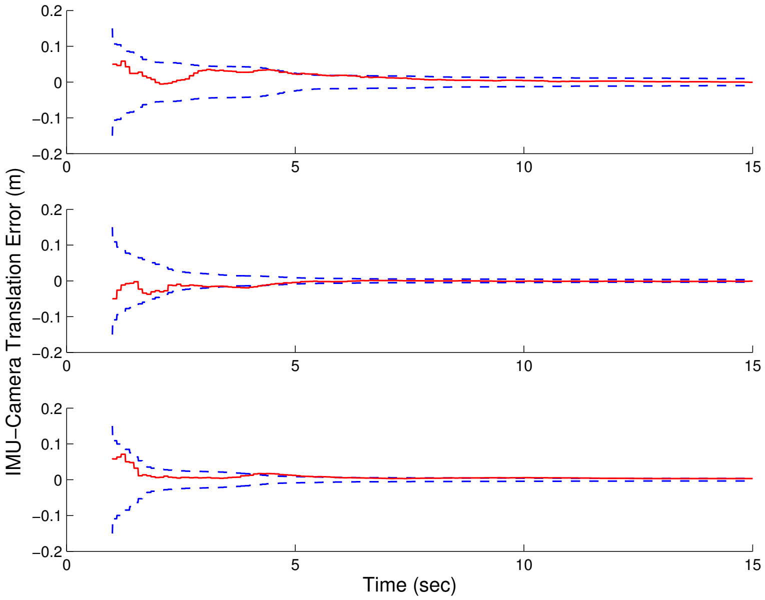


Figure 3.3: State-estimate error and 3σ bounds for the IMU-camera transformation: Translation along axes x, y and z. The initial error is ${}^I\widetilde{\mathbf{p}}_C = [5 \ \ -5 \ \ 6]^T$cm.

3.5 Simulation and Experimental Results

3.5.1 Simulation Results

In order to validate the proposed EKF algorithm for estimating the IMU-camera transformation when ground truth is available, we have performed a number of simulation experiments. In our simulation setup, an IMU-camera rig moves in front of a calibration target containing 25 known features. These correspond to the vertices of a rectangular grid with 50 × 50 cm cell size, which is aligned with the yz plane (see Fig. 3.2). The camera is assumed to have 50° field of view. Additionally, the image measurements received at a rate of 10 Hz, are distorted with noise of σ=1 pixel. The IMU noise characteristics are the same as those of the ISIS IMU used in the real-world experiments (see Section 3.5.2). The IMU measurements are received at 100 Hz.

The initial alignment error for translation is set to ${}^I\widetilde{\mathbf{p}}_C = [5 \ \ -5 \ \ 6]^T$ cm with a standard deviation of 5 cm in each axis. The initial alignment error for rotation is set to $\boldsymbol{\delta\theta} = [4^\circ \ \ -4^\circ \ 3^\circ]^T$ [see (3.15)] with 3° standard deviation of uncertainty in each axis of rotation. Consequently, the filter state vector and error-state covariance matrix are initialized according to the process described in Section 3.3.1.

Following the initialization step, the system performs a spiral motion within 3-5 m off the calibration pattern. The Extended Kalman filter (EKF) processes the IMU and

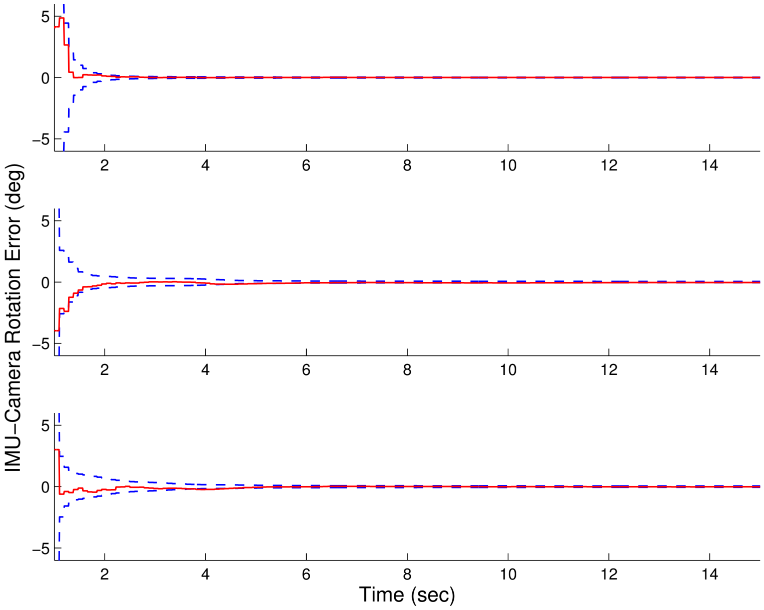


Figure 3.4: State-estimate error and 3σ bounds for the IMU-camera transformation: Rotation about axes x (roll), y (pitch), and z (yaw). The initial alignment errors are $\boldsymbol{\delta\theta} = [4^\circ \ \ -4^\circ \ \ 3^\circ]^T$.

camera measurements and concurrently estimates *all* the components of the state vector [see (3.2)]. The actual and estimated trajectories are shown in Fig. 3.2. For the duration of this simulation (only 15 sec), 150 images were processed and, on the average, 21.7 landmarks were visible in each image. The state-estimate errors and their 3σ bounds for the 6 d.o.f. transformation between the IMU and the camera in a typical simulation are shown in Figs. 3.3 and 3.4. As evident from these plots, even with a relatively large initial error for the IMU-camera transformation, the algorithm is still able to attain very accurate estimates of the calibration parameters. The final uncertainty (3σ) of the estimates is $[0.96 \ \ 0.84 \ \ 0.90]^T$ cm for translation and $[0.072^\circ \ 0.120^\circ \ 0.120^\circ]^T$ for rotation.

General motion vs. rotation only

In Section 3.4, we have shown that the system describing the IMU-camera calibration process is observable when the IMU-camera rig undergoes rotational motion even if no translation occurs. Hereafter, we examine the achievable accuracy for motions with and without translation after 100 sec when the IMU-camera rig undergoes (i) spiral motion (i.e., exciting all 6 d.o.f.) and (ii) pure rotation (i.e., exciting only the d.o.f. corresponding to attitude). In all these simulations, the initial uncertainty of the IMU-camera translation and rotation are set to 15 cm and 9° deg (3σ) in each axis respectively.

Table 3.1: Final uncertainty (3σ) of the IMU-camera parameters after 100 sec for two motion scenarios. xyz represents translation along the x, y, and z axes. rpy indicates rotation about the local x (roll), y (pitch), and z (yaw) axes.

3σ	x (cm)	y (cm)	z (cm)	r (°)	p (°)	y (°)
Initial	15	15	15	9	9	9
xyz-rpy	0.18	0.14	0.13	0.013	0.013	0.013
rpy	0.25	0.22	0.22	0.0082	0.024	0.024

Table 3.2: Monte Carlo Simulations: Comparison of the standard deviations of the final IMU-camera transformation error (σ_{err}), and the average computed uncertainty of the estimates (σ_{est}).

	x (cm)	y (cm)	z (cm)	r (°)	p (°)	y (°)
σ_{err}	0.29	0.23	0.28	0.019	0.036	0.039
σ_{est}	0.31	0.24	0.28	0.019	0.039	0.040

A summary of these results is shown in Table 3.1. The third row of Table 3.1, (xyz-rpy), corresponds to motion with all 6 d.o.f. excited. In this case, after sufficient time, the translation uncertainty is reduced to less than 2 mm (3σ) in each axis.

By comparing the results of Table 3.1 to those corresponding to Figs. 3.3 and 3.4, it is obvious that by allowing the EKF algorithm to run for longer period of time (i.e., 100 sec instead of 15 sec), we can estimate the calibration parameters more accurately. Additionally, as it can be seen in this particular example, the translational uncertainty along the x axis is slightly higher than the uncertainty along the other two axes. This is a typical result observed in all simulations with similar setup. The main reason for this is the limited range of pitch and yaw rotations (i.e., about the y and z axes, respectively) required for keeping the landmarks within the field of view. On the other hand, the roll rotation (about the x axis) is virtually unlimited and it can span a complete circle without losing visual contact with the landmarks (note that the optical axis of the camera is aligned with the local x axis).

The fourth row of Table 3.1 corresponds to a scenario where the motion is constrained to pure rotation. As expected the system is still observable and both the translation and the rotation between the IMU and the camera are accurately estimated [89]. The accuracy of the rotation estimation between the IMU and the camera in both scenarios (i.e., with or without translation) is shown in the last three columns of Table 3.1. As evident, in all cases the rotational parameters can be estimated extremely accurately,

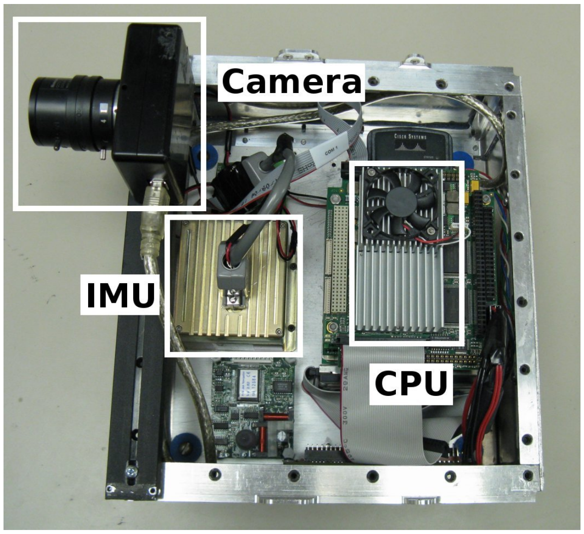


Figure 3.5: Testbed used for the experiments.

even when the system has not undergone any translation.

Monte Carlo Simulations

Finally, we have conducted Monte Carlo simulations to statistically evaluate the accuracy of the filter. We ran 100 simulations with a setup similar to the first simulation described in this section. The initial standard deviation of the IMU-camera transformation is set to 3 cm for translation and 3° for rotation. The initial values in each run are randomly generated according to a Gaussian probability distribution with these standard deviations. Each simulation is run for 15 sec and the final calibration errors along with their estimated uncertainty are recorded. The ensemble mean of the recorded errors is $[0.058\ -0.002\ \ 0.044]^T$ cm for translation and $[-0.0038^\circ\ 0.0013^\circ\ -0.0009^\circ]^T$ for rotation. It can be seen that the mean error is at least one order of magnitude smaller than the typical error, demonstrating that the filter is indeed unbiased.

The standard deviations of the recorded errors are shown in the second row of Table 3.2. The third row of this table shows the average of the standard deviations computed by the EKF at each realization of the experiment. Comparison of these two rows indicates consistency of the filter as the standard deviation of the actual error is smaller than or equal to the standard deviations computed by the EKF (i.e., the filter estimates are not overconfident).

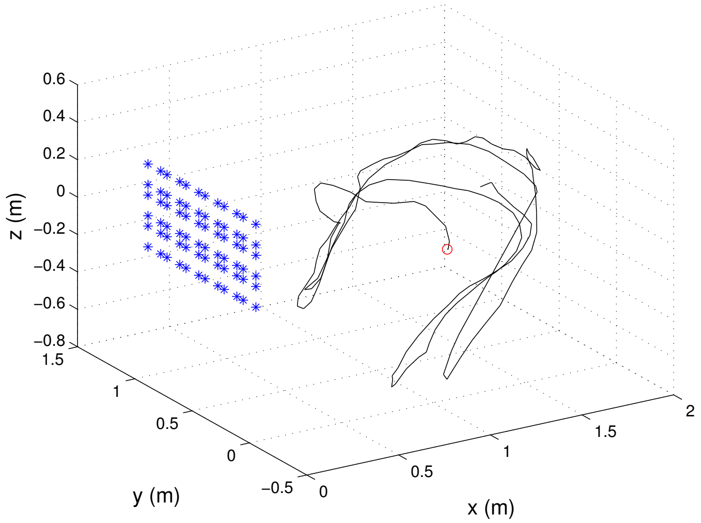


Figure 3.6: Estimated trajectory of the IMU for 50 sec. The starting point is shown by a circle on the trajectory.

3.5.2 Experimental Results

In order to demonstrate the validity of our EKF algorithm in realistic situations, we have conducted experiments using a testbed which consists of an ISIS IMU, a firewire camera, and a PC104 computer for data acquisition (see Fig. 3.5). The IMU and the camera are rigidly mounted on the chassis and their relative pose does not change during the experiment. The intrinsic parameters of the camera were calibrated prior to the experiment [15] and are assumed constant. The camera's field of view is 60° with a focal length of 9 mm. The resolution of the images is 1024 × 768 pixels. Images are recorded at a rate of 3.75 Hz while the IMU provides measurements at 100 Hz. The PC104 stores the images and the IMU measurements for post-processing using our EKF algorithm. Furthermore, considering that the exact values of the IMU-camera transformation (ground truth) were not available in this experiment, a Batch Least Squares (BLS) estimator was implemented to provide the *best possible estimates* of the alignment parameters by post-processing all the collected IMU and image measurements concurrently (see Appendix A).

A calibration pattern (checker board) was used to provide 72 globally known landmarks which were placed 5.5 cm-11 cm apart from each other. The bottom-left corner of this checker board was selected as the origin of the global reference frame, and the calibration pattern was aligned with the direction of the gravitational acceleration. The

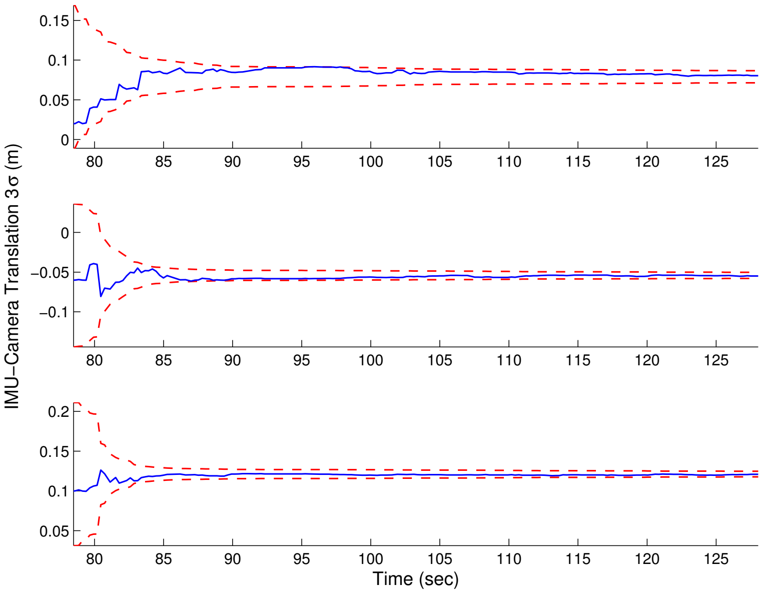


Figure 3.7: Time-evolution of the estimated IMU-camera translation along the x, y, and z axes (solid blue lines) and the corresponding 3σ bounds centered around the BLS estimates (dashed red lines).

landmarks (i.e., the corners of the squares) were extracted using a least-squares corner detector. We have assumed that the camera measurements are corrupted by additive white Gaussian noise with standard deviation equal to 2 pixels.[3]

The hand-measured translation and rotation between the IMU and the camera was used as an initial guess for the unknown transformation. Additionally, the pose of the IMU was initialized as described in Section 3.3.1. Finally, initialization of the gyro and the accelerometer biases was performed by placing the testbed in a static position for approximately 80 sec. During this time the EKF processed IMU and camera measurements while enforcing the static constraint (zero position and attitude displacement). The resulting state vector along with the error-state covariance matrix were then directly used to run the experiment.

Once the initialization process was complete, we started moving the testbed while the camera was facing the calibration pattern. For the duration of the experiment, the distance between the camera and the calibration pattern varied between 0.5 m-2.5 m in order to keep the corners of the checker board visible. Additionally, the testbed was moved in such a way so as to excite all degrees of freedom while at the same time keeping

[3] The actual pixel noise is less than 2 pixels. However, in order to compensate for the existence of unmodeled nonlinearities and imperfect camera calibration, we have inflated the noise standard deviation to 2 pixels.

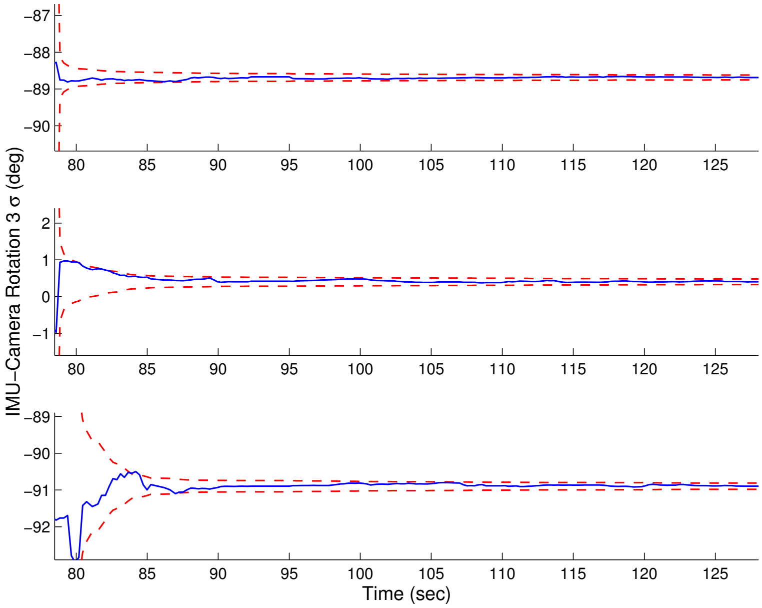


Figure 3.8: Time-evolution of the estimated IMU-camera rotation about the axes x, y, and z (solid blue lines), and the corresponding 3σ bounds centered around the BLS estimates (dashed red lines).

the landmarks within the camera's field of view.

During the motion of the testbed ($\sim$ 50 sec), 180 images were recorded, of which 24 were not processed due to motion-blur. The EKF algorithm was able to estimate the IMU-camera transformation while keeping track of the IMU pose, velocity, and IMU biases. The estimated trajectory of the IMU is shown in Fig. 3.6.

The time-evolution of the estimated calibration parameters along with their estimated 3σ bounds centered around the BLS estimates, are depicted in Figs. 3.7 and 3.8. As evident from these plots, the calibration parameters converge to steady-state values after approximately 130 sec (including the 80 sec of the duration of the initialization process). The small inconsistencies observed during the initial transient period are due to the nonlinearities of the system and measurement models, and the imprecise initialization of the filter state vector. In particular, evaluating the Jacobians using the inaccurate state estimates available at the beginning of the calibration process, causes the estimates to fluctuate significantly around their true values. As more feature observations become available, the accuracy of the state estimates improves, which subsequently increases the accuracy of the system's Jacobians and eventually leads to convergence to the estimates' true values.

A summary of the results from this experiment is provided in Table 3.3. It is

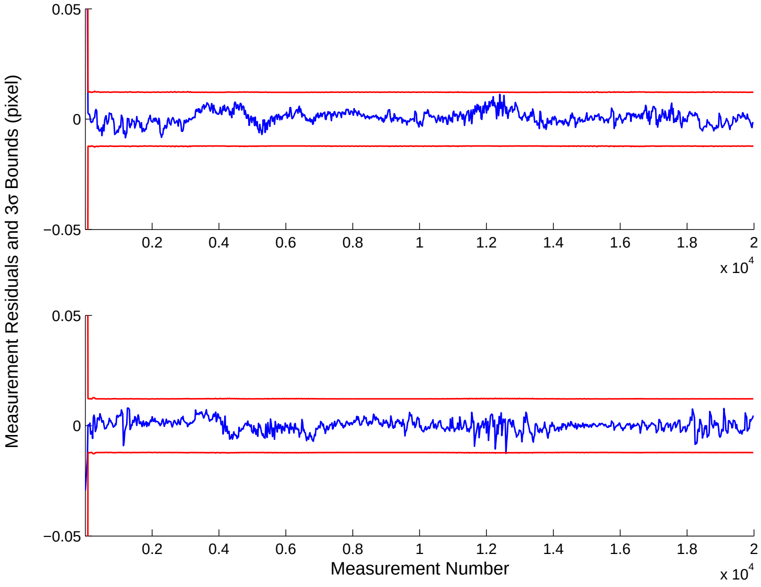


Figure 3.9: [Calibrated IMU-Camera] Measurement residuals along with their 3σ bounds for the horizontal u (top plot) and vertical v (bottom plot) axes of the images.

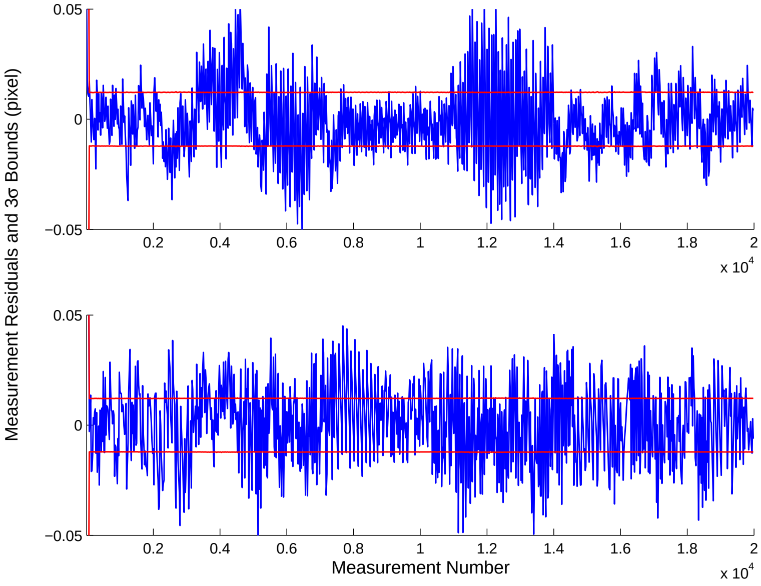


Figure 3.10: [Uncalibrated IMU-Camera] Measurement residuals along with their 3σ bounds for the horizontal u (top plot) and vertical v (bottom plot) axes of the images.

Table 3.3: Initial, EKF, and BLS estimates of the IMU-camera parameters and their uncertainty for the described experiment.

	x $\pm 3\sigma$ (cm)	y$\pm 3\sigma$ (cm)	z$\pm 3\sigma$ (cm)
Initial	2 ± 9	-6 ± 9	10 ± 9
EKF	7.93 ± 0.76	-5.49 ± 0.38	12.09 ± 0.35
BLS	7.43 ± 0.15	-5.19 ± 0.13	12.12 ± 0.10

	roll$\pm 3\sigma$ (°)	pitch$\pm 3\sigma$ (°)	yaw$\pm 3\sigma$ (°)
Initial	-90 ± 6	0 ± 6	-90 ± 6
EKF	-88.69 ± 0.07	0.40 ± 0.08	-90.89 ± 0.08
BLS	-88.71 ± 0.02	0.38 ± 0.04	-90.93 ± 0.05

worth mentioning that the initial uncertainty of 9 cm (3σ) in the translation parameters improves to less than 0.8 cm (3σ) for all axes. Additionally the initial uncertainty of 6° (3σ) decreases to less than 0.1° (3σ) for each axis of rotation. Moreover, this table shows that the EKF estimator, *which can run in real-time*, attains a level of accuracy close to that of the BLS. Note also that the final accuracy of the EKF is consistent with that of the BLS, demonstrating that the EKF is not overconfident. A further indicator of the consistency of the EKF is provided in Fig. 3.9. As shown in these plots, the measurement residuals of the filter along the image axes (i.e., re-projection errors) lie within their estimated 3σ bounds.

In order to stress the importance of acquiring precise IMU-camera calibration estimates, we have also tested with the same experimental setup, an EKF-based estimator that does *not* estimate the calibration parameters online. Instead this filter uses the initial guess for the unknown IMU-camera transformation to estimate the IMU pose, velocity, and biases. In this case, and as evident from the camera measurement residuals shown in Fig. 3.10, the approximate values for the calibration parameters lead to large inconsistencies of the estimator.

3.6 Summary

In this chapter, we have presented an EKF-based algorithm for estimating the transformation between an IMU and a camera rigidly attached on a mobile platform. To the best of our knowledge this is the first approach to the IMU-camera calibration problem that appropriately accounts for the *time correlation* between the IMU measurement. Additionally, and contrary to previous work on this subject, we do *not* separate the task of translation estimation from rotation estimation, and hence prevent error propagation. Moreover, by treating the problem within the Kalman filtering framework, we are also able to compute the *covariance* of the estimated quantities as an indicator of the achieved level of accuracy. Therefore by accounting for this uncertainty in the consequent estimation algorithm, we are able to explicitly model their impact. Last but not the least, an important feature of this algorithm is the ability to perform the calibration process *without* requiring any specific testbed (such as rotating table [75] or high precision 3D laser scanner [62]) except the calibration pattern which is also needed when calibrating the intrinsic parameters of the camera. The derived estimator was tested both in simulation and experimentally and it was shown to achieve accuracy in the order of millimeters and sub-degrees, respectively, for the translational and rotational components of the IMU-camera transformation. Additionally and for the first time, the observability of the nonlinear system describing the IMU-camera calibration was investigated by employing the observability rank condition based on Lie derivatives. As presented, estimating the IMU-camera transformation requires exciting only 2 of the rotational d.o.f., while no translational motion is necessary.

Chapter 4

3D Lidar-Camera Calibration[†]

4.1 Introduction and Related Work

As demonstrated in the Defense Advanced Research Projects Agency (DARPA) Urban Challenge, commercially available high-speed 3D LIDARs, such as the Velodyne, have made autonomous navigation and mapping within dynamic environments possible. In most applications, however, another sensor is employed in conjunction with the 3D LIDAR to assist in localization and place recognition. In particular, spherical cameras are often used to provide visual cues and to construct photorealistic maps of the environment. In these scenarios, accurate extrinsic calibration of the six degrees of freedom (d.o.f.) transformation between the two sensors is a prerequisite for optimally combining their measurements.

Several methods exist for calibrating a 2D laser scanner with respect to a camera. The work of Zhang and Pless relies on the observation of a planar checkerboard by both sensors. In particular, corners are detected in the images and planar surfaces are extracted from the laser measurements [139]. The detected corners are used to determine the normal vector and distance of the planes where the laser-scan endpoints lie. Using this geometric constraint, the estimation of the transformation between the two sensors is formulated as a non-linear least-squares problem and solved iteratively. A simplified linear least-squares solution is also provided to initialize the iterative nonlinear algorithm. More recently, a minimal approach for calibrating a 2D laser scanner with

[†] This is a joint work with Dimitrios G. Kottas, partially appeared at the 15th International Symposium on Robotics Research (ISRR), Flagstaff, AZ, 2011 [86]. A complete version of this work is in press in the International Journal of Robotics Research, 2012.

respect to a camera using only six measurements of a planar calibration board, is presented in [97] . The computed transformation is then used in conjunction with RANdom SAmple Consensus (RANSAC) [41] to initialize an iterative least-squares refinement.

The existing 2D laser scanner-camera calibration methods are extended to 3D LIDARs in [132] and [103]. In both works, a geometric constraint similar to the one presented in [139] is employed to form a nonlinear least-squares cost function which is iteratively minimized to estimate the LIDAR-camera transformation. In addition, an initialization method for the iterative minimization based on a simplified linear least-squares formulation is presented in [132]. Specifically, the estimation of relative rotation and translation are decoupled, and then each of them is computed from a geometric constraint between the planar segments detected in the measurements of both the 3D LIDAR and the camera. An alternative 3D LIDAR-camera calibration approach is described in [113], where several point correspondences are manually selected in images and their associated LIDAR scans. Then, the Perspective n-point Pose (PnP) estimation algorithm of Quan and Lan [106] is employed to find the transformation between the camera and the 3D LIDAR based on these point correspondences. In a different approach, presented in [118], the structural edges extracted from 3D LIDAR scans are matched with the vanishing points of the corresponding 2D images to compute a coarse 3D LIDAR-camera transformation, followed by an iterative least-squares refinement.

The main limitation of the above methods is that they assume the 3D LIDAR to be intrinsically calibrated. If the LIDAR's intrinsic calibration is not available or sufficiently accurate, then the calibration accuracy as well as the performance of subsequent LIDAR-camera data fusion significantly degrades. Pandey *et al.* have partially addressed this issue for the Velodyne 3D LIDAR by first calibrating only some of its intrinsic parameters [103]. However, the suggested intrinsic calibration procedure is also iterative, and no method is provided for initializing it. While several of the intrinsic parameter1s of a LIDAR may be initialized using the technical drawings of the device (if available), other parameters, such as the offset in the range measurements induced by the delay in the electronic circuits, cannot be determined in this way.

To address these limitations, in this chapter we propose a novel algorithm for jointly estimating the intrinsic parameters of a revolving-head 3D LIDAR as well as the LIDAR-camera transformation. Specifically, we use measurements of a calibration plane at various configurations to establish geometric constraints between the LIDAR's intrinsic parameters and the LIDAR-camera 6 d.o.f. relative transformation. We process these measurement constraints to estimate the calibration parameters as follows: First, we analytically compute an initial estimate for the intrinsic and extrinsic calibration parameters in two steps. Subsequently, we employ a batch iterative (nonlinear) least-squares method to refine the accuracy of the estimated parameters.

In particular, to analytically compute an initial estimate, we relax the estimation problem by seeking to determine the transformation between the camera and each one of the conic laser scanners within the LIDAR, along with its intrinsic parameters. As a first step, we formulate a nonlinear least-squares problem to estimate the 3 d.o.f. rotation between each conic laser scanner and the camera, as well as a subset of the laser scanner's intrinsic parameters. The optimality conditions of this nonlinear least-squares form a system of polynomial equations, which we solve analytically using an algebraic-geometry approach to find all its critical points. Amongst these, the one that minimizes the least-squares cost function corresponds to the global minimum and provides us with the initial estimates for the relative rotation and the first set of intrinsic LIDAR parameters. In the next step, we use a linear least-squares algorithm to compute the initial estimate for the relative translation between the camera and the conic laser scanners, and the remaining intrinsic parameters.

Once all initial estimates are available, we finally perform a batch iterative joint-optimization of the LIDAR-camera transformation and the LIDAR's intrinsic parameters. As part of our contributions, we also study the observability properties of the problem and present the minimal necessary conditions for concurrently estimating the LIDAR's intrinsic parameters and the LIDAR-camera transformation. Our experimental results demonstrate that our proposed method significantly improves the accuracy of the intrinsic calibration parameters of the LIDAR, as well as, the LIDAR-camera transformation.

The remainder of this chapter is structured as follows: The calibration problem is formulated in Section 4.2, and the proposed solution is presented in Sections 4.3 and 4.4. In Section 4.5 the observability of the problem is investigated and in Section 4.6, an experimental comparison of our method with the approach of Pandey *et al.* [103] is provided, and photorealistic 3D reconstruction of indoor and outdoor scenes using the estimated calibration parameters are presented. In Section 4.7, a summary of this chapter is provided.

Table 4.1: Notations Pertinent to 3D LIDAR-Camera Calibration.

$\{L\}$	LIDAR's coordinate frame of reference.
$\{L_i\}$	Coordinate frame of reference corresponding to the i-th laser scanner, $i = 1, \ldots, K$.
$\{B_j\}$	Coordinate frame of reference corresponding to the calibration board at the j-th configuration.
$\{C\}$	Camera's coordinate frame of reference.
ϕ_i	Elevation angle of the i-th laser scanner.
θ_{oi}	Azimuth angle between coordinate frames $\{L\}$ and $\{L_i\}$.
θ_{ik}	Azimuth angle of the k-th shot of the i-th laser scanner w.r.t. $\{L_i\}$.
ρ_{ik}	Range measurement of the k-th shot of the i-th laser scanner w.r.t. $\{L_i\}$.
ρ_{oi}	Range offset of the i-th laser scanner.
α_i	Scale factor of the i-th laser scanner.
h_i	Vertical offset of the i-th laser scanner w.r.t. $\{L\}$.
${}^{C}\bar{\mathbf{n}}_j$	Normal vector of the calibration plane, at its j-th configuration, w.r.t. $\{C\}$.
d_j	Distance of the calibration plane, at its j-th configuration from the origin of $\{C\}$.
${}^{L_i}\mathbf{p}_{ijk}$	k-th intrinsically corrected point, belonging to the calibration board at its j-th configuration, measured by the i-th laser scanner, w.r.t. $\{L_i\}$.

4.2 Problem Formulation

A revolving-head 3D LIDAR consists of K conic laser scanners mounted on a rotating head so that they span a 360° panoramic (azimuth) view (see Fig. 4.1). Each laser scanner has a horizontal offset from the axis of rotation, and a vertical offset from adjacent laser scanners. Additionally, each laser scanner points to a different elevation angle, such that, collectively, all the laser scanners cover a portion of the vertical field of view. Therefore, once the LIDAR's head completes a full rotation, each laser scanner has swept a cone in space specified by its elevation angle. Let $\{L\}$ be the LIDAR's fixed frame of reference whose z-axis is the axis of rotation of the sensor's head (see Fig. 4.1). Also, let $\{L_i\}$, $i = 1, \ldots, K$, be the coordinate frame corresponding to the i-th laser scanner, such that its origin is at the center of the associated cone on the z-axis of $\{L\}$ with vertical offset h_i from the origin of $\{L\}$, its z-axis aligned with that of $\{L\}$, and its x-axis defining an angle θ_{oi} with the x-axis of $\{L\}$. We determine the direction of the k-th shot of the i-th laser beam from its corresponding elevation angle, ϕ_i, and azimuth measurement, θ_{ik}, and denote it with:

$$
{}^{L_i}\bar{\mathbf{p}}_k \triangleq \begin{bmatrix} \cos\phi_i \cos\theta_{ik} \\ \cos\phi_i \sin\theta_{ik} \\ \sin\phi_i \end{bmatrix}. \tag{4.1}
$$

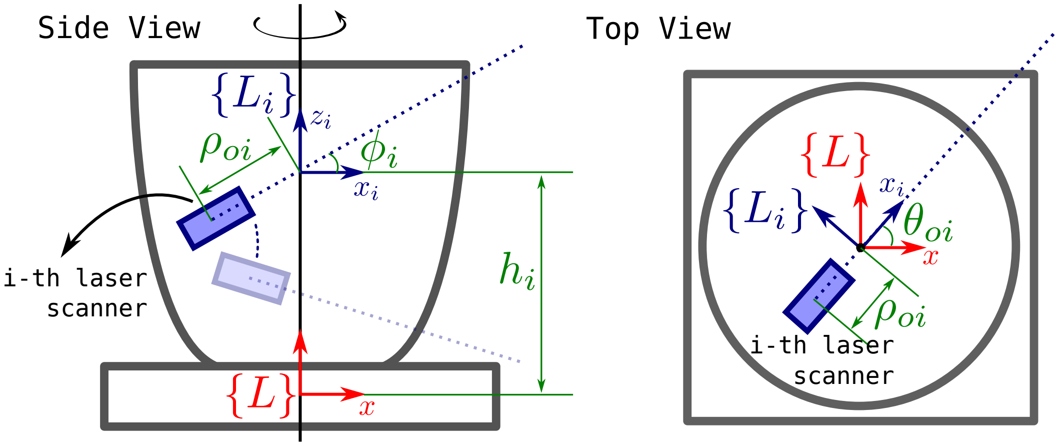


Figure 4.1: A revolving-head 3D LIDAR consists of K laser scanners, pointing to different elevation angles, and rotating around a common axis. The intrinsic parameters of the LIDAR describe the measurements of each laser scanner in its coordinate frame, $\{L_i\}$, and the transformation between the LIDAR's fixed coordinate frame, $\{L\}$, and $\{L_i\}$. Note that besides the physical offset of the laser scanners from the axis of rotation, the value of ρ_{oi} may depend on the delay in the electronic circuits of the LIDAR.

The distance measured by the k-th shot of the i-th laser scanner is represented by ρ_{ik}. The real distance to the object that reflects the k-th shot of the i-th laser beam is $\alpha_i(\rho_{ik} + \rho_{oi})$, where α_i is the scale factor, and ρ_{oi} is the range offset due to the delay in the electronic circuits of the LIDAR and the offset of each laser scanner from its cone's center. In this way, the position of the k-th point measured by the i-th laser scanner is described by

$$^{L_i}\mathbf{p}_{ik} = \alpha_i(\rho_{ik} + \rho_{oi})\,^{L_i}\bar{\mathbf{p}}_k. \tag{4.2}$$

The transformation between $\{L_i\}$ and $\{L\}$ (i.e., h_i and θ_{oi}), the scale α_i, offset ρ_{oi}, and elevation angle ϕ_i, for $i = 1, \ldots, K$, comprise the intrinsic parameters of the LIDAR that must be precisely known for any application, including photorealistic reconstruction of the surroundings. Since the intrinsic parameters supplied by the manufacturer are typically not accurate (except for the elevation angle ϕ_i), in this work we estimate them along with the transformation with respect to a camera.[1]

We assume that an *intrinsically calibrated* camera is rigidly connected to the LIDAR, and our objective is to determine the 6 d.o.f. relative transformation between the

[1] Note that when the technical drawings of the LIDAR are available, a coarse initial estimate for h_i, θ_{oi}, and ϕ_i can be readily obtained. Computing an initial estimate for ρ_{oi} and α_i, however, is significantly more challenging even for the manufacturer, since their values do not solely depend on the physical dimensions of the device.

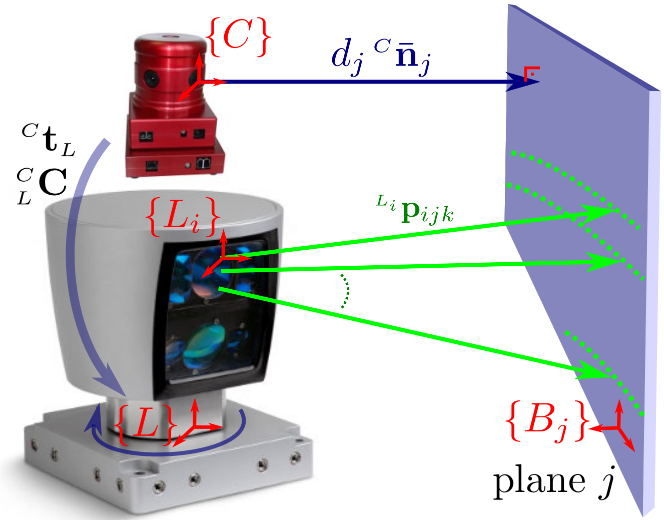


Figure 4.2: Geometric constraint between the j-th plane, the camera $\{C\}$, and the i-th laser scanner, $\{L_i\}$. Each laser beam is described by a vector ${}^{L_i}\mathbf{p}_{ijk}$. The plane is described by its normal vector ${}^C\bar{\mathbf{n}}_j$ and its distance d_j both expressed with respect to the camera.

two, as well as the intrinsic parameters of the LIDAR. For this purpose, we employ a planar calibration board with fiducial markers[2] at M different configurations to establish geometric constraints between the measurements of the LIDAR and the camera, their relative transformation, and the LIDAR's intrinsic parameters.

4.2.1 Noise-free Geometric Constraints

At the j-th configuration of the calibration board, $j = 1, \ldots, M$, (see Fig. 4.2), the fiducial markers whose positions are known with respect to the calibration board's frame of reference $\{B_j\}$, are first detected in the camera's image. The 6 d.o.f. transformation between $\{C\}$ and $\{B_j\}$ is then computed using a PnP algorithm [106, 3, 53], from which the normal vector and the distance of the target plane in the camera's frame are extracted as:

$$ {}^C\bar{\mathbf{n}}_j \triangleq {}^C_{B_j}\mathbf{C}\begin{bmatrix}0 & 0 & -1\end{bmatrix}^T \tag{4.3}$$

$$ d_j \triangleq {}^C\bar{\mathbf{n}}_j^T\,{}^C\mathbf{t}_{B_j} \tag{4.4}$$

where ${}^C_{B_j}\mathbf{C}$ and ${}^C\mathbf{t}_{B_j}$ represent the relative rotation and translation between the camera and the calibration board at the j-th configuration. Consequently, in the absence of

[2] For example, see Alvar fiducial markers at http://www.vtt.fi/multimedia/alvar.html

noise, any point ${}^{C}\mathbf{p}$ that lies on the j-th plane satisfies:

$$ {}^{C}\bar{\mathbf{n}}_j^T \, {}^{C}\mathbf{p} - d_j = 0. \tag{4.5} $$

We now turn our attention to the LIDAR point measurements reflected from the j-th calibration plane and identified based on the depth discontinuity. Let us denote such points as ${}^{L_i}\mathbf{p}_{ijk}$, $k = 1\ldots, N_{ij}$, measured by the LIDAR's i-th laser scanner [see (4.2)]. Transforming these points to the camera's frame, and substituting them in (4.5) yields:

$$ {}^{C}\bar{\mathbf{n}}_j^T \left({}^{C}_{L_i}\mathbf{C} \, {}^{L_i}\mathbf{p}_{ijk} + {}^{C}\mathbf{t}_{L_i} \right) - d_j = 0 \overset{(4.2)}{\Longrightarrow} \tag{4.6} $$

$$ \alpha_i(\rho_{ijk} + \rho_{oi}){}^{C}\bar{\mathbf{n}}_j^T \, {}^{C}_{L_i}\mathbf{C} \, {}^{L_i}\bar{\mathbf{p}}_{ijk} + {}^{C}\bar{\mathbf{n}}_j^T \, {}^{C}\mathbf{t}_{L_i} - d_j = 0 \tag{4.7} $$

where ${}^{C}_{L_i}\mathbf{C}$ and ${}^{C}\mathbf{t}_{L_i}$ are the relative rotation and translation between the camera and the i-th laser scanner.

4.2.2 Geometric Constraints in the Presence of Noise

In the presence of noise, the geometric constraint in (4.7) is not exactly satisfied. Instead, we will have:

$$ \alpha_i(\rho_{ijk} + \rho_{oi}){}^{C}\bar{\mathbf{n}}_j^T \, {}^{C}_{L_i}\mathbf{C} \, {}^{L_i}\bar{\mathbf{p}}_{ijk} + {}^{C}\bar{\mathbf{n}}_j^T \, {}^{C}\mathbf{t}_{L_i} - d_j = \epsilon_{ijk} \tag{4.8} $$

where ϵ_{ijk} is the residual due to the noise in the image and the LIDAR measurements. The covariance of this residual is:

$$ \begin{aligned} \sigma^2_{\epsilon_{ijk}} = & \left(\alpha_i {}^{C}\bar{\mathbf{n}}_j^T \, {}^{C}_{L_i}\mathbf{C} \, {}^{L_i}\bar{\mathbf{p}}_{ijk} \right)^2 \sigma_\rho^2 + \mathbf{h}_{p_{ijk}} \mathbf{R}_p \mathbf{h}_{p_{ijk}}^T \\ & + \mathbf{h}_n \mathbf{R}_{n_j} \mathbf{h}_n^T + \sigma_{d_j}^2 \end{aligned} \tag{4.9} $$

where σ_ρ is the standard deviation of noise in ρ_{ijk}, and σ_{d_j} is the standard deviation of the uncertainty in d_j. The covariance of the uncertainty in the laser beam directions, ${}^{L_i}\bar{\mathbf{p}}_{ijk}$, and the plane normal vector, ${}^{C}\bar{\mathbf{n}}_j$, expressed in their local tangent planes [48, Appendix 6.9.3], are represented by $\mathbf{R}_p$ and $\mathbf{R}_{n_j}$, respectively. The corresponding Jacobians are:

$$ \mathbf{h}_{p_{ijk}} = \alpha_i(\rho_{ijk} + \rho_{oi}){}^{C}\bar{\mathbf{n}}_j^T \, {}^{C}_{L_i}\mathbf{C} \, \mathbf{H}_{\bar{\mathbf{p}}_{ijk}} \begin{bmatrix} \mathbf{I}_2 & \mathbf{0}_{2\times 1} \end{bmatrix}^T \tag{4.10} $$

$$ \mathbf{h}_{n_j} = \left(\alpha_i(\rho_{ijk} + \rho_{oi}){}^{L_i}\bar{\mathbf{p}}_{ijk}^T \, {}^{C}_{L_i}\mathbf{C}^T + {}^{C}\mathbf{t}_{L_i}^T \right) \mathbf{H}_{\bar{\mathbf{n}}_j} \begin{bmatrix} \mathbf{I}_2 & \mathbf{0}_{2\times 1} \end{bmatrix}^T \tag{4.11} $$

where $\mathbf{H}_{\bar{\mathbf{u}}}$ is the 3×3 Householder matrix associated with the unit vector $\bar{\mathbf{u}}$ [45], defined as:

$$\mathbf{H}_{\bar{\mathbf{u}}} \triangleq \mathbf{I}_3 - 2\frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}, \quad \mathbf{v} \triangleq \bar{\mathbf{u}} + \text{sign}(\mathbf{e}_3^T\bar{\mathbf{u}})\mathbf{e}_3, \quad \mathbf{e}_3 \triangleq [0\ 0\ 1]^T \tag{4.12}$$

Note that the characteristics of ϵ_{ijk} depends not only on the uncertainty of the measurements, but also on the unknown calibration parameters.

4.2.3 Structural Constraints

In addition to the camera and laser scanner measurements, the following constraints can also be used to increase the accuracy of the calibration process. Specifically, since the z-axis of $\{L_i\}$ is aligned with the z-axis of $\{L\}$, while their x-axes form an angle θ_{oi}, the following constraint holds for all ${}^{C}_{L_i}\mathbf{C}$:

$${}^{C}_{L_i}\mathbf{C} = {}^{C}_{L}\mathbf{C}\mathbf{C}_z(\theta_{oi}) \tag{4.13}$$

where $\mathbf{C}_z(\theta_{oi})$ represents a rotation around the z-axis by an angle θ_{oi}. Additionally, the origin of each laser-scanner frame lies on the z-axis of $\{L\}$ with vertical offset of h_i from the origin of $\{L\}$, resulting in the following constraint:

$${}^{C}_{L}\mathbf{C}^T \left({}^{C}\mathbf{t}_{L_i} - {}^{C}\mathbf{t}_L\right) = [0\ 0\ h_i]^T \tag{4.14}$$

4.3 Algorithm Description

In order to estimate the unknown calibration parameters, we form a constrained non-linear least-squares cost function from the residuals of the geometric constraint over all point and plane observations [see (4.8)]. To minimize this least-squares cost, one has to employ iterative minimizers such as the Levenberg-Marquardt [105], that require a precise initial estimate to ensure convergence. To provide accurate initialization, in this section we present a novel analytical method to estimate the LIDAR-camera transformation and all intrinsic parameters of the LIDAR (except the elevation angle ϕ_i which is precisely known from the manufacturer). In order to reduce the complexity of the initialization process, we temporarily drop the constraints in (4.13) and (4.14) and seek to determine the transformation between the camera and each of the laser scanners (along with each scanner's intrinsic parameters) independently (see Sections 4.3.1-4.3.2). Once an accurate initial estimate is computed, we lastly perform an iterative non-linear

Algorithm 1 Estimate intrinsic LIDAR and extrinsic LIDAR-camera calibration parameters.

```
for $j$-th configuration of the calibration plane do
    Record an image and a LIDAR snapshot.
    Detect the known fiducial markers on the image.
    Compute ${}^C\bar{\mathbf{n}}_j$ and $d_j$ using a PnP algorithm.
    for $i$-th laser scanner do
        Identify the laser points hitting the calibration plane using depth discontinuity.
        Compute the contributions of $j$-th plane's observation to the rotation-offset
        optimality equations [see (4.23)].
    end for
end for
for $i$-th laser scanner do
    Solve the optimality equations in (4.23) to compute critical points of (4.21).
    Estimate $\hat{\mathbf{s}}_i$ and $\hat{\rho}_{oi}$ as the critical point that minimizes (4.22).
    Solve the linear least-squares problem in (4.27) to estimate ${}^C\mathbf{t}_{L_i}$ and $\alpha_i$.
end for
Refine the estimates for all the unknowns and enforce (4.13) and (4.14) by iteratively
minimizing (4.28).
```

least-squares refinement that explicitly considers (4.13) and (4.14), and increases the calibration accuracy (see Section 4.3.3).

4.3.1 Analytical Estimation of Offset and Relative Rotations

Note that the term ${}^C\bar{\mathbf{n}}_j^T\,{}^C\mathbf{t}_{L_i} - d_j$ in (4.7) is constant for all points k of the i-th laser scanner that hit the calibration plane at its j-th configuration. Therefore, subtracting two noise-free constraints of the form (4.7) for the points ${}^{L_i}\mathbf{p}_{ijk}$ and ${}^{L_i}\mathbf{p}_{ijl}$, and dividing the result by the nonzero scale, α_i, yields:

$$
{}^C\bar{\mathbf{n}}_j^T\,{}^C_{L_i}\mathbf{C}(\mathbf{u}^i_{jkl} + \rho_{oi}\mathbf{v}^i_{jkl}) = 0 \tag{4.15}
$$

where $\mathbf{u}^i_{jkl} \triangleq \rho_{ijk}\,{}^{L_i}\bar{\mathbf{p}}_{ijk} - \rho_{ijl}\,{}^{L_i}\bar{\mathbf{p}}_{ijl}$ and $\mathbf{v}^i_{jkl} \triangleq {}^{L_i}\bar{\mathbf{p}}_{ijk} - {}^{L_i}\bar{\mathbf{p}}_{ijl}$. Note that this constraint involves as unknowns only the relative rotation of the i-th laser scanner with respect to the camera, ${}^C_{L_i}\mathbf{C}$, and its offset, ρ_{oi}. Let us express the former, ${}^C_{L_i}\mathbf{C}$, using the Cayley-Gibbs-Rodriguez (CGR) parameterization [117], i.e.,

$$
{}^C_{L_i}\mathbf{C}(\mathbf{s}) = \frac{\bar{\mathbf{C}}(\mathbf{s}_i)}{1 + \mathbf{s}_i^T\mathbf{s}_i} \tag{4.16}
$$

$$
\bar{\mathbf{C}}(\mathbf{s}_i) \triangleq ((1 - \mathbf{s}_i^T\mathbf{s}_i)\mathbf{I}_3 + 2\lfloor \mathbf{s}_i \times \rfloor + 2\mathbf{s}_i\mathbf{s}_i^T) \tag{4.17}
$$

where $\mathbf{s}_i^T = [s_{i1} \;\; s_{i2} \;\; s_{i3}]$ is the vector of CGR parameters that represent the relative orientation of the i-th laser scanner with respect to the camera, and

$$\lfloor \mathbf{s} \times \rfloor \triangleq \begin{bmatrix} 0 & -s_3 & s_2 \\ s_3 & 0 & -s_1 \\ -s_2 & s_1 & 0 \end{bmatrix} \tag{4.18}$$

is the corresponding skew-symmetric matrix. Substituting (4.16) in (4.15), and multiplying both sides with the nonzero term $1 + \mathbf{s}_i^T \mathbf{s}_i$ yields:

$${}^C\bar{\mathbf{n}}_j^T \, \bar{\mathbf{C}}(\mathbf{s}_i)(\mathbf{u}_{jkl}^i + \rho_{oi}\mathbf{v}_{jkl}^i) = 0 \tag{4.19}$$

This algebraic constraint holds exactly in the absence of noise. In that case, the method presented in Section 4.5 can be employed to recover the unknowns given the minimum number of measurements. In the presence of noise, however, (4.19) becomes:

$${}^C\bar{\mathbf{n}}_j^T \, \bar{\mathbf{C}}(\mathbf{s}_i)(\mathbf{u}_{jkl}^i + \rho_{oi}\mathbf{v}_{jkl}^i) = \eta_{jkl}^i \tag{4.20}$$

where η_{jkl}^i is a nonzero residual. In this case, we estimate $\mathbf{s}_i$ and ρ_{oi} by solving the following nonlinear least-squares problem:

$$\hat{\mathbf{s}}_i, \hat{\rho}_{oi} = \min_{\mathbf{s}_i, \rho_{oi}} \mathcal{C}_i \tag{4.21}$$

$$\mathcal{C}_i \triangleq \frac{1}{2} \sum_{j=1}^{M} \sum_{k=1}^{\frac{N_{ij}}{2}} \sum_{l=\frac{N_{ij}}{2}+1}^{N_{ij}} \left({}^C\bar{\mathbf{n}}_j^T \, \bar{\mathbf{C}}(\mathbf{s}_i)(\mathbf{u}_{jkl}^i + \rho_{oi}\mathbf{v}_{jkl}^i) \right)^2 \tag{4.22}$$

where, without loss of generality, we have assumed N_{ij} is even. Note that the N_{ij} points from the i-th laser scanner, and the j-th configuration of the calibration plane are divided into two mutually exclusive groups so as to ensure that each point appears in the least-squares cost only once and hence avoid noise correlations.

When a sufficient number of plane configurations are observed, we employ a recently proposed algebraic method to directly solve this nonlinear least-squares problem without requiring initialization [125]. Specifically, we first form the following polynomial system describing the optimality conditions of (4.21):

$$f_{i\ell} = \frac{\partial \mathcal{C}_i}{\partial s_{i\ell}} = \sum_{j=1}^{M} \sum_{k=1}^{\frac{N_{ij}}{2}} \sum_{l=\frac{N_{ij}}{2}+1}^{N_{ij}} \left({}^{C}\bar{\mathbf{n}}_j^T \, \bar{\mathbf{C}}(\mathbf{s}_i)(\mathbf{u}_{jkl}^i + \rho_{oi}\mathbf{v}_{jkl}^i) \right)$$
$$\cdot \underbrace{\frac{\partial}{\partial s_{i\ell}} \left({}^{C}\bar{\mathbf{n}}_j^T \, \bar{\mathbf{C}}(\mathbf{s}_i)(\mathbf{u}_{jkl}^i + \rho_{oi}\mathbf{v}_{jkl}^i) \right)}_{J_{i\ell}} = 0, \quad \ell = 0, \ldots, 3 \tag{4.23}$$

For $\ell = 1, 2, 3,\ J_{i\ell}$ is

$$J_{i\ell} = {}^{C}\bar{\mathbf{n}}_j^T \, \mathbf{D}_\ell(\mathbf{s}_i)(\mathbf{u}_{jkl}^i + \rho_{oi}\mathbf{v}_{jkl}^i) \tag{4.24}$$

where

$$\mathbf{D}_\ell(\mathbf{s}_i) = -2s_{i\ell}\mathbf{I}_3 + 2\lfloor \mathbf{e}_\ell \times \rfloor + 2\mathbf{e}_\ell \mathbf{s}_i^T + 2\mathbf{s}_i \mathbf{e}_\ell^T \tag{4.25}$$
$$\mathbf{e}_1 \triangleq [1\ 0\ 0]^T, \quad \mathbf{e}_2 \triangleq [0\ 1\ 0]^T, \quad \mathbf{e}_3 \triangleq [0\ 0\ 1]^T \tag{4.26}$$

and for $\ell = 0$, i.e., $s_{i0} \triangleq \rho_{oi}$, $J_{i0} = {}^{C}\bar{\mathbf{n}}_j^T \, \bar{\mathbf{C}}(\mathbf{s}_i)\, \mathbf{v}_{jkl}^i$. Note that the cost function in (4.22) is a polynomial of degree six in the elements of $\mathbf{s}_i$ and ρ_{oi}. Therefore, (4.23) consists of four polynomials of degree five in four variables. This polynomial system has 243 solutions that comprise the critical points of the least-squares cost function $\mathcal{C}_i$, and can be computed using the eigenvalue decomposition of the so-called *multiplication matrix* (see Section 4.4). The globally optimal solution of the least-squares problem is the critical point that minimizes (4.22), and it is selected through direct evaluation of the cost function $\mathcal{C}_i$. We point out that the computational complexity of solving (4.23) and finding the global minimum does not increase with the addition of measurements, since the degree and number of polynomials expressing the optimality conditions are fixed regardless of the number of calibration-plane configurations and laser-scanner points reflected from them. Moreover, computing the contribution of all points to the coefficients of the polynomials $f_{i\ell}$, $\ell = 0, \ldots, 3$, increases only linearly with the number of measurements.

4.3.2 Analytical Estimation of Scale and Relative Translation

Once the relative rotation, ${}^{C}_{L_i}\mathbf{C}$, and offset, ρ_{oi}, of each laser scanner, $i = 1, \ldots, K$, is computed, we use linear least-squares to determine the relative translation and scale from (4.7). Specifically, we stack together all the measurement constraints on the i-th laser scanner's scale and relative translation (from different points and calibration-plane configurations), and write them in a matrix form as:

$$\begin{bmatrix} {}^C\bar{\mathbf{n}}_1^T & (\rho_{i11}+\rho_{oi}){}^C\bar{\mathbf{n}}_1^T\,{}^C_{L_i}\mathbf{C}\,{}^{L_i}\bar{\mathbf{p}}_{i11} \\ {}^C\bar{\mathbf{n}}_1^T & (\rho_{i12}+\rho_{oi}){}^C\bar{\mathbf{n}}_1^T\,{}^C_{L_i}\mathbf{C}\,{}^{L_i}\bar{\mathbf{p}}_{i12} \\ \vdots & \vdots \\ {}^C\bar{\mathbf{n}}_M^T & (\rho_{iMN_{iM}}+\rho_{oi}){}^C\bar{\mathbf{n}}_M^T\,{}^C_{L_i}\mathbf{C}\,{}^{L_i}\bar{\mathbf{p}}_{iMN_{iM}} \end{bmatrix} \begin{bmatrix} {}^C\mathbf{t}_{L_i} \\ \alpha_i \end{bmatrix} = \begin{bmatrix} d_1 \\ d_1 \\ \vdots \\ d_M \end{bmatrix} \tag{4.27}$$

Under the condition that the coefficient matrix on the left-hand side of this equality is full rank (see Section 4.5), we can easily obtain the i-th laser scanner's scale factor, α_i, and relative translation, ${}^C\mathbf{t}_{L_i}$, by solving (4.27).

4.3.3 Iterative Refinement

Once the initial estimates for the transformation between the camera and the laser scanners, and the intrinsic parameters of the LIDAR are known (Sections 4.3.1 to 4.3.2), we employ an iterative refinement method to enforce the constraints in (4.13) and (4.14). Specifically, we choose the coordinate frame of one of the laser scanners (e.g., the 1-st laser scanner) as the LIDAR's fixed coordinate frame, i.e., $\{L\} = \{L_1\}$. Then for $\{L_i\},\ i = 2, \ldots, K$, we employ the estimated relative transformation with respect to the camera (i.e., ${}^C_{L_i}\mathbf{C}$ and ${}^C\mathbf{t}_{L_i}$) to obtain the relative transformations between $\{L_i\}$ and $\{L\}$. From these relative transformations, we only use the z component of the translation to initialize each laser scanner's vertical offset, h_i [see (4.14)], and the *yaw* component of the rotation to initialize each laser scanner's θ_{oi} [see (4.13)].

We then formulate the following constrained minimization problem to enforce (4.13) and (4.14):

$$\begin{aligned} \min \sum_{i,j,k} & \frac{\left[\alpha_i(\rho_{ijk}+\rho_{oi}){}^C\bar{\mathbf{n}}_j^T\,{}^C_{L_i}\mathbf{C}\,{}^{L_i}\bar{\mathbf{p}}_{ijk} + {}^C\bar{\mathbf{n}}_j^T\,{}^C\mathbf{t}_{L_i} - d_j\right]^2}{\sigma^2_{\epsilon_{ijk}}} \\ & \text{subject to: } {}^C_{L_i}\mathbf{C} = {}^C_L\mathbf{C}\mathbf{C}_z(\theta_{oi}) \\ & \qquad {}^C_L\mathbf{C}^T\left({}^C\mathbf{t}_{L_i} - {}^C\mathbf{t}_L\right) = [0\ 0\ h_i]^T \end{aligned} \tag{4.28}$$

where the optimization variables are $\alpha_i,\ \rho_{oi},\ \theta_{oi},\ h_i,\ i = 2, \ldots, K,\ \alpha_1,\ \rho_{o1}, {}^C\mathbf{t}_L, {}^C_L\mathbf{C}$, and $\sigma^2_{\epsilon_{ijk}}$ is defined in (4.9).[3] Note that the constraints in (4.28) should be taken into account

[3] In general, the optimization should be performed over ϕ_i as well. However, in our experiments, we observed that the provided value of ϕ_i by the manufacturer is sufficiently accurate, and thus excluded

using the method of Lagrange multipliers. Alternatively, we minimized a reformulation of (4.28) that uses a minimal parameterization of the unknowns to avoid the constraints (and hence the Lagrange multipliers). The details of this alternative cost function are provided in Appendix B.

4.4 Polynomial System Solver

Several methods exist for solving the polynomials describing the optimality conditions of (4.23). Amongst them, numerical methods, such as Newton-Raphson, need proper initialization and may not find all the solutions. Symbolic reduction methods based on the computation of the system's Gröbner basis are capable of finding all roots without any initialization [30]. However, they can only be used for integer or rational coefficients since their application to floating-point numbers suffers from quick accumulation of round-off errors, which in turn, results in incorrect solutions [30]. Instead, we employ a method developed by [8] that computes a generalization of the companion matrix to systems of multivariate polynomial equations, namely the *multiplication matrix*, whose eigenvalues are the roots of the associated polynomial system. In the following, we briefly describe an efficient method for computing the multiplication matrix.

Let us denote a *monomial* in $\mathbf{x} = [x_1 \cdots x_n]^T$ by $x^{\boldsymbol{\gamma}} \triangleq x_1^{\gamma_1} x_2^{\gamma_2} \cdots x_n^{\gamma_n}$, $\gamma_i \in \mathbb{Z}_{\geq 0}$, with degree $\sum_{i=1}^{n} \gamma_i$. A polynomial of degree d in $\mathbf{x}$ is denoted by $f = \mathbf{c}^T \mathbf{x}_d$ where $\mathbf{x}_d$ is the $\binom{n+d}{n}$-dimensional vector of monomials of degree up to and including d, and $\mathbf{c}$ is the vector of coefficients of equal size. We assume that the given system of equations has n polynomials, denoted by $f_i = \mathbf{c}_i^T \mathbf{x}_{d_i} = 0,\ i = 1, \ldots, n$, each of them with degree d_i. The total degree of the polynomial system is $d \triangleq \max_i d_i$. By padding the coefficient vectors of f_i's with zeros, and stacking them together in $\mathbf{C}$, we can present the polynomial system in the matrix form of $\mathbf{C}\mathbf{x}_d = \mathbf{0}$.

A system of polynomial equations defines an *ideal* I as the set of all the polynomials that can be generated as $\sum_i f_i h_i$ where h_i is any polynomial in $\mathbf{x}$. Clearly the elements of the ideal become zero at the solutions of the original (generator) polynomial system. The Gröbner basis $G \triangleq \langle g_1, \ldots g_t \rangle$ of an ideal is a finite subset of the ideal such that (i) the remainder of the division of any polynomial to it is unique, (ii) any polynomial whose division by the Gröbner basis results in zero remainder, is a member of the associated

it from the calibration.

ideal. The first property can be expressed as:

$$\varphi(\mathbf{x}) = r(\mathbf{x}) + \sum_{i=1}^{t} g_i(\mathbf{x})h_i(\mathbf{x}) \tag{4.29}$$

where φ is any polynomial in $\mathbf{x}$, h_i's are the quotient polynomials, and r is the unique remainder. We hereafter use the name "remainder" as the remainder of the division of a polynomial by the Gröbner basis. The Gröbner basis for an ideal generated from polynomials with integer or rational numbers can be computed using implementations of the so-called Buchberger's algorithm [30] on symbolic software packages such as Macaulay2 or Maple. Computation of the Gröbner basis for polynomials with floating-point coefficients is much more difficult due to quick accumulation of round-off errors in the Buchberger's algorithm.

The remainders of the polynomials that are not in an ideal are instrumental in finding the solutions (i.e., variety) of that ideal. It can be shown that all such remainders can be expressed as a linear combination of a specific (unique) group of monomials that comprise the so-called *normal set* [30]. The normal set can be easily obtained from the Gröbner basis of an ideal, and under mild conditions,[4] its cardinality equals the number of solutions (real and complex) of the ideal, and it will contain the monomial 1 [30, p.43]. The important point here is that the normal set is generically fixed across different instantiations of the polynomials. Therefore, we can compute the normal set of an instance of the problem (e.g., integer or rational coefficients) and use it when the coefficients are floating point.

Example. Consider the following simple example polynomials in $\mathbf{x} = [x_1 \quad x_2]^T$:

$$f_1 = x_1 + x_1x_2 + 5 \tag{4.30}$$
$$f_2 = x_1^2 + x_2^2 - 10 \tag{4.31}$$

These equations are of degree $d_1 = d_2 = 2$. The Gröbner basis of this polynomial system (using graded reverse lex ordering [30]) is:

$$g_1 = x_1x_2 + x_1 + 5 \tag{4.32}$$
$$g_2 = x_1^2 + x_2^2 - 10 \tag{4.33}$$
$$g_3 = x_2^3 + x_2^2 - 5x_1 - 10x_2 - 10 \tag{4.34}$$

[4] These conditions are: (i) the ideal must be radical, (ii) its variety must be non-empty and zero dimensional [30]. These conditions are generally satisfied for the current problem.

and, consequently its normal set is:

$$\{1,\ x_2,\ x_1,\ x_2^2\} \tag{4.35}$$

Note that this normal set is *generically*[5] the same for different coefficients of the system in (4.30)-(4.31). For example the following system yields the same normal set:

$$f_1' = 1.5x_1 + e^{-1}x_2x_1 + 0.5 \tag{4.36}$$

$$f_2' = 2.3x_1^2 + \frac{4}{3}x_2^2 - \pi \tag{4.37}$$

■

Let us assume that the cardinality of the normal set is s, and represent its monomials in a vector form $\mathbf{x}_B$. Then multiplication of $\mathbf{x}_B$ with a generic polynomial $\varphi(\mathbf{x})$ yields [see (4.29)]:

$$\varphi(\mathbf{x}) \cdot \mathbf{x}_B = \mathbf{M}_\varphi \mathbf{x}_B + \begin{bmatrix} h_{11} & \cdots & h_{1t} \\ \vdots & & \vdots \\ h_{s1} & \cdots & h_{st} \end{bmatrix} \begin{bmatrix} g_1 \\ \vdots \\ g_t \end{bmatrix} \tag{4.38}$$

where h_{ij}'s are polynomials in $\mathbf{x}$, and g_i's are the elements of the Gröbner basis. In this expression, $\mathbf{M}_\varphi$ is called the *multiplication matrix* associated with φ. This relationship holds since the remainder of any polynomial (including $x^\gamma \varphi(\mathbf{x})$, $x^\gamma \in \mathbf{x}_B$) can be written as a linear combination of $\mathbf{x}_B$. Now, if we evaluate (4.38) at $\mathbf{x} = \mathbf{p}$, a solution of the ideal, all g_i's become zero, and we get:

$$\varphi(\mathbf{p}) \cdot \mathbf{p}_B = \mathbf{M}_\varphi \mathbf{p}_B \tag{4.39}$$

where $\mathbf{p}_B$ is $\mathbf{x}_B$ evaluated at $\mathbf{p}$. Clearly, $\mathbf{p}_B$ is an eigenvector of $\mathbf{M}_\varphi$, and $\varphi(\mathbf{p})$ is the associated eigenvalue. Therefore, if we select $\varphi(\mathbf{x})$ equal to one of the variables (e.g., x_i), we can read off the x_i-coordinate of the solutions as the eigenvalues of $\mathbf{M}_\varphi$. Furthermore, depending on the ordering of the monomials when computing the Gröbner basis, $\mathbf{x}_B$ may include all first-order monomials $x_1, \ldots, x_n$. In that case, one can simultaneously read off all the coordinates of the solutions, for an arbitrary choice of φ, as long as it is nonzero and distinct at each solution of the ideal.

When the Gröbner basis is available (such as in polynomial systems with integer coefficients), one can use it directly to compute remainders of $\varphi(\mathbf{x}) \cdot \mathbf{x}_B$, and construct

[5] In other words, except for singular choices of the coefficients (e.g., zero coefficients), the normal set remains the same. For a more precise definition of *genericity*, please refer to [30].

$\mathbf{M}_\varphi$. This is not possible, however, when working with polynomials with floating-point coefficients. Therefore we employ the method proposed by [20] to compute $\mathbf{M}_\varphi$. We first note that some of the monomials of $\varphi(\mathbf{x}) \cdot \mathbf{x}_B$ remain in $\mathbf{x}_B$, while some others do not. We form the vector $\mathbf{x}_R$ from the latter monomials, and write:

$$\varphi(\mathbf{x}) \cdot \mathbf{x}_B = \mathbf{M}'_\varphi \begin{bmatrix} \mathbf{x}_R \\ \mathbf{x}_B \end{bmatrix} \tag{4.40}$$

where $\mathbf{M}'_\varphi$ is called the unreduced multiplication matrix. Our objective is to express the remainders of $\mathbf{x}_R$ as a linear combination of $\mathbf{x}_B$ without using the Gröbner basis. For this purpose, we expand each original polynomial f_i by multiplying it with all the monomials up to degree $\ell - d_i$ (ℓ to be determined later). Clearly all these new expanded polynomials belong to the ideal generated by the original polynomials, and they have monomials up to degree ℓ. Thus, we can write them collectively in matrix form as $\mathbf{C}_e \mathbf{x}_\ell = 0$. We reorder $\mathbf{x}_\ell$ and $\mathbf{C}_e$ as:

$$\mathbf{C}_e \mathbf{x}_\ell = \begin{bmatrix} \mathbf{C}_E & \mathbf{C}_R & \mathbf{C}_B \end{bmatrix} \begin{bmatrix} \mathbf{x}_E \\ \mathbf{x}_R \\ \mathbf{x}_B \end{bmatrix} = 0 \tag{4.41}$$

where $\mathbf{x}_E$ are the monomials that belong neither to $\mathbf{x}_R$ nor to $\mathbf{x}_B$. Multiplying (4.41) with $\mathbf{N}^T$, the left null space of $\mathbf{C}_E$, and decomposing $\mathbf{N}^T \mathbf{C}_R = \mathbf{Q}\mathbf{R} = [\mathbf{Q}_1 \ \mathbf{Q}_2] \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix}$ using QR factorization, yields:

$$\begin{bmatrix} \mathbf{N}^T \mathbf{C}_R & \mathbf{N}^T \mathbf{C}_B \end{bmatrix} \begin{bmatrix} \mathbf{x}_R \\ \mathbf{x}_B \end{bmatrix} = \mathbf{Q} \begin{bmatrix} \mathbf{R}_1 & \mathbf{Q}_1^T \mathbf{N}^T \mathbf{C}_B \\ \mathbf{0} & \mathbf{Q}_2^T \mathbf{N}^T \mathbf{C}_B \end{bmatrix} \begin{bmatrix} \mathbf{x}_R \\ \mathbf{x}_B \end{bmatrix} = \mathbf{0}. \tag{4.42}$$

If ℓ is selected sufficiently large, $\mathbf{R}_1$ will be full rank [107], which allows us to solve (4.42) and find $\mathbf{x}_R$ as a function of $\mathbf{x}_B$, i.e., $\mathbf{x}_R = -\mathbf{R}_1^{-1} \mathbf{Q}_1^T \mathbf{N}^T \mathbf{C}_B \mathbf{x}_B$. Substituting this relationship in (4.40) yields the multiplication matrix:

$$\mathbf{M}_\varphi = \mathbf{M}'_\varphi \begin{bmatrix} \mathbf{I}_s \\ -\mathbf{R}_1^{-1} \mathbf{Q}_1^T \mathbf{N}^T \mathbf{C}_B \end{bmatrix}. \tag{4.43}$$

For solving equations (4.23), we had to expand the polynomials up to degree $\ell = 15$ and arrived at a multiplication matrix $\mathbf{M}_\varphi$ of dimensions 243×243. Finally, we mention that it is possible to compute the multiplication matrix without explicit computation of the normal set. Further details on this subject and also on possible numerical instabilities and their remedies are given in [20, 107, 127].

Example. Let us arrange the normal set for the previous example in the vector form $\mathbf{x}_B = [1,\ x_2,\ x_1,\ x_2^2]^T$ and choose $\varphi(\mathbf{x}) = x_2$. Then multiplying $\varphi(\mathbf{x})$ with $\mathbf{x}_B$ and expressing the result in terms of $\mathbf{x}_B$ and $\mathbf{x}_R$ [see (4.40)] yields:

$$\varphi(\mathbf{x}) \cdot \mathbf{x}_B = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \underbrace{\begin{bmatrix} 1 \\ x_2 \\ x_1 \\ x_2^2 \end{bmatrix}}_{\mathbf{x}_B} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} x_1 x_2 \\ x_2^3 \end{bmatrix}}_{\mathbf{x}_R} \tag{4.44}$$

In order to express $\mathbf{x}_R$ in terms of $\mathbf{x}_B$, we expand the polynomials f_1 and f_2 up to degree $\ell = 3$ by multiplying each of them with $\{1,\ x_2,\ x_1\}$. As a result, we obtain $\mathbf{C}_e = [\mathbf{C}_E\ \ \mathbf{C}_R\ \ \mathbf{C}_B]$ where:

$$\mathbf{C}_E = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix},\ \mathbf{C}_R = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix},\ \mathbf{C}_B = \begin{bmatrix} 5 & 0 & 1 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ -10 & 0 & 0 & 1 \\ 0 & -10 & 0 & 0 \\ 0 & 0 & -10 & 0 \end{bmatrix}$$

Note that $\mathbf{C}_E$ corresponds to $\mathbf{x}_E = [x_1 x_2^2\ \ x_1^2 x_2\ \ x_1^2\ \ x_1^3]^T$, i.e., the monomials that appear neither in $\mathbf{x}_B$ nor in $\mathbf{x}_R$. Following the algebraic manipulations of (4.40)-(4.42), we obtain the following multiplication matrix:

$$\mathbf{M}_{x_2} = \begin{bmatrix} 0 & 0 & -5 & 10 \\ 1 & 0 & 0 & 10 \\ 0 & 0 & -1 & 5 \\ 0 & 1 & 0 & -1 \end{bmatrix} \tag{4.45}$$

In the next step, we compute the left eigenvectors of $\mathbf{M}_{x_2}$, and then scale them such that their first elements become 1 (corresponding to the first element in $\mathbf{x}_B$). Consequently, the solutions of the polynomial system are the elements of the eigenvectors that correspond to x_1 and x_2 in $\mathbf{x}_B$. Specifically, the set of solutions (i.e., variety) of

(4.30)-(4.31) is:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \left\{ \begin{bmatrix} -1.2856 \\ 2.8891 \end{bmatrix}, \begin{bmatrix} -3.1026 \\ 0.6116 \end{bmatrix}, \begin{bmatrix} 2.1941 + 1.2056i \\ -2.7504 + 0.9618i \end{bmatrix}, \begin{bmatrix} 2.1941 - 1.2056i \\ -2.7504 - 0.9618i \end{bmatrix} \right\} \tag{4.46}$$

■

4.5 Observability Conditions

In this section, we examine the conditions under which the unknown LIDAR-camera transformation and the intrinsic parameters of the LIDAR are identifiable, and thus can be estimated using the algorithms in Sections 4.3.1 to 4.3.3.

4.5.1 Observation of One Plane

Suppose we are provided with LIDAR measurements that lie only on one plane whose normal vector is denoted as ${}^{C}\bar{\mathbf{n}}_1$. In this case, it is easy to show that the measurement constraint in (4.6) does not change if ${}^{C}_{L_i}\mathbf{C}$ is perturbed by a rotation around ${}^{C}\bar{\mathbf{n}}_1$, represented by the rotation matrix $\mathbf{C}'$:

$$\begin{aligned} {}^{C}\bar{\mathbf{n}}_1^T\, \mathbf{C}'\, {}^{C}_{L_i}\mathbf{C}\, {}^{L_i}\mathbf{p}_{i1k} + {}^{C}\bar{\mathbf{n}}_1^T\, {}^{C}\mathbf{t}_{L_i} - d_1 = 0 \qquad (4.47) \\ \Longrightarrow \quad {}^{C}\bar{\mathbf{n}}_1^T\, {}^{C}_{L_i}\mathbf{C}\, {}^{L_i}\mathbf{p}_{i1k} + {}^{C}\bar{\mathbf{n}}_1^T\, {}^{C}\mathbf{t}_{L_i} - d_1 = 0. \qquad (4.48) \end{aligned}$$

The second equation is obtained from the first, since ${}^{C}\bar{\mathbf{n}}_1$ is an eigenvector of $\mathbf{C}'$, thus ${}^{C}\bar{\mathbf{n}}_1^T\mathbf{C}' = {}^{C}\bar{\mathbf{n}}_1^T$. Therefore, when observing only one plane, any rotation around the plane's normal vector is unidentifiable. Similarly, if we perturb ${}^{C}\mathbf{t}_{L_i}$ by a translation parallel to the plane, represented by $\mathbf{t}'$, the measurement constraint does not change:

$$\begin{aligned} {}^{C}\bar{\mathbf{n}}_1^T\, {}^{C}_{L_i}\mathbf{C}\, {}^{L_i}\mathbf{p}_{i1k} + {}^{C}\bar{\mathbf{n}}_1^T\, ({}^{C}\mathbf{t}_{L_i} + \mathbf{t}') - d_1 = 0 \qquad (4.49) \\ \Longrightarrow \quad {}^{C}\bar{\mathbf{n}}_1^T\, {}^{C}_{L_i}\mathbf{C}\, {}^{L_i}\mathbf{p}_{i1k} + {}^{C}\bar{\mathbf{n}}_1^T\, {}^{C}\mathbf{t}_{L_i} - d_1 = 0. \qquad (4.50) \end{aligned}$$

This relationship holds since ${}^{C}\bar{\mathbf{n}}_1^T\, \mathbf{t}' = 0$. Therefore, when observing only one plane, any translation parallel to the plane's normal is unidentifiable.

4.5.2 Observation of Two Planes

Consider now that we are provided with measurements from two planes, described by ${}^{C}\bar{\mathbf{n}}_1,\ d_1,\ {}^{C}\bar{\mathbf{n}}_2,\ d_2$. If we perturb the laser scanner's relative translation with $\mathbf{t}'' \propto {}^{C}\bar{\mathbf{n}}_1 \times {}^{C}\bar{\mathbf{n}}_2$ [see (4.49)], none of the measurement constraints will change, since ${}^{C}\bar{\mathbf{n}}_1^T\,\mathbf{t}'' = {}^{C}\bar{\mathbf{n}}_2^T\,\mathbf{t}'' = 0$. Therefore, we conclude that the relative translation cannot be determined if only two planes are observed.

4.5.3 Observation of Three Planes

In this section, we prove that when three planes with linearly independent normal vectors are observed, we can determine all the unknowns. For this purpose, we first determine the relative orientation ${}^{C}_{L_i}\mathbf{C}$ and the offset ρ_{oi} and then find the scale α_i and relative translation ${}^{C}\mathbf{t}_{L_i}$. Let us assume that the i-th laser scanner has measured four points on each plane, denoted as $(\rho_{ijk},\, {}^{L_i}\bar{\mathbf{p}}_{ijk}),\ j = 1,2,3,\ k = 1,\ldots,4$. Each of these points provides one constraint of the form (4.7). We first eliminate the unknown relative translation and scale, by subtracting the constraints for point $k = 1$ from $k = 2$, point $k = 2$ from $k = 3$, and point $k = 3$ from $k = 4$, and obtain:

$$ {}^{C}\bar{\mathbf{n}}_j^T\,{}^{C}_{L_i}\mathbf{C}\,\left(\mathbf{u}^i_{j12} + \rho_{oi}\,\mathbf{v}^i_{j12}\right) = 0 \tag{4.51} $$

$$ {}^{C}\bar{\mathbf{n}}_j^T\,{}^{C}_{L_i}\mathbf{C}\,\left(\mathbf{u}^i_{j23} + \rho_{oi}\,\mathbf{v}^i_{j23}\right) = 0 \tag{4.52} $$

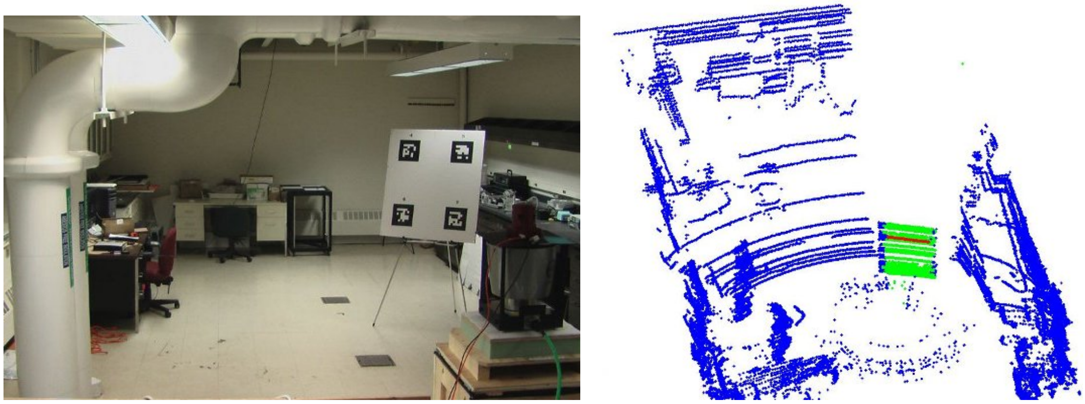
$$ {}^{C}\bar{\mathbf{n}}_j^T\,{}^{C}_{L_i}\mathbf{C}\,\left(\mathbf{u}^i_{j34} + \rho_{oi}\,\mathbf{v}^i_{j34}\right) = 0 \tag{4.53} $$

where $\mathbf{u}^i_{jkl} \triangleq \rho_{ijk}\,{}^{L_i}\bar{\mathbf{p}}_{ijk} - \rho_{ijl}\,{}^{L_i}\bar{\mathbf{p}}_{ijl}$, $\mathbf{v}^i_{jkl} \triangleq {}^{L_i}\bar{\mathbf{p}}_{ijk} - {}^{L_i}\bar{\mathbf{p}}_{ijl}$, and $j = 1,2,3$. Note that ${}^{L_i}\bar{\mathbf{p}}_{ijk}$ and ${}^{L_i}\bar{\mathbf{p}}_{ijl}$ lie on the intersection of the unit sphere and the cone specified by the beams of the i-th laser scanner. Since the intersection of a co-centric unit sphere and a cone is always a circle, we conclude that all $\mathbf{v}^i_{jkl}$ for a given i belong to a plane and have only two degrees of freedom. Thus, we can write $\mathbf{v}^i_{j34}$ as a linear combination of $\mathbf{v}^i_{j12}$ and $\mathbf{v}^i_{j23}$, i.e.,

$$ \mathbf{v}^i_{j34} = a\,\mathbf{v}^i_{j12} + b\,\mathbf{v}^i_{j23} \tag{4.54} $$

for some known scalars a and b. Substituting this relationship in (4.53), and using (4.51)-(4.52) to eliminate the terms containing ρ_{oi} yields:

$$ {}^{C}\bar{\mathbf{n}}_j^T\,{}^{C}_{L_i}\mathbf{C}\,\left(\mathbf{u}^i_{j34} - a\,\mathbf{u}^i_{j12} - b\,\mathbf{u}^i_{j23}\right) = 0 \tag{4.55} $$



(a)

(b)

Figure 4.3: (a): A view of the calibration environment. Note the Velodyne-Ladybug pair at the bottom-right of the picture. The configuration (i.e., position and orientation) of the calibration board (center-right of the picture) changed for each data capture; (b): A typical LIDAR snapshot, constructed using the LIDAR's intrinsic parameters provided by the manufacturer. The extracted calibration plane is shown in green. The red dots specify the extracted points corresponding to the laser scanner 20.

for $j = 1, 2, 3$. The only unknown in this equation is the relative orientation ${}^{C}_{L_i}\mathbf{C}$ of the i-th laser scanner. These equations are identical to those for orientation estimation using line-to-plane correspondences, which is known to have at most eight solutions that can be analytically computed when ${}^{C}\bar{\mathbf{n}}_j,\ j = 1, 2, 3$, are linearly independent [26]. Once ${}^{C}_{L_i}\mathbf{C}$ is known, we can use any of (4.51)-(4.53) to compute the offset ρ_{oi}. Finally, the scale and the relative translation can be obtained from (4.27).

4.6 Experiments

4.6.1 Setup

In order to validate the proposed calibration method, we conducted a series of experiments using a Velodyne revolving-head 3D LIDAR and a Ladybug2 spherical vision system. The Velodyne consists of 64 laser scanners that collectively span 27° of the vertical field of view. The Ladybug consists of six rigidly connected and intrinsically calibrated cameras equipped with wide-angle lenses (see Fig. 4.2). The extrinsic calibration between the different cameras is also provided by the manufacturer with high accuracy. Therefore, the measurements from any of the cameras can be easily transformed to the Ladybug's fixed frame of reference. We rigidly connected the Velodyne and the Ladybug, and recorded measurements of a 36" × 40" calibration plane at 18

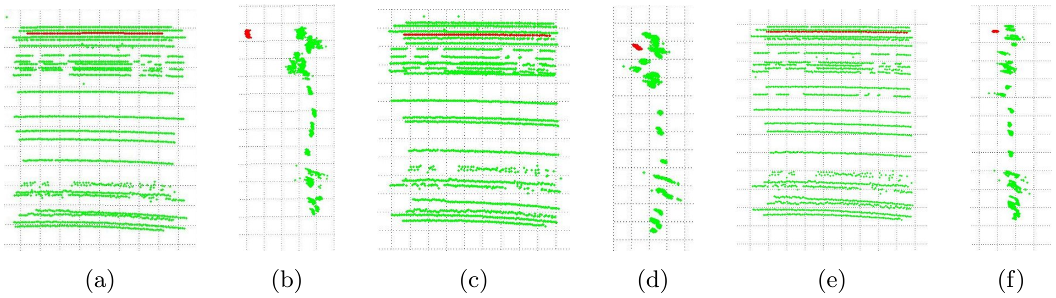


Figure 4.4: Consistency of the intrinsic parameters. (a,b): LIDAR points reflected from the calibration plane in a test dataset viewed from front and side. The points' Euclidean coordinates were computed using the **Factory** parameters. Note the considerable bias of the points from the laser scanner 20 shown in red (grid size 10 cm); (c,d): The same LIDAR points, front and side view, when their Euclidean coordinates are computed using the **PMSE** intrinsic parameters; (e,f): The same LIDAR points, front and side view, when their Euclidean coordinates are computed using the **AlgBLS** intrinsic parameters. Note that the points from the laser scanner 20 (shown in red) no longer exhibit a significant bias (grid size 10 cm).

different configurations (see Fig. 4.3). By processing the Ladybug's images using the PnP algorithm of [53], we computed the normal vector and the distance of the calibration plane at each configuration. We then identified the approximate location of the calibration plane in the LIDAR scans based on a coarse prior estimate for the relative rotation of the Velodyne and the Ladybug. Within these approximate locations, we detected the LIDAR data points reflected from the calibration plane, based on their depth discontinuity.

Once the Velodyne's measurements for each configuration of the calibration plane were available, we used the method described in Section 4.2 to accurately estimate the LIDAR's intrinsic parameters and the LIDAR-camera transformation. Note, however, that in order to increase the robustness of our algorithm to outliers, we did not directly use the raw laser points measured by the LIDAR. Instead, for each laser scanner, we fit small line segments to the intersection of the laser scanner's beam and the calibration plane, and used the endpoints of these line segments as the LIDAR's measurements.[6]

4.6.2 Implemented Methods

We compared the accuracy and consistency of the calibration parameters estimated by our proposed algorithm (denoted as **AlgBLS**), with the results of the approach presented by [103] (denoted as **PMSE**), and with those when using the calibration

[6] Note that in general the intersection of the cone induced by the laser scanner's beam with a plane results in a conic section, and not a straight line. However, in practical situations this conic section can be well approximated with consecutive straight line segments.

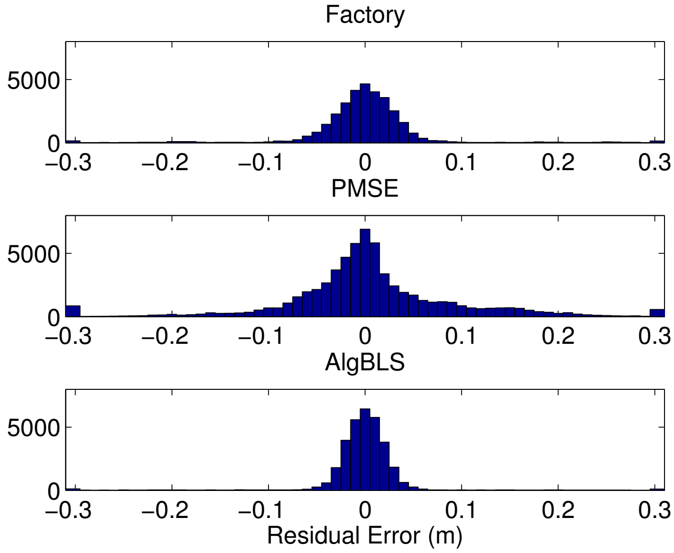


Figure 4.5: Histograms of the signed distance between laser points reflected from the calibration target and the corresponding fitted plane. The laser points' Euclidean coordinates in each of the above plots are computed using the intrinsic LIDAR parameters determined by three different methods.

parameters provided by the manufacturer (intrinsic parameters only – denoted as **Factory**). Note that the **PMSE** only calibrates the offset in the range measurements of each laser scanner [i.e., ρ_{oi} – see (4.2)], while for the rest of the parameters it uses the **Factory** values.

We implemented the **PMSE** as follows: For each calibration-plane configuration, we transformed the laser points reflected from the plane surface to the LIDAR's Euclidean frame [see (4.1), (4.2), (4.13), and (4.14)] based on the **Factory** parameters, and fitted a plane to them using RANSAC [48]. In the next step, we employed least-squares to minimize the distance of the laser points from the fitted planes by optimizing over the range offsets, ρ_{oi}. The Euclidean coordinates of the laser points are then adjusted accordingly, and processed to fit new planes using RANSAC; these re-fitted planes are used, in turn, to re-estimate the range offsets. This process is continued until convergence, or until a maximum number of iterations is reached. Once the range offsets were calibrated, we minimized a least-square cost function similar to (4.28), but only over the extrinsic calibration parameters (i.e., ${}^{C}_{L}\mathbf{C}$ and ${}^{C}\mathbf{t}_{L}$).

4.6.3 Consistency of Intrinsic Parameters

In order to evaluate the consistency of the estimated intrinsic parameters, we collected a new test dataset comprising the image observations and LIDAR snapshots of the calibration plane at 17 different configurations. We then extracted the LIDAR points belonging to the calibration plane following the procedure described in Section 4.6.1, and computed their Euclidean coordinates using the intrinsic parameters of the **Factory**, the **PMSE**, and the **AlgBLS**. A sample reconstruction from each of these methods is shown in Fig. 4.4, where the inaccuracy in some of the intrinsic LIDAR parameters of the **Factory** is clearly visible. To quantitatively compare the estimated intrinsic parameters, we fitted planes to the LIDAR points and computed the (signed) residual distance[7] of each point to the corresponding plane, from which the histograms in Fig. 4.5 are obtained. The detailed statistics of the signed distances for all three methods are presented in Table 4.2. It is evident from these statistics that the **AlgBLS** results in a significantly smaller median value, indicating lower skewness and bias in the errors of the **AlgBLS** than those of the **Factory** and **PMSE**. This, in effect, shows that the **AlgBLS** leads to more consistent calibration across different laser scanners of the LIDAR.

4.6.4 Comparison of Intrinsic & Extrinsic Parameters

The evaluations in Section 4.6.3 are incomplete since they do not reflect the accuracy of the extrinsic camera-LIDAR parameters. To complement the above results, we transformed the laser points in the test datasets to the Ladybug's frame of reference using the extrinsic calibration parameters estimated by the **AlgBLS** and the **PMSE**. Then, we computed the signed residual distance of the points reflected by the calibration planes, from the planes as detected by the Ladybug by evaluating (4.5) for each point. The histograms of these errors (i.e., signed distances) are shown in Fig. 4.6 and their statistics are provided in Table 4.3. As evident from these results, the **AlgBLS** yields superior accuracy compared to the **PMSE**. In particular, higher median and mean error indicates a bias in the estimates of the **PMSE**. This may be explained by the fact that the **PMSE** does not calibrate the scale of the LIDAR measurements.

In order to ensure the repeatability of the proposed calibration procedure, we collected three independent datasets, and employed the **AlgBLS** to determine the calibration parameters from each one. A sample of these estimates is provided in Table 4.4.

[7] We assigned a positive (negative) sign to the distance of the points in front (behind) the plane.

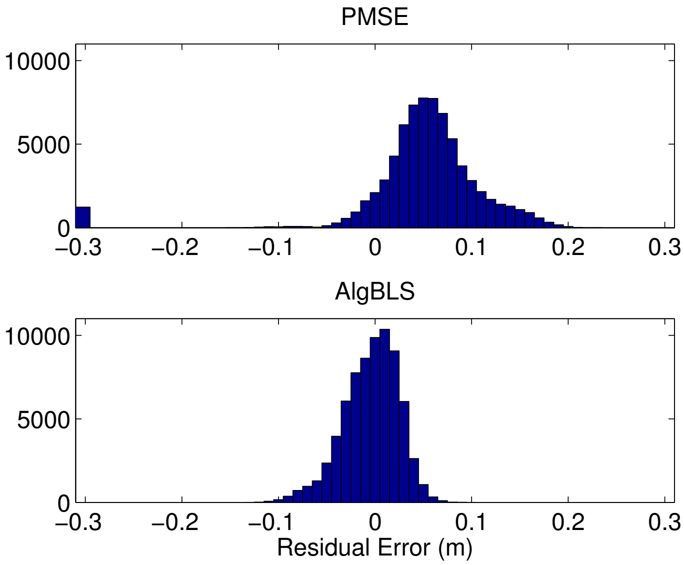


Figure 4.6: Histograms of the signed distance errors of the laser points from the calibration planes detected by the Ladybug.

The parameters' estimates are relatively stable, except for the z component of ${}^{C}\mathbf{t}_{L}$ which shows a few centimeters of variation. This can be explained the fact that due to the limited and mostly horizontal field of view of the LIDAR, the geometric constraints in (4.7) provide less information along the vertical axis of the LIDAR.

Finally, Table 4.4 provides a selection of the calibration parameters obtained by (i) iteratively minimizing the least-squares cost function of (4.28) with an inaccurate initialization (labeled as Bad Init.), (ii) the **PMSE** algorithm, and (iii) the **Factory** intrinsic parameters. Note that with inaccurate initialization, the estimate for the z component of ${}^{C}\mathbf{t}_{L}$ has converged to a local minimum, far from the actual value of the translation between the Velodyne and the Ladybug. The **PMSE** method performs optimization only over the range offset, and uses the **Factory** for the rest of the parameters (labeled with N.A in Table 4.4). In this table, note the inaccurate range offset estimate of the laser scanner 20 as provided by the **Factory**. The point cloud corresponding to a plane obtained using these estimates is shown in Figs. 4.4(a) and 4.4(b). In particular, the laser points from scanner 20 (marked in red) are easily distinguishable as they suffer from ∼30 cm bias. These errors become smaller when using the calibration parameters of **PMSE** method [see Figs. 4.4(c) and 4.4(d)], but they are still noticeable. In contrast, the reconstruction obtained using the calibration parameters estimated by the **AlgBLS** [Figs. 4.4(e) and 4.4(f)] exhibit considerably lower systematic error.

4.6.5 Photorealistic Reconstructions

To further demonstrate the accuracy of the estimated calibration parameters by the **AlgBLS**, we employed the following procedure to create photorealistic reconstructions of several indoor and outdoor scenes from the University of Minnesota campus (see Figs. 4.7-4.8):

Table 4.2: Statistics of the signed distance error of the laser points from the fitted planes.

	Mean (mm)	Median (mm)	Std. Dev. (mm)
Factory	−0.85	0.11	65
PMSE	2.56	−0.61	152
AlgBLS	−0.67	−0.05	55

Table 4.3: Statistics of the signed distance error of the laser points from the calibration planes detected by the Ladybug.

	Mean (mm)	Median (mm)	Std. Dev. (mm)
PMSE	40	55	151
AlgBLS	−4.3	−1.4	28

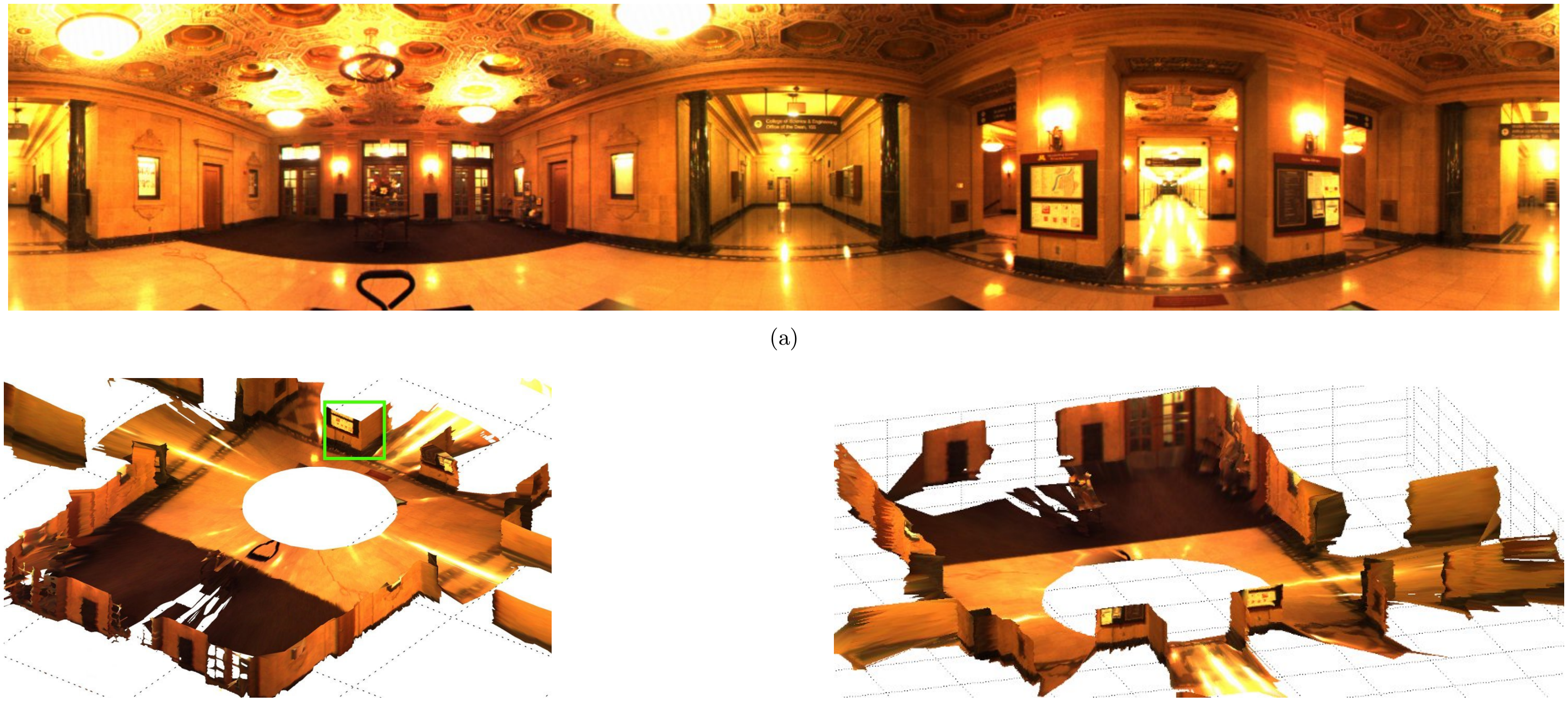
Table 4.4: The first three rows show a selection of calibration parameters as estimated by **AlgBLS** for three different datasets. The column labeled as rpy represents roll, pitch, and yaw of ${}^{C}_{L}\mathbf{C}$. Rows 4-6 show a selection of calibration parameters that are obtained by other methods. Note that the scale parameters α_i are unitless, and N.A. indicates non-applicable fields.

	# of planes	rpy (deg)	${}^{C}\mathbf{t}_L$ (cm)	α_1	ρ_{o1} (cm)	α_{20}	ρ_{o20} (cm)	θ_{o20} (deg)	h_{20} (cm)
AlgBLS 1	18	[1.31 1.27 −88.45]	[−2.5 0.04 −20.10]	1.02	99.49	1.01	100.88	36.91	−3.71
AlgBLS 2	17	[1.62 0.79 −88.40]	[−1.64 0.19 −12.92]	1.01	101.83	1.00	101.82	36.03	−4.98
AlgBLS 3	18	[1.26 0.84 −88.43]	[−2.07 −0.12 −17.95]	1.01	101.05	0.98	105.94	36.05	−0.85
Bad Init.	18	[0.34 −0.63 −87.93]	[−1.64 −0.50 −117.08]	1.01	98.99	0.99	104.49	34.95	98.31
PMSE	18	[3.19 −0.97 −88.32]	[1.40 −2.12 −53.64]	N.A.	91.49	N.A.	90.85	N.A.	N.A.
Factory	18	N.A.	N.A.	1	95.50	1	66.53	34.95	−0.24

- Transform the LIDAR points to the Ladybug's frame of reference using the **AlgBLS** intrinsic and extrinsic calibration parameters.
- Overlay the spherical image of the Ladybug on the LIDAR points, and assign each pixel a depth according to the corresponding LIDAR point. If a pixel has no corresponding LIDAR point, compute an approximate depth by linearly interpolating the nearest laser points.
- Render 3D surfaces from the 3D pixels using Delaunay triangulation [34].

In Figs. 4.7(b)-4.7(c) and 4.8(b)-4.8(c), a selection of the rendered surfaces in Matlab using the estimated calibration parameters by **AlgBLS** are shown for indoor and outdoor scenes. Note that white gaps in the reconstructed surfaces result from relatively large patches of missing LIDAR measurements due to occlusion or specular reflection of the laser beams from glass and shiny surfaces.

The reconstructions using the calibration parameters of **PMSE** and **Factory** look quite similar to Figs. 4.7(b)-4.7(c) and 4.8(b)-4.8(c) when zoomed out, and hence we omit the corresponding figures. However, Figs. 4.9(a)-4.9(d) and 4.10(a)-4.10(d) show close-up views of the same surfaces with renderings obtained by using **AlgBLS** and alternative methods. In particular, in Figs. 4.9(b) and 4.10(b), the calibration parameters estimated by minimizing (4.28) given an imprecise initialization are employed to render the image. In creating Figs. 4.9(c) and 4.10(c), we have only estimated the extrinsic calibration parameters, and used the **Factory** intrinsic parameters. Finally Figs. 4.9(d) and 4.10(d) are rendered using the estimated calibration parameters by **PMSE**. The superior quality of the renderings obtained using the calibration parameters of the **AlgBLS** is clear in these images. In particular, in Figs. 4.10(a)-4.10(d), the lines corresponding to the corner of the wall are estimated based on the LIDAR measurements, and depicted as green dashed line. Given a perfect calibration, the shadow edge of the overlaid image should match the extracted corner. Clearly, while the rendering by **AlgBLS** provides an almost perfect match between the two, the other rendered images suffer from mismatches caused by inaccurate estimates of the calibration parameters. Furthermore, the intrinsic calibration parameters provided by the manufacturer often lead to spikes in the reconstructed surface, as shown by the green arrows in 4.9(c) and 4.10(c).

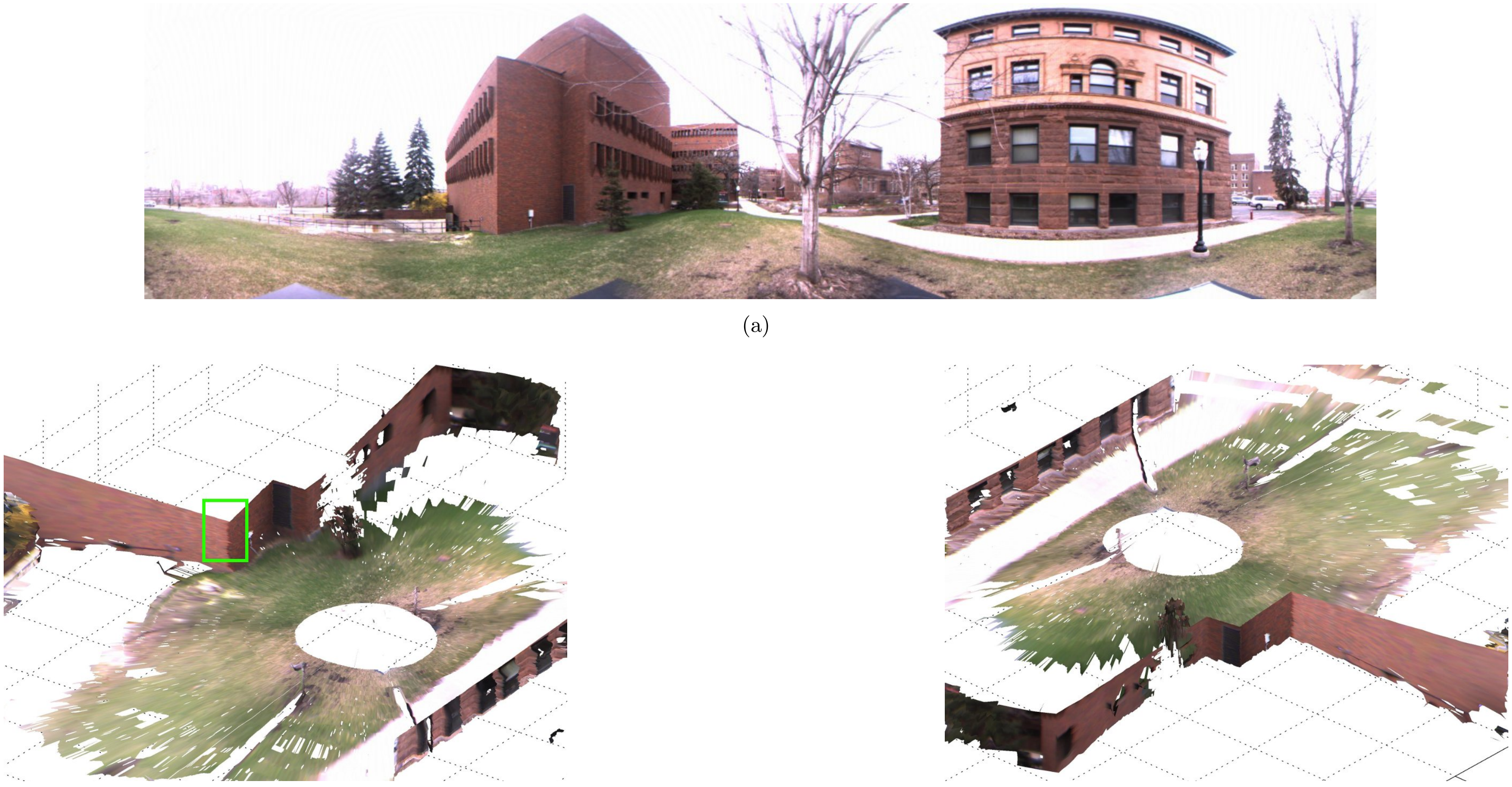


(a)

(b)

(c)

Figure 4.7: Photorealistic reconstruction of an indoor scene (best viewed in color). (a): Panoramic image of an indoor scene with several corridors provided by the Ladybug; (b,c): The corresponding photorealistic reconstruction using the calibration parameters obtained from the **AlgBLS** algorithm, viewed from two different directions. The green rectangle in (b) marks the close-up area shown in Fig. 4.9. The white gaps are the regions where at least one of the sensors did not return meaningful measurements (e.g., due to occlusion, specular reflections, or limited resolution and field of view). Note that the depth of the scene can be inferred from the dotted grids.



(a)

(b)

(c)

Figure 4.8: Photorealistic reconstruction of an outdoor scene (best viewed in color). (a): Panoramic view of an outdoor scene; (b,c): Photorealistic reconstruction of the scene viewed from two different directions. The green rectangle in (b) marks the close-up area shown in Fig. 4.10. The white gaps are the regions where at least one of the sensors did not return meaningful measurements (e.g., due to occlusion, specular reflections, or limited resolution and field of view). Note that some of the occlusions are due to the trees, the lamp post, and different elevations of the grassy area.

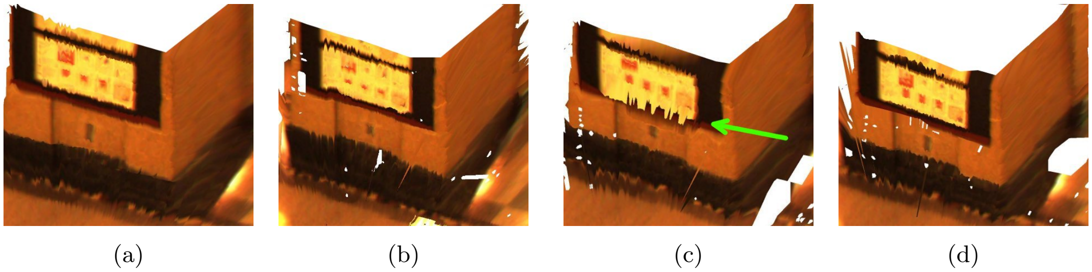


Figure 4.9: The close-up views corresponding to the green rectangle in Fig. 4.7(b) (best viewed in color). (a): The close-up view rendered using the parameters estimated by the **AlgBLS** algorithm; (b): The close-up view rendered using the parameters estimated by an iterative least-squares refinement with inaccurate initialization; (c): The close-up view rendered using the intrinsic parameters provided by the manufacturer; (d): The close-up view rendered using the algorithm proposed by [103].

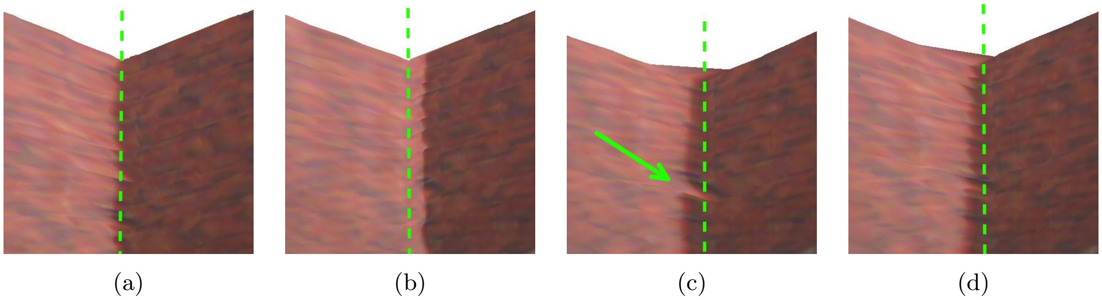


Figure 4.10: The close-up views corresponding to the green rectangle in Fig. 4.8(b) (best viewed in color). (a): The close-up view obtained using the parameters estimated by the **AlgBLS** algorithm; (b): The close-up view obtained using the parameters estimated by an iterative least-squares refinement with inaccurate initialization; (c): The close-up view obtained using the intrinsic parameters provided by the manufacturer. The green arrow points to the spikes created due to inaccurate range offset parameters; (d): The close-up view obtained using the algorithm proposed by [103]. The green dashed lines mark the corner of the wall detected from the LIDAR points.

4.7 Summary

In this chapter, we presented a novel method for intrinsic calibration of a revolving-head 3D LIDAR and extrinsic calibration with respect to a camera. Specifically, we developed an analytical method for computing a precise initial estimate for both the LIDAR's intrinsic parameters and the LIDAR-camera transformation. Subsequently, we used these estimates to initialize an iterative nonlinear least-squares refinement of all the calibration parameters. Additionally, we presented an observability analysis to determine the minimal conditions under which it is possible to estimate the calibration parameters. Experimental results from both indoor and outdoor scenes are used to demonstrate the achieved accuracy of the calibration process by photorealistic reconstruction of the observed areas.

Chapter 5

Extrinsic Camera Calibration from Known Lines[†]

5.1 Introduction

Determining a camera's position and attitude (pose) from known 3D lines and their projections (corresponding 2D lines) in an image has numerous applications in robot localization, computer vision, and augmented reality. While several algorithms exist for *algebraically* determining a camera's pose based on line correspondences [74, 26, 3], they are specifically designed for *noise-free* scenarios where the measurement constraints are exactly satisfied. In the presence of noise, the camera pose computed by these algorithms may become unreliable and inaccurate, since the impact of noise is not explicitly modeled.

On the other hand, many iterative algorithms exist which account for measurement noise by formulating the camera pose estimation as a *nonlinear* least-squares problem [69, 38, 32]. However, these methods do not provide any guarantee of global optimality since the iterative minimization of least-squares cost functions only converges to a stationary point. In fact, in the absence of accurate initialization, these approaches often converge to a point far from the true sensor pose. One workaround is to use the output of an algebraic method for initializing an iterative least-squares algorithm. This, however, inherits the unreliability of existing algebraic methods, and does not ensure convergence to the globally optimal solution.

To address these issues, we introduce a novel approach that *directly* computes the

[†] This work is appeared at the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 2011 [92].

global minimum of the nonlinear least-squares cost for the camera's attitude in one step using tools from algebraic geometry. Moreover, we show that once the sensor's attitude is found, its position can be readily computed from the measurements by means of ordinary least squares. Specifically, we first address the more challenging task of attitude determination by considering the optimality conditions of the least-squares problem for minimizing the measurement residuals due to orientation errors. These optimality constraints form a system of polynomial equations, whose solutions (i.e., all *critical* points of the least-squares cost function) are efficiently computed using eigendecomposition of a so-called multiplication matrix. The globally optimal estimates for the sensor's orientation are then the critical points which minimize the least-squares cost function. In the second stage, we compute the sensor's position using ordinary least squares.

Our proposed approach has several advantages compared to existing algorithms: (i) the optimality of the sensor's orientation estimate is guaranteed in a least-squares sense, (ii) no initialization is required, and (iii) the computational complexity of our approach is *linear* in the number of measurements, whereas state-of-the-art algebraic methods have *quadratic* computational complexity [3]. Furthermore, the developed algorithm can be applied without any modification to solve another robot localization problem, namely estimating pose from line-to-plane correspondences [99, 52]. This is particularly useful when a 2D laser scanner is used to localize a robot inside an *a priori* known building. In this case the straight-line segments in the laser scan that correspond to the structural planes of the building are employed to determine the pose of the laser scanner. The details of this application are provided in [91].

The remainder of this chapter is organized as follows. Section 5.2 provides an overview of the related literature. Section 5.3 presents the least-squares formulation and polynomial optimality conditions for estimating the sensor's orientation, while Section 5.4 describes the employed polynomial solver. In Section 5.5, a least-squares algorithm is presented to estimate the sensor's position given the orientation estimates. The proposed method is validated with extensive simulations and experiments in Section 5.6. Finally, a summary of this chapter is provided in Section 5.7.

5.2 Related Work

Exploiting line correspondences to estimate camera pose has received significant attention in the last two decades. In one of the earliest works, Liu *et al.* [74] propose a method using eight or more measurements to linearly constrain the elements of a rotation matrix expressing the sensor's orientation with respect to the global frame. An up-to-scale estimate of the rotation matrix is then obtained by (linear) least squares, followed by a constraint to ensure the Frobenius norm of the estimated matrix is three. This approach, however, does not necessarily result in a proper orthonormal matrix for noisy measurements.

In [26], Chen proposes an algebraic method to find pose from line correspondences using only three measurements (i.e., the minimum number of measurements required). He also investigates the necessary conditions under which the problem has a finite number of solutions. While the recovered pose is precise in the noise-free case, it is highly unstable in the presence of noise (partially due to utilization of minimal number of measurements), and often produces complex solutions [26]. In addition, this method cannot exploit more than three measurements.

The state-of-the-art algebraic approach to estimate pose from line correspondences, presented by Ansar and Daniilidis in [3], employs *lifting* to convert the polynomials describing four or more measurement constraints to linear equations in the components of the rotation matrix. While this method recovers the orientation precisely in the absence of noise, its performance degrades with increasing measurement-noise variance, and it may even result in complex solutions. Additionally, the lifting method is only guaranteed to work if the polynomial system has exactly one solution. Therefore, in singular configurations where an observed image can correspond to multiple different camera orientations (e.g., when the 3D lines are orthogonal to each other [98]), this method may fail. Moreover, this algorithm has $\mathcal{O}(N^2)$ computational complexity in the number, N, of line measurements used, which can be prohibitive when processing resources are limited.

The main drawback of the aforementioned algebraic methods is that they attempt to solve measurement constraints which are only satisfied in the absence of noise. In the presence of noise and disturbances, however, the coefficients of the polynomials describing the measurement constraints are perturbed. The solutions of a perturbed polynomial system though are extremely unreliable approximations of the roots of the unperturbed system [30, Ch. 2]. In particular, in many instances the solutions of the perturbed system become complex numbers whose real parts are arbitrarily far from the roots of the unperturbed system. To address this issue, one must explicitly account for the measurement noise and formulate the problem as nonlinear least squares with the objective to minimize the measurements' residuals.

In the literature, several iterative (linearization-based) methods have been applied to estimate pose from line correspondences based on nonlinear least-squares minimization. Kumar and Hanson [69] present an iterative least-squares algorithm for recovering the sensor pose. In [38], two iterative methods based on para-perspective and weak-perspective camera models are proposed which show better convergence performance compared to a perspective model in the absence of good initialization. David *et al.* [32] propose an iterative method for finding the camera pose with ambiguous data association. All these methods are *iterative*, and since the nonlinear least-squares cost function is nonconvex, they may converge to a local minimum or a saddle point, and cannot make any claims regarding global optimality.[1] Furthermore, in the absence of a good initialization, these methods are typically slow, and often diverge.

In order to address these limitations, in this chapter we introduce an algebraic method for solving the nonlinear least-squares pose estimation problem from line correspondences. The main advantages of our approach are: (i) it does not require initialization; (ii) it guarantees the global optimality of the estimated sensor orientation in a least-squares sense; and (iii) it computes all possible poses, if more than one solution exists. Specifically, we formulate the nonlinear least-squares problem for minimizing the measurement residuals due to orientation errors and find *all* its critical points by directly solving the system of polynomial equations describing the optimality (minimization) conditions. This multivariate polynomial system is solved in linear (in the number of measurements) time by efficient construction and eigendecomposition of the so-called multiplication matrix [30]. Subsequently, the objective function is evaluated at all critical points and the one(s) that results in the smallest cost is selected as the global minimizer. Finally, once the sensor's orientation is determined, we compute its position using linear least squares.

5.3 Problem Formulation

As mentioned before the most challenging part of pose determination from line correspondences is estimating the camera's attitude since it requires solving a set of nonlinear (polynomial) equations. In the following two sections, we present our method for estimating the camera's attitude, while position determination, given the estimated

[1] Notice that convergence to a local minimum (or generally a stationary point) is due to the nonconvexity of the cost function (e.g., see [11]), and is a completely separate issue from the numerical stability or the convergence properties of these algorithms. Thus this issue *cannot* be addressed by using numerically robust (e.g., conjugate-gradient type) iterative least-squares methods.

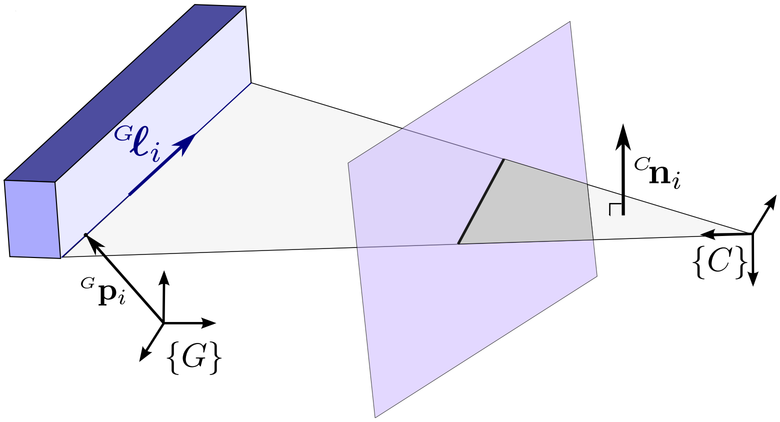


Figure 5.1: The i-th 3D line is described in the global frame $\{G\}$ by its direction ${}^{G}\boldsymbol{\ell}_i$, and its moment ${}^{G}\mathbf{m}_i = {}^{G}\mathbf{p}_i \times {}^{G}\boldsymbol{\ell}_i$, where ${}^{G}\mathbf{p}_i$ is any arbitrary point on the line. Camera observations of the i-th 3D line can be represented as the projection plane (colored in gray) passing through the 3D line and the optical center of the camera. This plane is described by the normal vector ${}^{C}\mathbf{n}_i$ expressed in the camera frame. The observed 2D line is the intersection of this plane and the image plane (colored in violet).

attitude, is described in Section 5.5.

We assume that N 3D lines with known coordinates (e.g., *a priori* mapped edges of doors and intersections of walls, ceiling, and floor) and their corresponding 2D projections on the image plane of a pinhole camera are given. Let us define the i-th *projection plane*, $i = 1, \ldots, N$, as the plane that passes through the origin of the camera and the image of the i-th line in that camera (see Fig. 5.1). The normal vector of this plane can be obtained from the line measured by the camera. Specifically, if we represent the image of the i-th line with polar parameters (θ_i, ρ_i), then every point $[u\ v\ 1]^T$ belonging to that line satisfies the equation $u \cos\theta_i + v \sin\theta_i + \rho_i = 0$, and ${}^{C}\mathbf{n}_i = [\cos\theta_i\ \sin\theta_i\ \rho_i]^T$ is the normal vector of the corresponding projection plane. From Fig. 5.1, it is clear that in the absence of noise, a 3D line lays on its corresponding projection plane. Therefore, if we denote the *a priori* known direction of the i-th 3D line with ${}^{G}\boldsymbol{\ell}_i$, the unknown orientation of the camera in the global frame, represented by the rotation matrix ${}^{C}_{G}\mathbf{C}^T$, satisfies the following constraint:

$$ {}^{C}\mathbf{n}_i^T\, {}^{C}_{G}\mathbf{C}\, {}^{G}\boldsymbol{\ell}_i = 0. \tag{5.1}$$

In the presence of noise, we do not have perfect measurements of ${}^{C}\mathbf{n}_i$. Instead, we measure ${}^{C}\hat{\mathbf{n}}_i = {}^{C}\mathbf{n}_i + {}^{C}\tilde{\mathbf{n}}_i$, where ${}^{C}\tilde{\mathbf{n}}_i$ is the 3×1 vector of zero-mean Gaussian noise with known covariance $\mathbf{R}_{\mathbf{n},i}$. Substituting ${}^{C}\mathbf{n}_i$ in (5.1) yields:

$$ {}^{C}\hat{\mathbf{n}}_i^T \, {}^{C}_{G}\mathbf{C} \, {}^{G}\boldsymbol{\ell}_i = {}^{C}\tilde{\mathbf{n}}_i^T \, {}^{C}_{G}\mathbf{C} \, {}^{G}\boldsymbol{\ell}_i \triangleq \eta_i. \tag{5.2} $$

The measurement *residual*, η_i, is a zero-mean Gaussian random variable with variance $\sigma_i^2 \triangleq {}^{G}\boldsymbol{\ell}_i^T \, {}^{C}_{G}\mathbf{C}^T \, \mathbf{R}_{\mathbf{n},i} \, {}^{C}_{G}\mathbf{C} \, {}^{G}\boldsymbol{\ell}_i$.

Given several noisy line correspondences, the objective is to estimate ${}^{C}_{G}\mathbf{C}$. This can be achieved by minimizing the following cost function:

$$ {}^{C}_{G}\widehat{\mathbf{C}} = \arg\min_{\mathbf{C}} \frac{1}{2} \sum_{i=1}^{N} \sigma_i^{-2} \left({}^{C}\hat{\mathbf{n}}_i^T \, \mathbf{C} \, {}^{G}\boldsymbol{\ell}_i \right)^2 \tag{5.3} $$

$$ \text{subject to } \mathbf{C}^T\mathbf{C} = \mathbf{I}_3, \;\; \det(\mathbf{C}) = 1. \tag{5.4} $$

This nonlinear weighted least-squares problem for $N \geq 3$ can be solved using iterative methods such as Gauss-Newton [63]. However, iterative approaches often converge to local minima, and require an accurate initial estimate. To address these limitations, we hereafter present a new algebraic method that *directly* solves the nonlinear least-squares problem without requiring initialization.

We start by expressing the orientation of the sensor using the Cayley–Gibbs–Rodriguez (CGR) parametrization since (i) the components of the rotation matrix are naturally expressed as rational functions of the CGR parameters, and (ii) CGR is a minimal representation of rotation, and thus, does not require additional constraints such as the ones in (5.4) to ensure that it corresponds to a valid rotation [117]. Furthermore, the CGR parametrization introduces the minimum number of unknowns in the resulting polynomial system and hence allows fast computation of its solutions.

In CGR representation, a rotation matrix is expressed as

$$ \mathbf{C} = (\mathbf{I}_3 - \lfloor \mathbf{s} \times \rfloor)^{-1} (\mathbf{I}_3 + \lfloor \mathbf{s} \times \rfloor) \tag{5.5} $$

where $\mathbf{s}^T = [s_1 \;\, s_2 \;\, s_3]$ is the vector of CGR parameters, and $\lfloor \mathbf{s} \times \rfloor$ is the corresponding skew-symmetric matrix. Equation (5.5) can be expanded as

$$ \mathbf{C} = \frac{\bar{\mathbf{C}}}{1 + \mathbf{s}^T\mathbf{s}}, \quad \bar{\mathbf{C}} \triangleq \left((1 - \mathbf{s}^T\mathbf{s})\mathbf{I}_3 + 2\lfloor \mathbf{s} \times \rfloor + 2\mathbf{s}\mathbf{s}^T \right). \tag{5.6} $$

Substituting this expression in the constraint equation (5.1), and multiplying both sides by $(1+\mathbf{s}^T\mathbf{s})$ yields:

$$^{C}\mathbf{n}_i^T \, {}^{C}_{G}\bar{\mathbf{C}} \, {}^{G}\boldsymbol{\ell}_i = 0. \tag{5.7}$$

This constraint is linear in the components of ${}^{C}_{G}\bar{\mathbf{C}}$ which are, in turn, quadratic monomials in the elements of $\mathbf{s}$ [see (5.6)]. In the presence of noise, substituting ${}^{C}\mathbf{n}_i = {}^{C}\hat{\mathbf{n}}_i - {}^{C}\tilde{\mathbf{n}}_i$ in (5.7) yields

$$^{C}\hat{\mathbf{n}}_i^T \, {}^{C}_{G}\bar{\mathbf{C}} \, {}^{G}\boldsymbol{\ell}_i = {}^{C}\tilde{\mathbf{n}}_i^T \, {}^{C}_{G}\bar{\mathbf{C}} \, {}^{G}\boldsymbol{\ell}_i \triangleq \bar{\eta}_i \tag{5.8}$$

where the measurement residual, $\bar{\eta}_i$, is a zero-mean Gaussian random variable with variance $\bar{\sigma}_i^2 \triangleq {}^{G}\boldsymbol{\ell}_i^T \, {}^{C}_{G}\bar{\mathbf{C}}^T \, \mathbf{R}_{\mathbf{n},i} \, {}^{C}_{G}\bar{\mathbf{C}} \, {}^{G}\boldsymbol{\ell}_i$. Based on (5.8), we form the following weighted least-squares problem for estimating $\hat{\mathbf{s}}$ from multiple noisy line observations:

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} J, \qquad J \triangleq \frac{1}{2}\sum_{i=1}^{N} \bar{\sigma}_i^{-2} \left({}^{C}\hat{\mathbf{n}}_i^T \bar{\mathbf{C}} \, {}^{G}\boldsymbol{\ell}_i \right)^2 . \tag{5.9}$$

Compared to (5.3)-(5.4) the optimization constraint is now removed since the Cayley transformation [see (5.5)] ensures the orthonormality of the estimated rotation matrix. Minimizing J, however, turns out to be computationally intractable, as the degree of the polynomials describing its optimality conditions quickly increases with the addition of each new measurement. For example, given three measurements, the optimality conditions for minimization of (5.9) will be polynomials of degree 63, while for 10 measurements, they will be of degree $2^{20}-1$.

To mitigate this challenge, we relax the problem by assuming that the variance of the measurement residuals, $\bar{\sigma}_i^2$, is approximately the same for all measurements. This relaxation yields the following least-squares problem:

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} J', \qquad J' \triangleq \frac{1}{2}\sum_{i=1}^{N} \left({}^{C}\hat{\mathbf{n}}_i^T \bar{\mathbf{C}} \, {}^{G}\boldsymbol{\ell}_i \right)^2 . \tag{5.10}$$

To algebraically find the global minimum of (5.10), we first determine all the critical points of J' by solving the following *optimality conditions* [11]:

$$f_j(\mathbf{s}) = \frac{\partial J'}{\partial s_j} = \sum_{i=1}^{N} \left({}^{C}\hat{\mathbf{n}}_i^T \bar{\mathbf{C}} \, {}^{G}\boldsymbol{\ell}_i \right) \frac{\partial}{\partial s_j} \left({}^{C}\hat{\mathbf{n}}_i^T \bar{\mathbf{C}} \, {}^{G}\boldsymbol{\ell}_i \right) = 0 \tag{5.11}$$

for $j = 1, 2, 3$ and $N \geq 3$. While it is possible to employ the polynomial solver that was described in Section 4.4, we choose to follow a simpler approach that does not require us to compute the Gröbner basis even for an integer instance of the problem. For this purpose, we employ the following proposition:

Proposition 5.1 (Bézout Theorem [30])**.** *When a system of equations composed of n polynomials of degrees $d_1, d_2, \cdots, d_n$, (i) has finite number of solution and no solution at infinity, and (ii) all of its solutions are of multiplicity one, it generically*[2] *has $d_1 d_2 \cdots d_n$ distinct solutions. In this case, all the solutions can be obtained using the procedure described in Section 5.4.*[3]

As shown in [26], *at least three* measurements from lines with linearly independent directions are required in order to recover the camera's attitude. Note, however, that the three optimality conditions [see (5.11)] are always *cubic* polynomials regardless of the number of measurements; thus, according to Proposition 5.1, the polynomial system describing the optimality conditions has 27 solutions, each of which is a critical point of J'. The globally optimal solutions of (5.10) are the critical points that minimize J'.[4]

Note that the computational complexity of solving (5.11) and finding the global minimum does not increase with the addition of measurements, since the degree and number of polynomials expressing the optimality conditions are fixed. Moreover, computing the contribution of all measurements to the coefficients of the cubic polynomials $f_j,\ j = 1, 2, 3$ increases linearly with the number of measurements.

5.4 Solving Polynomial Systems using Macaulay Matrix

Once the optimality conditions (5.11) are expressed as a system of multivariate polynomial equations, there exist several methods for solving them. Amongst them, numerical methods, such as Newton-Raphson, need initialization and may not find all the solutions. Symbolic reduction methods based on the computation of the system's Gröbner basis are capable of finding all roots without any initialization [30]. However, they can only be used for integer coefficients since their application to floating-point numbers

[2] The word 'generically' implies that exceptions can occur for singular values of the coefficients, raised by singular configurations of the camera or the observed lines. For a more precise definition of *genericity*, please refer to [30].

[3] These conditions, which are stronger than those required for the methodology described in Section 4.4, are generally satisfied when all the monomials up to the degree d_j are present in the j-th polynomial (see Section 5.4.2). While this is the case for the pose-from-line-correspondences problem, it is not so, for the polynomial system corresponding to the 3D LIDAR-camera calibration system (see Section 4.3). Intuitively, this can be verified by noting that while the system in (4.23) has four polynomials of degree five, it does not have $5^4 = 625$ solutions, and instead, it only has 243 solutions.

[4] In general, when $N > 3$ lines are observed, there exists a unique global minimum.

suffers from quick accumulation of round-off errors, which in turn, results in incorrect solutions [30]. Instead, we employ a method developed by Auzinger and Stetter [8] that computes a generalization of the companion matrix to systems of multivariate polynomial equations, namely the *multiplication matrix*, whose eigenvalues are the roots of the associated polynomial system.

Computing the multiplication matrix as described in Section 4.4 requires us to first determine the normal set of the polynomial system. Instead, considering that the polynomial system in (5.11) satisfies the conditions of Proposition 5.1, we employ an alternative technique to constructs the multiplication matrix, which does not require explicit calculation of the normal set. This is achieved by means of an intermediate so-called *Macaulay* matrix that was originally developed to determine the resultant of a system of polynomial equations [30]. In the following, we first describe a method to construct the Macaulay matrix, and then compute the multiplication matrix using Schur decomposition of the Macaulay matrix. It is important to note (see Section 5.4.3) that the Macaulay matrix needs to be constructed only once in symbolic form (i.e., treating the coefficients of the polynomials as unknown parameters) and then, in each realization of the problem, we substitute the coefficients obtained from the measurements.

5.4.1 Constructing the Macaulay Matrix

We start by introducing the necessary notation and provide a brief overview of algebraic geometry concepts that will be used to compute the solutions of (5.11). For a detailed discussion of this topic, we refer the reader to [30].

We denote a *monomial* in n variables by $\mathbf{x}^{\gamma} \triangleq x_1^{\gamma_1} x_2^{\gamma_2} \cdots x_n^{\gamma_n}$, $\gamma_i \in \mathbb{Z}_{\geq 0}$, and a polynomial in n variables with complex coefficients by $f = \sum_j c_j \mathbf{x}^{\gamma_j}, c_j \in \mathbb{C}$. The degree of each monomial is defined as $\sum_{i=1}^{n} \gamma_i$, and the degree of a polynomial is the maximum degree of all its monomials. We assume that the given system of equations has n polynomials, denoted by $f_i = 0,\ i = 1, \ldots, n$, each of them with degree d_i. We define an auxiliary *linear* polynomial, so-called u-polynomial, as $f_0 = u_0 + u_1 x_1 + \cdots + u_n x_n$, where u_i are independently drawn random numbers. Notice that, in general, f_0 will not be zero at the roots of the given system of polynomial equations.

We proceed with defining the total degree of the system of equations, including the auxiliary polynomial as $d \triangleq \sum_{i=0}^{n} d_i - n = 1 + \sum_{i=1}^{n} d_i - n$. Then we define the set of all possible monomials of degree less than or equal to d as $\mathcal{S} = \{\mathbf{x}^{\gamma} \ : \ \sum_j \gamma_j \leq d\}$. It can be easily shown that $\mathcal{S}$ has $\binom{n+d}{n}$ members [30]. For illustration purposes, consider the following system of $n = 2$ polynomials:

$$f_1 = x_1 + 2x_2 + 5 \quad , \quad f_2 = x_1^2 + x_2^2 - 100. \tag{5.12}$$

In this example, $d_1 = 1$ and $d_2 = 2$, and the total degree after adding the auxiliary polynomial $f_0 = u_0 + u_1 x_1 + u_2 x_2$ is $d = 2$. The set of monomials with degree less than or equal to $d = 2$ is $\mathcal{S} = \{1, x_1, x_2, x_1x_2, x_1^2, x_2^2\}$.

In the next step, we partition $\mathcal{S}$ into $n+1$ disjoint subsets:

$$\begin{aligned}
\mathcal{S}_n &= \{\mathbf{x}^{\boldsymbol{\gamma}} \ : \mathbf{x}^{\boldsymbol{\gamma}} \in \mathcal{S};\ x_n^{d_n} \text{ divides } \mathbf{x}^{\boldsymbol{\gamma}}\} \\
\mathcal{S}_{n-1} &= \{\mathbf{x}^{\boldsymbol{\gamma}} \ : \mathbf{x}^{\boldsymbol{\gamma}} \in \mathcal{S}, \notin \mathcal{S}_n;\ x_{n-1}^{d_{n-1}} \text{ divides } \mathbf{x}^{\boldsymbol{\gamma}}\} \\
&\vdots \\
\mathcal{S}_0 &= \{\mathbf{x}^{\boldsymbol{\gamma}} \ : \mathbf{x}^{\boldsymbol{\gamma}} \in \mathcal{S}, \notin \mathcal{S}_n, \ldots, \notin \mathcal{S}_1\}
\end{aligned}$$

which, for the example system in (5.12), yields $\mathcal{S}_2 = \{x_2^2\}$, $\mathcal{S}_1 = \{x_1^2, x_1x_2, x_1\}$, $\mathcal{S}_0 = \{1, x_2\}$.

Note that in this partitioning $|\mathcal{S}_0| = d_1 d_2 \cdots d_n$, where $|\cdot|$ denotes the cardinality of a set. This is easy to see if we consider that since $x_i^{d_i}$, $i = 1, \ldots, n$ do not divide $\mathbf{x}^{\boldsymbol{\gamma}} \in \mathcal{S}_0$, the power γ_j of each factor x_j in $\mathbf{x}^{\boldsymbol{\gamma}} = x_1^{\gamma_1} x_2^{\gamma_2} \cdots x_n^{\gamma_n} \in \mathcal{S}_0$ can be any integer such that $0 \leq \gamma_j < d_j$. Clearly, under this condition there exist $d_1 d_2 \cdots d_n$ possible choices for γ_j, $j = 1, \ldots, n$, and accordingly, the same number of distinct monomials belonging to $\mathcal{S}_0$. Additionally, observe that when $|\mathcal{S}_0| > n$, the set $\mathcal{S}_0$ contains 1 and all the monomials $x_1, x_2, \ldots, x_n$. Later on, we will use this important fact to retrieve the solutions of the polynomial system from the eigenvectors of the multiplication matrix.

Based on $\mathcal{S}_0, \ldots, \mathcal{S}_n$, we define the following sets of monomials

$$\mathcal{S}'_i = \left\{ \frac{\mathbf{x}^{\boldsymbol{\gamma}}}{x_i^{d_i}} \ : \mathbf{x}^{\boldsymbol{\gamma}} \in \mathcal{S}_i \right\}, \quad i = 1, \ldots, n, \quad \mathcal{S}'_0 = S_0$$

and generate an extended set of polynomials by multiplying each polynomial f_i by all monomials in the corresponding $\mathcal{S}'_i$:

$$\begin{aligned}
g_{0,j} &\triangleq \mathbf{x}^{\boldsymbol{\gamma}_j} f_0,\ j = 1, \ldots, |\mathcal{S}'_0|, \quad \text{for each } \mathbf{x}^{\boldsymbol{\gamma}_j} \in \mathcal{S}'_0 \\
&\vdots \\
g_{n,j} &\triangleq \mathbf{x}^{\boldsymbol{\gamma}_j} f_n,\ j = 1, \ldots, |\mathcal{S}'_n|, \quad \text{for each } \mathbf{x}^{\boldsymbol{\gamma}_j} \in \mathcal{S}'_n.
\end{aligned}$$

Note that by construction, we have $|\mathcal{S}'_0 \cup \cdots \cup \mathcal{S}'_n| = |\mathcal{S}| = \binom{n+d}{n}$ extended polynomials. For the example system (5.12), we have $\mathcal{S}'_2 = \{1\},\ \mathcal{S}'_1 = \{x_1, x_2, 1\},\ \mathcal{S}'_0 = \{1, x_2\}$, and

$$\begin{aligned} g_{0,1} &= u_0 + u_1 x_1 + u_2 x_2 \\ g_{0,2} &= u_0 x_2 + u_1 x_1 x_2 + u_2 x_2^2 \\ g_{1,1} &= x_1^2 + 2x_2 x_1 + 5x_1 \\ g_{1,2} &= x_1 x_2 + 2x_2^2 + 5x_2 \\ g_{1,3} &= x_1 + 2x_2 + 5 \\ g_{2,1} &= x_1^2 + x_2^2 - 100. \end{aligned}$$

Since the members of $\mathcal{S}_i$ have degrees less than or equal to d, the members of $\mathcal{S}'_i$ will have degrees less than or equal to $d - d_i$. Therefore, all the monomials of $g_{i,j}$ are of degree less than or equal to d. This enables us to express them as a linear combination of the elements of $\mathcal{S}$ (recall that by construction, $\mathcal{S}$ contains all the monomials with degree up to d). We write this linear combination as the inner product of a vector of coefficients $\mathbf{c} \triangleq [c_1\ c_2\ \cdots\ c_\ell]^T$, and $\underline{\mathbf{x}}^{\boldsymbol{\gamma}} \triangleq [\mathbf{x}^{\gamma_1}\ \mathbf{x}^{\gamma_2}\ \cdots\ \mathbf{x}^{\gamma_\ell}]^T$ with $\mathbf{x}^{\gamma_i} \in \mathcal{S}$ and $\ell \triangleq |\mathcal{S}| = \binom{n+d}{n}$, i.e.,

$$g_{i,j} = \mathbf{x}^{\gamma_j} f_i = \mathbf{c}_{i,j}^T \underline{\mathbf{x}}^{\boldsymbol{\gamma}},\ i = 0, \ldots, n,\ j = 1, \ldots, |\mathcal{S}'_i|.$$

Stacking together all available $g_{i,j}$ polynomials and arranging $\underline{\mathbf{x}}^{\boldsymbol{\gamma}} = [\underline{\mathbf{x}}^{\boldsymbol{\alpha}}\ \underline{\mathbf{x}}^{\boldsymbol{\beta}}]^T$, where $\underline{\mathbf{x}}^{\boldsymbol{\alpha}}$ are monomials of $\mathcal{S}_0$ and $\underline{\mathbf{x}}^{\boldsymbol{\beta}}$ are the rest of the monomials, yields:

$$\begin{bmatrix} g_{0,1}(\underline{\mathbf{x}}^{\boldsymbol{\gamma}}) \\ g_{0,2}(\underline{\mathbf{x}}^{\boldsymbol{\gamma}}) \\ \vdots \\ g_{1,1}(\underline{\mathbf{x}}^{\boldsymbol{\gamma}}) \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{c}_{0,1}^T \\ \mathbf{c}_{0,2}^T \\ \vdots \\ \mathbf{c}_{1,1}^T \\ \vdots \end{bmatrix} \underline{\mathbf{x}}^{\boldsymbol{\gamma}} = M \underline{\mathbf{x}}^{\boldsymbol{\gamma}} = M \begin{bmatrix} \underline{\mathbf{x}}^{\boldsymbol{\alpha}} \\ \underline{\mathbf{x}}^{\boldsymbol{\beta}} \end{bmatrix}. \tag{5.13}$$

The *Macaulay matrix*, M, is a square matrix of dimension $|\mathcal{S}| = \binom{n+d}{n}$ comprising the coefficients of $f_0, \ldots, f_n$. This matrix produces the extended set of polynomials $g_{i,j}$ from the vector of monomials $\underline{\mathbf{x}}^{\boldsymbol{\gamma}}$, and plays an important role in computing the resultant of a system of polynomial equations [30]. In the next section, we describe the process for extracting the multiplication matrix from the Macaulay matrix and for finding the roots of the polynomial system (5.11).

5.4.2 Computing the Roots of the Polynomial System

Let $\mathbf{p} = [p_1 \ \cdots \ p_n]^T$ be a solution of the system of polynomial equations, i.e., $f_1(\mathbf{p}) = \cdots = f_n(\mathbf{p}) = 0$ [see (5.11)], and thus $g_{1,1}(\mathbf{p}) = \cdots = g_{n,|\mathcal{S}_n'|}(\mathbf{p}) = 0$ (note that $f_0(\mathbf{p})$ and $g_{0,j}(\mathbf{p})$ are not generally zero). Denoting the vector of monomials $\underline{\mathbf{x}}^{\gamma}$ evaluated at $\mathbf{p}$ as $\underline{\mathbf{p}}^{\gamma}$, and substituting in (5.13), yields:

$$\begin{bmatrix} g_{0,1}(\mathbf{p}) \\ \vdots \\ g_{0,|\mathcal{S}_0|}(\mathbf{p}) \\ 0 \\ \vdots \\ 0 \end{bmatrix} = M \begin{bmatrix} \underline{\mathbf{p}}^{\boldsymbol{\alpha}} \\ \underline{\mathbf{p}}^{\boldsymbol{\beta}} \end{bmatrix} \Leftrightarrow \begin{bmatrix} f_0(\mathbf{p})\, \underline{\mathbf{p}}^{\boldsymbol{\alpha}} \\ \mathbf{0} \end{bmatrix} = M \begin{bmatrix} \underline{\mathbf{p}}^{\boldsymbol{\alpha}} \\ \underline{\mathbf{p}}^{\boldsymbol{\beta}} \end{bmatrix} \tag{5.14}$$

where by construction, $[g_{0,1} \ \cdots \ g_{0,|\mathcal{S}_0|}]^T = f_0\, \underline{\mathbf{x}}^{\boldsymbol{\alpha}}$. We introduce the partitioning $M = \left[\begin{smallmatrix} M_{00} & M_{01} \\ M_{10} & M_{11} \end{smallmatrix}\right]$ where M_{00} is of dimensions $|\mathcal{S}_0| \times |\mathcal{S}_0|$, and the other submatrices are of compatible size, and write (5.14) as

$$\begin{bmatrix} f_0(\mathbf{p})\, \underline{\mathbf{p}}^{\boldsymbol{\alpha}} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} M_{00} & M_{01} \\ M_{10} & M_{11} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{p}}^{\boldsymbol{\alpha}} \\ \underline{\mathbf{p}}^{\boldsymbol{\beta}} \end{bmatrix} \tag{5.15}$$

Employing the Schur complement of M, we obtain

$$f_0(\mathbf{p})\, \underline{\mathbf{p}}^{\boldsymbol{\alpha}} = \widetilde{M}\, \underline{\mathbf{p}}^{\boldsymbol{\alpha}} \tag{5.16}$$

where $\widetilde{M} = M_{00} - M_{01}M_{11}^{-1}M_{10}$ has dimensions $|\mathcal{S}_0| \times |\mathcal{S}_0|$ and is the *multiplication matrix*. Note that M_{11} is generically invertible when the conditions of Proposition 5.1 are satisfied. Intuitively, this is the case when the coefficients of the most of the monomials appearing in the original polynomials are not deterministically zero (i.e., most of the monomials are always present in the polynomial). This is a relatively strong condition, and often difficult to satisfy. For the optimality conditions in (5.11), however, all the monomials up to the maximum degree (i.e., cubic) are generically present, and therefore M_{11} is generically invertible.

As evident from (5.16), the vector of monomials of $\mathcal{S}_0$ (i.e., $\underline{\mathbf{x}}^{\boldsymbol{\alpha}}$), evaluated at a root $\mathbf{p}$ of the original polynomial system, is an eigenvector of $\widetilde{M}$, while $f_0(\mathbf{p})$ is the corresponding eigenvalue. Therefore, to find the 27 solutions of our polynomial system [see (5.11)], we first compute the eigenvectors of $\widetilde{M}$ which has dimension 27. Then, considering that one of the monomials in $\underline{\mathbf{x}}^{\boldsymbol{\alpha}}$ should be equal to 1, we scale the eigenvectors such that their components corresponding to this monomial become one. Finally, the roots of (5.11) appear in the elements of the scaled eigenvectors that correspond to the monomials $x_1, x_2, \ldots, x_n$, $n = 3$ in the vector $\underline{\mathbf{x}}^{\boldsymbol{\alpha}}$ (note that for the case of (5.11) $|\mathcal{S}_0| = 27 > n = 3$).

5.4.3 Implementation Remarks

The method described for determining the roots of (5.11) can be implemented very efficiently. Note that the construction of the Macaulay matrix is independent of the explicit values of each polynomial's coefficients since the degrees of the polynomials remain the same. Hence, although the coefficients change in each realization of the problem (as they depend on the measurements), we can treat each coefficient as a symbolic parameter, and construct the Macaulay matrix (of dimension $|\mathcal{S}| = 120$) as a function of these parameters off-line (e.g., using publicly available Maple packages [85, 18]). For each realization of the problem, we (i) replace the symbolic parameters of the Macaulay matrix with floating-point coefficients obtained from the measurements, (ii) compute its Schur complement to obtain the multiplication matrix $\widetilde{M}$ (of dimension $|\mathcal{S}_0| = 27$), (iii) determine the eigenvectors of $\widetilde{M}$, and (iv) read the roots of (5.11) from the eigenvectors' corresponding elements after scaling. Note that the 27 roots of (5.11) are the critical points of (5.10). To find the global minimum, we substitute them in (5.10) and select the one(s) that minimizes the cost function J'.

In practice, M_{11} in (5.15) may be bad conditioned or even rank deficient, preventing accurate computation of the Schur complement of M. This can happen in several situations: (i) when the u-polynomial f_0 is close to zero at the solution of the polynomial system; (ii) when the rotation angle corresponding to ${}^{C}_{G}\bar{\mathbf{C}}$ is close to 180° (leading to extremely large CGR parameters); (iii) if the 3D line directions ${}^{G}\boldsymbol{\ell}_i$ or the line measurements ${}^{C}\mathbf{n}_i$ have one or two zero components (e.g., when 3D lines are aligned with the cardinal axes). The first problem is easily addressed by re-generating another random u-polynomial. The last two problems are resolved by rotating the measurements or 3D lines to an arbitrary (randomly generated) frame of reference, finding the global minimum(s), and then rotating the solutions back to the original frame.

5.5 Estimation of Sensor Position

Once the orientation of the camera is known, we can easily compute its position, ${}^{G}\mathbf{p}_C$ using the *a priori* known *moments* of the 3D lines, expressed in the global frame as ${}^{G}\mathbf{m}_i \triangleq {}^{G}\mathbf{p}_i \times {}^{G}\boldsymbol{\ell}_i, i \geq 3$, where ${}^{G}\mathbf{p}_i$ is any point on the 3D line (see Fig. 5.1). Following the same convention, the moment of the i-th line expressed in the camera frame is ${}^{C}\mathbf{m}_i = {}^{C}\mathbf{p}_i \times {}^{C}\boldsymbol{\ell}_i$, where ${}^{C}\boldsymbol{\ell}_i = {}^{C}_{G}\mathbf{C}\,{}^{G}\boldsymbol{\ell}_i$, and ${}^{C}\mathbf{p}_i = {}^{C}_{G}\mathbf{C}\,({}^{G}\mathbf{p}_i - {}^{G}\mathbf{p}_C)$. Expanding ${}^{C}\mathbf{m}_i$ yields:

$$\begin{aligned}{}^{C}\mathbf{m}_i &= {}^{C}\mathbf{p}_i \times {}^{C}\boldsymbol{\ell}_i \\ &= {}^{C}_{G}\mathbf{C}\,({}^{G}\mathbf{p}_i - {}^{G}\mathbf{p}_C) \times {}^{C}_{G}\mathbf{C}\,{}^{G}\boldsymbol{\ell}_i \\ &= {}^{C}_{G}\mathbf{C}\,{}^{G}\mathbf{m}_i + {}^{C}_{G}\mathbf{C}\,\lfloor {}^{G}\boldsymbol{\ell}_i \times \rfloor\,{}^{G}\mathbf{p}_C. \end{aligned} \tag{5.17}$$

Although ${}^{C}\mathbf{m}_i$ cannot be measured directly, one can easily check that it is perpendicular to the projection plane of the i-th line. Therefore, in the absence of noise, any point ${}^{C}\mathbf{w}_i = [u_i \; v_i \; 1]^T$ that lays on the image of the i-th line, and thus the projection plane, satisfies the constraint:

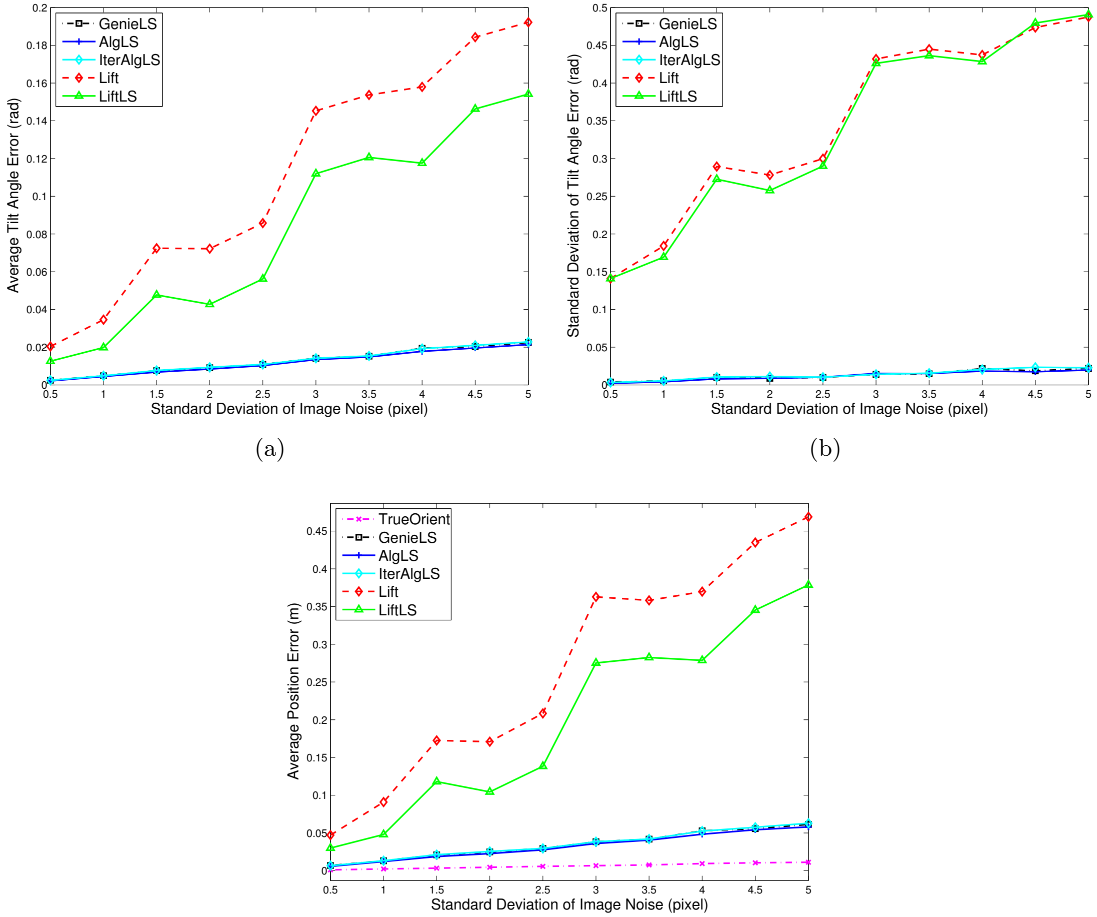
$$ {}^{C}\mathbf{w}_i^{T\,C}\mathbf{m}_i = 0. \tag{5.18}$$

In particular, if the image of the i-th line is parametrized as $u \cos\theta_i + v \sin\theta_i + \rho_i = 0$ (with θ_i and ρ_i computed using least-squares line fitting [137]), then we choose ${}^{C}\mathbf{w}_i = [-\rho_i \cos\theta_i \; -\rho_i \sin\theta_i \; 1]^T$. Substituting (5.17) in (5.18) we obtain:

$$ {}^{C}\mathbf{w}_i^T({}^{C}_{G}\widehat{\mathbf{C}}\,{}^{G}\mathbf{m}_i + {}^{C}_{G}\widehat{\mathbf{C}}\,\lfloor {}^{G}\boldsymbol{\ell}_i \times \rfloor\,{}^{G}\mathbf{p}_C) = 0 \tag{5.19}$$

where we have replaced ${}^{C}_{G}\mathbf{C}$ with its estimate ${}^{C}_{G}\widehat{\mathbf{C}}$ (see (5.5) and Section 5.4). Given measurements to at least three lines with linearly independent directions [26], the following system of equations can be solved using ordinary least squares to obtain an estimate for the camera's position in the global frame of reference:

$$\begin{bmatrix} {}^{C}\mathbf{w}_1^T\,{}^{C}_{G}\widehat{\mathbf{C}}\,\lfloor {}^{G}\boldsymbol{\ell}_1 \times \rfloor \\ {}^{C}\mathbf{w}_2^T\,{}^{C}_{G}\widehat{\mathbf{C}}\,\lfloor {}^{G}\boldsymbol{\ell}_2 \times \rfloor \\ \vdots \end{bmatrix} {}^{G}\hat{\mathbf{p}}_C = \begin{bmatrix} -{}^{C}\mathbf{w}_1^T\,{}^{C}_{G}\widehat{\mathbf{C}}\,{}^{G}\mathbf{m}_1 \\ -{}^{C}\mathbf{w}_2^T\,{}^{C}_{G}\widehat{\mathbf{C}}\,{}^{G}\mathbf{m}_2 \\ \vdots \end{bmatrix}. \tag{5.20}$$



(a)

(b)

(c)

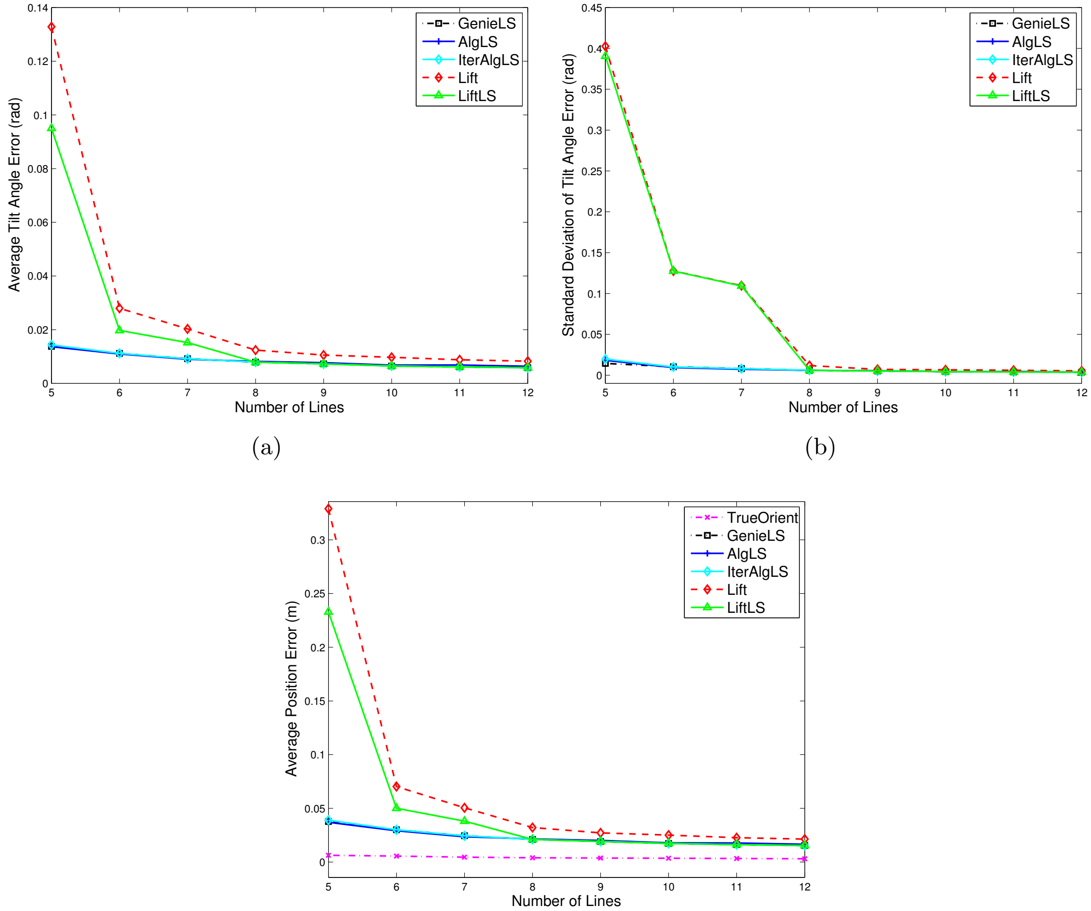
Figure 5.2: Monte Carlo simulation results for different standard deviations of the image noise when 5 lines are observed: (a) Average tilt-angle error; (b) Standard deviation of the tilt-angle error; and (c) Average position error.

5.6 Simulation and Experimental Results

5.6.1 Simulations

We hereafter present Monte Carlo simulation results that confirm the superior performance of our proposed pose-from-line-correspondences algorithm over existing approaches. Specifically, we compare the error in the estimated camera attitude obtained from each of the following algorithms:

- **Lift:** Lifting method of Ansar and Daniilidis [3].
- **LiftLS:** Weighted least squares proposed in [69], and initialized using the estimates from **Lift**.



(a)

(b)

(c)

Figure 5.3: Monte Carlo simulation results for different numbers of detected lines when the standard deviation of the image noise is 3 pixels: (a) Average tilt-angle error; (b) Standard deviation of the tilt-angle error; (c) Average position error.

- **AlgLS:** Our proposed single-step algebraic minimization of the relaxed least-squares cost function [see (5.10)].
- **GenieLS:** Weighted least-squares minimization of (5.3) initialized with the true orientation (used as a benchmark).
- **IterAlgLS:** Iterative algebraic minimization of the weighted least squares cost function [see (5.9)].

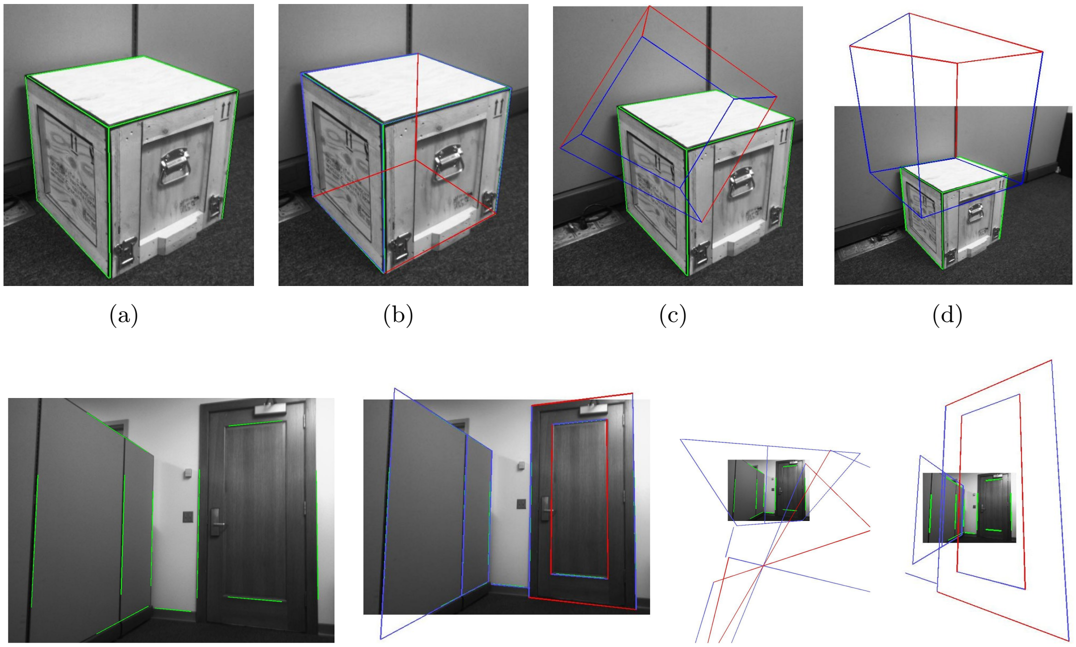
Note that **IterAlgLS** is our proposed extension to **AlgLS** for iteratively solving (5.9). The first iteration of **IterAlgLS** is the same as **AlgLS**. In the following iterations, the estimated orientation is used to approximate $\bar{\sigma}_i^2$, whose inverse appears in (5.9). This allows us to algebraically minimize the original cost function in (5.9) in a similar way to **AlgLS**.

We evaluate the performance of each algorithm by comparing the estimated and real pose. Specifically, we use the *norm of the tilt angle error* for assessing orientation accuracy. Let us denote by $\mathbf{C}$ and $\widehat{\mathbf{C}}$ the true and estimated rotation matrices, respectively. Then the orientation error is $\widetilde{\mathbf{C}} = \mathbf{C}^T\widehat{\mathbf{C}}$. After we convert $\widetilde{\mathbf{C}}$ to CGR parametrization and denote it as $\tilde{\mathbf{s}}$, then the tilt angle error can be obtained using the small-angle approximation as $||\boldsymbol{\delta\theta}|| = 2||\tilde{\mathbf{s}}||$. The error in the estimated position is $||\mathbf{p} - \hat{\mathbf{p}}||$.

The simulation setup is as follows: At each trial, the simulated pinhole camera with focal length of 512 pixels is placed at a random position and orientation with respect to the world. The camera measures pixelated projections of N randomly generated $3D$ line segments of different lengths, perturbed with i.i.d. random Gaussian noise with standard deviation of σ_p pixels. A least-squares line fitting is then employed to find the $2D$ line parameters ρ_i, θ_i from the pixelated line measurements.

We present two sets of simulation results. The first set demonstrates the performance of the aforementioned methods for different standard deviations of the image noise, σ_p, while fixing the number of observed lines to $N = 5$ (Fig. 5.2). The second set of results evaluates performance when varying the number of lines, N, while fixing the standard deviation of the image noise at $\sigma_p = 3$ pixels. The results correspond to 1000 trials for each value of N and σ_p (Fig. 5.3). In all simulations, when multiple global minimizers are obtained, we choose the one closest to the true camera pose. This is reasonable, since in practical situations we can often discard all but one minimizer by considering visibility constraints and re-projection errors.

Results from both simulations confirm the superior performance of our method compared to **Lift** and **LiftLS**. Specifically, **AlgLS** is almost always as good as the weighted least squares initialized with the true orientation (**GenieLS**). As it is expected, **Lift** is the worst in terms of accuracy since it does not account for noise. The least-squares algorithm initialized with the solution of lifting, labeled as **LiftLS**, has better accuracy compared to **Lift**; however, its performance is significantly inferior to our proposed method since it can diverge if its initialization is inaccurate. In Figs. 5.2(a) and 5.2(b) it can be seen that the performance of **Lift** and **LiftLS** quickly degrade as the image noise increases. However, **AlgLS** demonstrates significantly better robustness to noise. Similarly, Figs. 5.3(a) and 5.3(b) demonstrate that **Lift** and **LiftLS** perform very poorly



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

(e) (f) (g) (h)

Figure 5.4: Camera pose determination with respect to a box, and a corner inside an office, both of known dimensions. Manually selected lines are specified by green color and their back-projections are marked by blue. Projection of invisible (e.g., rear edges of the cube) or previously undetected (e.g., bottom edge of the door) lines are colored as red. (a,e): Initial selection of lines; (Back-) projection of the lines using the estimated camera pose from (b,f): **AlgLS**; (c,g): **Lift**; (d,h): **LiftLS**.

as the number of line measurements approaches the minimum required.[5]

Figs. 5.2(c) and 5.3(c) demonstrate the impact of estimating pose form line correspondences in two steps (i.e., first attitude and then position) on the performance of position estimation. In particular, observe that the error in the estimated position using orientation from **AlgLS** is significantly lower than **Lift**, and very close to the benchmark performance **TrueOrient** where the true orientation is used in (5.20). Finally, we point out that as evident from Figs. 5.2 and 5.3, the results obtained by **IterAlgLS** do not yield significant performance improvement compared to **AlgLS**.

Table 5.1: Computed orientation, expressed as CGR parameters, and average execution times for "cube" and "corner" experiments using different methods.

	cube (using 8 lines)	corner (using 10 lines)	exec. time
AlgLS	$[1.37 \;\; 3.96 \;\; -2.51]$	$[0.91 \;\; 1.26 \;\; -1.34]$	25 msec.
Lift	$[-6.99 \;\; 3.92 \;\; -0.79]$	$[-2.00 \;\; 11.94 \;\; 27.60]$	190 msec.
LiftLS	$[1.58 \;\; -0.55 \;\; 0.40]$	$[-1.10 \;\; 1.47 \;\; 1.38]$	194 msec.

5.6.2 Experiments

In order to validate the proposed algorithms in real situations, we have conducted a number of experiments. Specifically, we have taken an image of an object [wooden cube, see Fig. 5.4(a)] and of an indoor scene [corner of a room, see Fig. 5.4(e)] of known dimensions using an intrinsically-calibrated Dragonfly Express camera. We have then manually selected several lines in each image corresponding to the 3D lines with known coordinates [see Figs. 5.4(a) and 5.4(e)]. We have employed these line correspondences to estimate the camera orientation according to the procedure outlined in Section 5.4. In both experiments, four global minimizers are found for (5.10) and for each of them, the corresponding camera position is estimated (see Section 5.5). Among these candidate poses, the one that results in the scene to be in front of the camera is selected. Using this selected pose, the known 3D lines (including the ones that were not detected before, for example due to invisibility) are back-projected onto the image [see Figs. 5.4(b) and 5.4(f)] to validate the obtained results. The estimated camera poses from **Lift** and **LiftLS** are also used to plot the (back-)projection of the known 3D lines in Figs. 5.4(c), 5.4(g), and 5.4(d), 5.4(h), respectively. It can be clearly seen that in this experiment both **Lift** and **LiftLS** result in completely wrong camera poses, possibly due to the existence of multiple solutions for the camera orientation. Table 5.1 provides a summary of the estimated orientations (vector $\mathbf{s}$) by different methods along with their average execution times (for the corner and the cube experiments) from Matlab implementations on a 2 GHz Core 2 Duo processor. Note that the execution time of **LiftLS** includes initialization by **Lift**.

[5] Note that it is also possible to use the lifting algorithm of [3] with four lines. However, that would require a different implementation than when five lines or more are available, and hence it is not considered here. Note also that according to the simulation results in [3] the lifting algorithm's accuracy for four lines is considerably inferior to that of five or more lines.

5.7 Summary

In this chapter, we have presented an efficient algorithm for precisely estimating a camera's pose given observations of three or more known 3D lines. Contrary to previous algebraic approaches that solve a deterministic (noise-free) version of this problem, our formulation explicitly accounts for the presence of noise in the image measurements. Moreover, in contrast to existing nonlinear least-squares methods, which consider noisy observations but only guarantee convergence to a stationary point (through iterative minimization), our algorithm requires no initial estimate and is guaranteed to find the global minimum of the least-squares cost function for the orientation error. The key idea behind our approach is that the optimality conditions of the nonlinear least-squares problem form a system of multivariate polynomial equations which is directly solved, using algebraic geometry techniques, to determine all the critical points of the cost function, and thus the estimate (global minimum) that minimizes the orientation error. Once the camera's attitude is computed, its position is then determined using ordinary (linear) least squares. Extensive simulation and experimental results demonstrate that our algorithm significantly outperforms existing methods and achieves accuracy almost indistinguishable from that of an (ideal) iterative least-squares estimator initialized with the true camera orientation.

In the next chapter we relax the assumption of knowing the data association for the observed lines. Although in this case the camera's position cannot be determined, we will show that it may be possible to obtain a finite number of hypotheses for the orientation of the camera in an urban environment.

Chapter 6

Optimal Estimation of Vanishing Points and Focal Length in a Manhattan World[†]

6.1 Introduction

In the previous chapter we discussed how to determine the pose of a camera from perspective observation of known lines. In many practical situations, however, we do not know the coordinates of the observed lines. In these cases, it may be possible to partially determine the extrinsic calibration of the camera. It is well known that in the so-called Manhattan world [29], where the 3D lines are aligned with the cardinal axes of the global frame, the noise-free vanishing points are the scaled rows of the rotation matrix representing the camera's orientation with respect to the global frame [68]. This relationship has been commonly exploited to estimate a camera's orientation from vanishing points. In addition, the same relationships can be used to partially calibrate the camera, for example, by estimating its focal length [68].

In this chapter, we consider the problem of estimating vanishing points of a (partially) calibrated camera in a Manhattan world, where the line directions are predominantly orthogonal to each other. Furthermore, we study methods for simultaneously estimating the focal length of a partially calibrated camera from line observations. The state-of-the-art methods for this task [35, 135] are iterative, require accurate initialization, and are not guaranteed to converge to the global optimum. Additionally, existing

[†] A short version of this work is appeared at the IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 2011 [87].

initialization methods [2, 121] do not enforce orthogonality of the vanishing points, and thus may not produce sufficiently accurate estimates. To address these limitations, we introduce two methods for analytically determining the optimal orthogonal vanishing points and focal length (for a partially calibrated camera), and an efficient RANSAC-based line classifier to group lines into parallel and mutually orthogonal sets. In summary, the contributions of this work are:

- Optimal estimators for vanishing points and focal length that (i) are not iterative and do not require any initialization, (ii) are guaranteed to find all globally optimal estimates of the orthogonal vanishing points and the camera's orientation (in a least-squares sense), as well as the focal length of the camera (in the case of partially calibrated camera), and (iii) can work with as few as three or four lines (the minimal problems for fully or partially calibrated cameras, respectively) or as many as hundreds of lines, and its computational complexity is only linear in the number of lines.
- An efficient RANSAC-based line classifier that uses minimal sets of lines to generate hypotheses for all three orthogonal vanishing points and the focal length (for the case of a partially calibrated camera) at once. This RANSAC algorithm works robustly with very few sample measurements, and does not require a dominant line direction.

The remainder of this chapter is organized as follows. In Section 6.2, we provide an overview of the related literature. In Section 6.3, we introduce the notation that is used in this chapter, and describe the minimal and over-determined formulation of the problem for the case of fully calibrated and partially calibrated camera (i.e., unknown focal length). In Section 6.4, we investigate the multiplicity of the solutions both for fully and partially calibrated cameras. The RANSAC-based classification method is described in Section 6.5, and the experimental evaluation of the proposed algorithms is provided in Section 6.6. Finally, a summary of this chapter is provided in Section 6.7.

6.2 Related Work

Early work on vanishing-point estimation relied on the Hough transform of the line segments on the Gaussian sphere [10, 78]. These approaches, however, are not reliable in the presence of noise and outliers and may miss-classify lines into incorrect parallel groups [116]. Furthermore, these methods do not typically enforce the orthogonality constraints between the vanishing points leading to suboptimal estimation of both the vanishing points and the camera's orientation. The exhaustive search method of Rother [108] enforces the orthogonality of the vanishing points. However, its computational complexity is prohibitive for real-time applications.

To address the miss-classification and optimality issues, Expectation-Maximization (EM)-based methods assign a probability to each line segment or image region, indicating the likelihood that it belongs to each of the parallel line groups or an outlier group (expectation step); then, they find the most likely vanishing points from the line assignments (maximization step) [4, 68, 29, 114, 135, 35]. The EM approaches suffer from two common drawbacks: (i) they are iterative in nature, and sensitive to initialization, (ii) the maximization step entails optimization of a nonconvex cost function. The EM is typically initialized using results of the Hough transform or heuristic clustering of line-segment intersections, which are not guaranteed to produce a sufficiently accurate initialization. Additionally, the maximization step is usually performed using iterative algorithms, such as gradient descent, which are not guaranteed to converge to the global optimum, and may not find all global optima, if more than one exist.

More recently, RANSAC-based algorithms have been proposed that consider intersections of line segments as hypotheses for vanishing points and prune improbable hypotheses using heuristic criteria [2, 135]. In [42], such RANSAC hypotheses are used to initialize an iterative maximum-likelihood estimator of the vanishing points. In [121], a J-Linkage algorithm is used to generate hypothetical classes of parallel lines, followed by EM to find the vanishing points. These methods, however, do not enforce orthogonality of the vanishing points when generating the hypotheses, and generally require a large number of line segments and sample hypotheses to converge to the correct solution.

The case of partially calibrated camera has been studied in the literature as a follow-up stage to the orthogonal vanishing point determination. Specifically, in [68], [67] and [121] an algorithm is proposed to recover the focal length after determining the principal vanishing points. Since these methods do not enforce orthogonality of the cardinal vanishing points and determine the focal length in a separate stage, they are not guaranteed to optimally estimate the focal length and the vanishing points.

In order to address these limitations, in this chapter we first introduce two algorithms for estimating a camera's orientation and the orthogonal vanishing points, along with the focal length (for the case of partially calibrated camera), and then propose a RANSAC classifier that uses the proposed algorithms to partition line segments into parallel groups (and an outlier group).

Specifically, we study minimal and over-determined problems of vanishing point and focal length (for the case of partially calibrated camera) determination from line segment observations. We directly translate the minimal problem into a system of polynomial equations. For the over-determined problem, we formulate a polynomial least-squares estimator whose optimality conditions form a system of polynomial equations in the orientation of the camera. Using tools from algebraic geometry, we solve these polynomial systems to find either the solution to the minimal problems or the finite set of all the critical points of the least-squares cost function. For the over-determined problems, the orthogonal vanishing points are readily computed from the critical points that minimize the least-squares cost function. The developed algorithm is not iterative, does not require any initialization, and is guaranteed to find the globally optimal estimates of the camera's orientation and the orthogonal vanishing points. Furthermore, the proposed algorithm can work with as few as three or fours lines (the minimal case - depending on whether the focal length has to be estimated or not) or as many as hundreds of lines, with linear computational complexity in the number of lines.

In the second part of this chapter, we propose a RANSAC-based line classifier that uses minimal sample sets of line segments to generate hypotheses for orthogonal vanishing points and the focal length (for the case of unknown focal length). These hypotheses are voted upon by other line segments, and the winner candidate is chosen to classify other line segments into three parallel and mutually orthogonal groups.

6.3 Estimation of Vanishing Points

6.3.1 Preliminaries

Throughout this chapter, lines are parameterized with Plücker coordinates consisting of a 3×1 direction vector ${}^{C}\bar{\mathbf{n}}$ and a 3×1 moment vector ${}^{C}\mathbf{m} \triangleq {}^{C}\mathbf{p} \times {}^{C}\bar{\mathbf{n}}$ where ${}^{C}\mathbf{p} = \mathbf{K}^{-1}\mathbf{p}_m$ is any normalized point on the line ($\{C\}$ denotes the camera frame of reference). Here $\mathbf{K}$ is the matrix of intrinsic parameters of the camera [48], and $\mathbf{p}_m \triangleq [u \ v \ 1]^T$, where u and v are the pixel coordinates of the point on the line. The direction of the line segment can be obtained from its endpoints, ${}^{C}\mathbf{p}_1$ and ${}^{C}\mathbf{p}_2$ as ${}^{C}\bar{\mathbf{n}} = \frac{{}^{C}\mathbf{p}_2 - {}^{C}\mathbf{p}_1}{||{}^{C}\mathbf{p}_2 - {}^{C}\mathbf{p}_1||}$. Consequently, the moment vector may be directly obtained from the endpoints as

$$ {}^{C}\mathbf{m} = \frac{{}^{C}\mathbf{p}_1 \times {}^{C}\mathbf{p}_2}{||{}^{C}\mathbf{p}_2 - {}^{C}\mathbf{p}_1||} \tag{6.1} $$

It follows from the moment definition that ${}^C\mathbf{m}^T {}^C\bar{\mathbf{n}} = 0$. The direction of a moment vector uniquely specifies a plane (called the moment plane) that passes through the line and the origin of the frame of reference.

In this work, we model rotation matrices using the Cayley-Gibbs-Rodriguez (CGR) parameterization [117] as

$$\mathbf{C}(\mathbf{s}) = \frac{\bar{\mathbf{C}}(\mathbf{s})}{1+\mathbf{s}^T\mathbf{s}}\,, \quad \bar{\mathbf{C}}(\mathbf{s}) \triangleq \left((1-\mathbf{s}^T\mathbf{s})\mathbf{I}_3 + 2\lfloor \mathbf{s}\times\rfloor + 2\mathbf{s}\mathbf{s}^T\right). \tag{6.2}$$

where $\mathbf{s}^T = [s_1\ s_2\ s_3]$ is the vector of CGR parameters, and $\lfloor \mathbf{s}\times\rfloor$ is the corresponding skew-symmetric matrix. The CGR parameterization is suitable for expressing geometric constraints as polynomial equations, since the components of the rotation matrix are naturally expressed as rational-polynomial functions of the CGR parameters. In addition, CGR is a minimal representation of rotation (e.g., in contrast with unit quaternions), and thus, does not require additional constraints to ensure that it corresponds to a valid rotation. Furthermore, the CGR parameterization introduces the minimum number of unknowns in the resulting polynomial system and hence allows fast computation of its solutions.

6.3.2 Vanishing Points in a Calibrated Camera

When a camera is intrinsically calibrated, it can be modeled using spherical projection, measuring 3D lines as the intersection of the lines' moment planes in the camera frame with the Gaussian unit sphere. These intersections are great circles that are uniquely determined by the *direction* of the lines' moment planes in the camera's frame (see Fig. 6.1). In practice, the line segments are often extracted from the image gradient [22, 17, 40, 134], followed by least-squares refinement to obtain the lines' moment directions. In this section, we abstract away from the particular method used to detect lines, and assume that the camera directly measures the direction of the moment planes in its frame, i.e., ${}^C\bar{\mathbf{m}} = \frac{{}^C\mathbf{m}}{||{}^C\mathbf{m}||}$.

In a calibrated camera, a vanishing point associated with a line is the direction of the line in the camera's frame of reference, i.e., $\bar{\mathbf{v}} \triangleq {}^C\bar{\mathbf{n}}$. By definition, the vanishing point lies on the moment plane, i.e.,

$$\bar{\mathbf{v}}^T\, {}^C\bar{\mathbf{m}} = 0 \tag{6.3}$$

Note that parallel lines have the same direction, and thus share the same vanishing

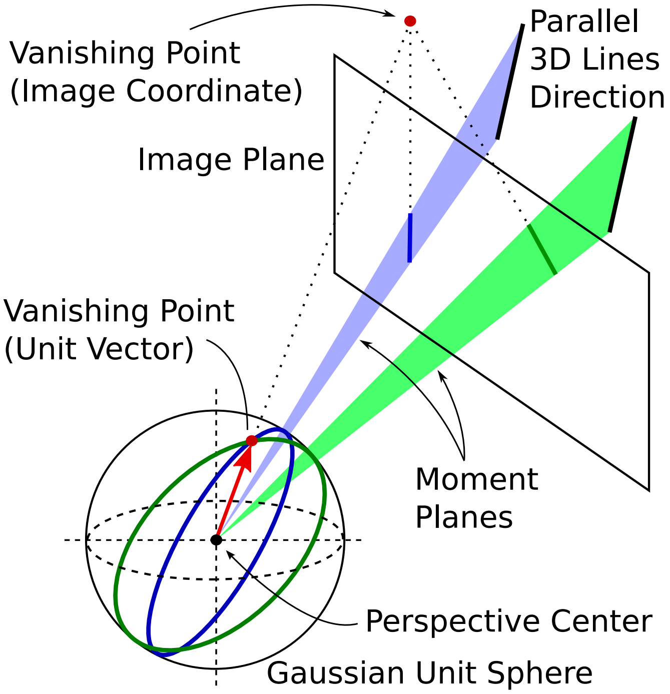


Figure 6.1: Illustration of the relationship between parallel lines in 3D, their corresponding line segments on the image plane, the moment planes, and the vanishing points.

point which corresponds to the intersection of their moment planes (see Fig. 6.1). In a Manhattan world, where 3D lines can be partitioned into three parallel and mutually orthogonal groups, the three vanishing points found in the image are also mutually orthogonal. In other words, if we denote them with $\bar{\mathbf{v}}_i,\ i = 1, 2, 3$, then $\bar{\mathbf{v}}_1^T \bar{\mathbf{v}}_2 = \bar{\mathbf{v}}_2^T \bar{\mathbf{v}}_3 = \bar{\mathbf{v}}_1^T \bar{\mathbf{v}}_3 = 0$. This enables us to model vanishing points of a Manhattan world as rows (or columns) of an orthonormal matrix. In particular, if we restrict the orthonormal matrix to be proper (by enforcing its determinant to be +1), it will represent the rotation between the Manhattan frame and the camera's frame of reference.[1] In this case, (6.3) can be written as:

$$\bar{\mathbf{e}}_i^T \, {}^G_C\mathbf{C} \, {}^C\bar{\mathbf{m}} = 0 \tag{6.4}$$

where $\bar{\mathbf{e}}_i$ is the one of the following cardinal vectors

$$\bar{\mathbf{e}}_1 = [1\ 0\ 0]^T, \quad \bar{\mathbf{e}}_2 = [0\ 1\ 0]^T, \quad \bar{\mathbf{e}}_3 = [0\ 0\ 1]^T$$

[1] Imposing this restriction does not lead to missing any of the vanishing points, since a proper orthonormal matrix can be obtained by negating rows (or columns) of an improper one. On the other hand, if $\bar{\mathbf{v}}$ is a vanishing point on the Gaussian sphere, so is $-\bar{\mathbf{v}}$ (see Fig. 6.1).

that corresponds to the direction of a 3D line in the Manhattan frame of reference. Note that in the remainder of this chapter, we refer to Manhattan world's vanishing points simply as vanishing points, except when explicitly expressed otherwise.

Deterministic Solution

As discussed in Section 6.3.1, estimating the three (or two[2]) orthogonal vanishing points is equivalent to estimating the rotation between the Manhattan frame and the camera frame. Therefore, it is easy to observe that the three mutually orthogonal vanishing points collectively constitute three degrees of freedom (d.o.f.). In order to uniquely determine these three d.o.f., at least three independent measurements are required. The conditions under which three line measurements are independent have been investigated in detail by Chen [26], and are summarized in Section 6.4.

Let us assume that the three measurement constraints are independent, and denote them as:

$$\bar{\mathbf{e}}_{\ell_i}^T \, {}^G_C\mathbf{C} \, {}^C\hat{\bar{\mathbf{m}}}_i = \eta_i, \quad i = 1, 2, 3, \;\; \ell_i \in \{1, 2, 3\} \tag{6.5}$$

where ${}^C\hat{\bar{\mathbf{m}}}_i$ is the measured normal direction of the moment plane, and η_i denotes the unknown residual corresponding to the cosine of the angle between the measured moment plane and the line direction. For now, we also assume that the direction $\bar{\mathbf{e}}_{\ell_i}$ corresponding to the observation ${}^C\hat{\bar{\mathbf{m}}}_i$ is known in advance. In Section 6.5, we will describe how to find the correspondence using RANSAC. In order to compute the *deterministic* solution to (6.5), we ignore the measurement noise in ${}^C\hat{\bar{\mathbf{m}}}_i$ and set $\eta_i = 0$. Representing the unknown rotation matrix ${}^G_C\mathbf{C}$ using CGR parameterization and multiplying both sides of (6.5) with $1 + \mathbf{s}^T\mathbf{s}$ yields:

$$p_i(\mathbf{s}) = \bar{\mathbf{e}}_{\ell_i}^T \, \bar{\mathbf{C}}(\mathbf{s}) \, {}^C\hat{\bar{\mathbf{m}}}_i = 0, \quad i = 1, 2, 3, \;\; \ell_i \in \{1, 2, 3\} \tag{6.6}$$

Since $\bar{\mathbf{C}}(\mathbf{s})$ comprises quadratic polynomials in $\mathbf{s}$ [see (6.2)], each $p_i(\mathbf{s})$ will also be a quadratic polynomial in $\mathbf{s}$. The system of polynomials $p_i(\mathbf{s})$, $i = 1, 2, 3$, in general has eight solutions which can be computed using the method described in Section 4.4. Note that this system of polynomials, consisting of three quadratic equations is sufficiently simple to solve in closed form (see, e.g., [26]). Note, however, that regardless of the solution method, it is well known that computing solutions of polynomial systems may

[2] Note that the third vanishing point (and the third row of the rotation matrix) can be obtained as the cross product of the first two.

not be well conditioned; often small perturbations in the coefficients of the polynomial (due to setting $\eta_i = 0$) may drastically displace the roots or even lead the polynomial system to have no real-valued solutions [30, Ch. 2].

Least-Squares Solution

To counter the adverse effect of the noise, we can explicitly take it into account and minimize its impact through a least-squares framework. Assume that ${}^{C}\hat{\bar{\mathbf{m}}}_i, i \in \mathcal{M} = \{1, \ldots, N\}$ are the measurements of 3D Manhattan-world lines observed by an intrinsically-calibrated camera. For now we assume that the lines are classified into three mutually orthogonal groups, and that no outlier exists among them. In Section 6.5, we describe how to relax these assumptions. Given a partitioning of $\mathcal{M}$ into $\mathcal{M}_j,\ j = 1, 2, 3$, sets each representing an orthogonal group, the optimal (in a weighted least-squares sense) values for the vanishing points on the Gaussian sphere are obtained by solving the following constrained least-squares problem:

$$\mathcal{P}_1 : \ \bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2, \bar{\mathbf{v}}_3 = \min_{\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2, \bar{\mathbf{v}}_3} \frac{1}{2} \sum_{j=1}^{3} \sum_{i \in \mathcal{M}_j} \sigma_i^{-2} (\bar{\mathbf{v}}_j^{T} \, {}^{C}\hat{\bar{\mathbf{m}}}_i)^2 \tag{6.7a}$$

$$\text{subject to} \quad \bar{\mathbf{v}}_1^T \bar{\mathbf{v}}_2 = \bar{\mathbf{v}}_2^T \bar{\mathbf{v}}_3 = \bar{\mathbf{v}}_1^T \bar{\mathbf{v}}_3 = 0 \tag{6.7b}$$

$$||\bar{\mathbf{v}}_1|| = ||\bar{\mathbf{v}}_2|| = ||\bar{\mathbf{v}}_3|| = 1 \tag{6.7c}$$

where σ_i are weights reflecting the uncertainty in each line-moment observation. As discussed in Section 6.3.1, the constraints in (6.7b)-(6.7c) can be satisfied, if we choose the vanishing points as the rows of a rotation matrix. Thus, we can rewrite $\mathcal{P}_1$ as:

$$\mathcal{P}_1' : \quad {}^{G}_{C}\hat{\mathbf{C}} = \min_{\mathbf{C}} \frac{1}{2} \sum_{j=1}^{3} \sum_{i \in \mathcal{M}_j} \sigma_i^{-2} (\bar{\mathbf{e}}_j^T \mathbf{C} \, {}^{C}\hat{\bar{\mathbf{m}}}_i)^2 \tag{6.8a}$$

$$\text{subject to } \mathbf{C}^T \mathbf{C} = \mathbf{I}_3, \ \det(\mathbf{C}) = 1 \tag{6.8b}$$

Note that this optimization problem can also be obtained directly by minimizing the residuals η_i in (6.5) for $\mathcal{M}$ observations.

This nonlinear weighted least-squares problem for $N \geq 3$ can be solved using iterative methods such as Gauss-Newton [48]. However, in the absence of an accurate initial estimate, iterative approaches may converge to local minima, and are not guaranteed to find all global minima. To address these limitations, we hereafter present an algebraic method that *directly* solves the nonlinear least-squares problem without requiring initialization.

We start by expressing the rotation matrix in (6.8) using the CGR parameterization, and obtain:

$$\mathcal{P}_2 : \; {}^{G}_{C}\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} J, \quad J = \frac{1}{2}\sum_{i=1}^{N} \sigma_i^{-2} \left(\bar{\mathbf{e}}_i^T \, \mathbf{C}(\mathbf{s}) \, {}^{C}\hat{\bar{\mathbf{m}}}_i \right)^2 . \tag{6.9}$$

Compared to $\mathcal{P}_1'$ [see (6.8)] the optimization constraint is now removed since the CGR parameterization ensures that $\mathbf{C}(\mathbf{s})$ is a rotation matrix. To algebraically find the global minimum of $\mathcal{P}_2$, we first determine all the critical points of J by solving the following optimality conditions, and then choose the one(s) that minimize $\mathcal{P}_2$. To derive the optimality conditions, we first factor out $(1 + \mathbf{s}^T\mathbf{s})$ from J in $\mathcal{P}_2$:

$$J = \frac{J'}{(1 + \mathbf{s}^T\mathbf{s})^2}, \;\; J' \triangleq \frac{1}{2}\sum_{i=1}^{N} \sigma_i^{-2} \left(\bar{\mathbf{e}}_i^T \, \bar{\mathbf{C}}(\mathbf{s}) \, {}^{C}\hat{\bar{\mathbf{m}}}_i \right)^2 . \tag{6.10}$$

Then, the optimality conditions of $\mathcal{P}_2$ are:

$$\frac{\partial J}{\partial s_j} = \frac{1}{(1 + \mathbf{s}^T\mathbf{s})^3} \left((1 + \mathbf{s}^T\mathbf{s}) \frac{\partial J'}{\partial s_j} - 4 s_j J' \right) = 0 \tag{6.11a}$$

$$\frac{\partial J'}{\partial s_j} = \sum_{i=1}^{N} \sigma_i^{-2} \left(\bar{\mathbf{e}}_i^T \bar{\mathbf{C}} \, {}^{C}\hat{\bar{\mathbf{m}}}_i \right) \frac{\partial}{\partial s_j} \left(\bar{\mathbf{e}}_i^T \bar{\mathbf{C}} \, {}^{C}\hat{\bar{\mathbf{m}}}_i \right) \tag{6.11b}$$

for $j = 1, 2, 3$ and $N \geq 3$. Considering that $(1 + \mathbf{s}^T\mathbf{s})$ is nonzero for real $\mathbf{s}$, we simplify the optimality conditions as

$$f_j(\mathbf{s}) = (1 + \mathbf{s}^T\mathbf{s}) \frac{\partial J'}{\partial s_j} - 4 s_j J' = 0, \quad j = 1, 2, 3. \tag{6.12}$$

These optimality conditions are fifth-order polynomials in the elements of $\mathbf{s}$ whose real variety (i.e., solutions) comprises the critical points of $\mathcal{P}_2$. Directly solving these polynomials, however, is challenging since the ideal generated by them turns out to be non-zero dimensional, due to having a continuous variety on the imaginary hypersphere defined by $1 + \mathbf{s}^T\mathbf{s} = 0$.[3] To overcome this challenge, we introduce the following auxiliary polynomial that removes the (complex) solutions of $1 + \mathbf{s}^T\mathbf{s} = 0$ from the variety

[3] This is verified by computing the Hilbert dimension of the ideals generated by instances of the problem with integer or rational coefficients. Saturating these ideals with $1 + \mathbf{s}^T\mathbf{s}$ makes them zero dimensional.

of $f_j = 0$:

$$f_0(s_0, \mathbf{s}) = s_0(1 + \mathbf{s}^T \mathbf{s}) - 1 = 0 \tag{6.13}$$

where s_0 is a new auxiliary variable. Note that this polynomial can be satisfied only if $1 + \mathbf{s}^T\mathbf{s}$ is nonzero. The new *saturated* system of polynomial equations consists of three fifth-order equations [see (6.12)] and one cubic equation [see (6.13)], in four unknowns $(s_0, \mathbf{s})$. Assuming that at minimum three lines are observed, from which at least two are nonparallel, this polynomial system will have 40 solutions that can be computed by the method described in Section 4.4. The globally optimal estimates are simply the solutions that minimize $\mathcal{P}_2$ [see (6.9)].

Note that the computational complexity of solving the saturated polynomial system and finding the global minimum does not increase with the addition of measurements, since the degree and number of polynomials expressing the optimality conditions are fixed. Moreover, computing the contribution of all measurements to the coefficients of the polynomials f_j, $j = 1, 2, 3$ increases only linearly with the number of measurements.

Relaxed Least-Squares Solution

When the computational resources are limited, solving the system of polynomial equations described in (6.12) and (6.13) may be too expensive (see Section 4.4 for details on the computational complexity). In this situation, we may relax the original problem $\mathcal{P}_2$ in (6.8) by requiring $\mathbf{C}$ to be only orthogonal and not necessarily orthonormal. The relaxed problem is

$$\mathcal{P}_3 : \quad {}^G_C\hat{\bar{\mathbf{C}}} = \arg\min_{\bar{\mathbf{C}},\beta} \frac{1}{2} \sum_{i=1}^{N} \sigma_i^{-2} \left(\bar{\mathbf{e}}_i^T \, \bar{\mathbf{C}} \, {}^C\hat{\bar{\mathbf{m}}}_i \right)^2 \tag{6.14a}$$

$$\text{subject to} \quad \bar{\mathbf{C}}^T \bar{\mathbf{C}} = \beta^2 \mathbf{I}_3, \;\; \det(\bar{\mathbf{C}}) \geq 1. \tag{6.14b}$$

This is equivalent to relaxing (6.7c) as $||\mathbf{v}_1|| = ||\mathbf{v}_2|| = ||\mathbf{v}_3|| \geq 1$. Although $\bar{\mathbf{C}}$ is not a rotation matrix, a valid rotation matrix can be easily obtained as $\bar{\mathbf{C}}/\beta$. Parameterizing $\bar{\mathbf{C}}$ using CGR parameters yields:

$$\mathcal{P}_3' : \quad {}^G_C\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} J' \tag{6.15}$$

where J' is defined in (6.10), and $\beta = 1 + \mathbf{s}^T\mathbf{s}$. An alternative interpretation of this cost function can be obtained by formulating a least-squares cost function that minimizes the residuals of the following constraints:

$$\bar{\mathbf{e}}_{\ell_i}^T \, \bar{\mathbf{C}}(\mathbf{s})^C \hat{\bar{\mathbf{m}}}_i = \bar{\eta}_i, \quad i = 1, \ldots, \mathcal{M}, \; \ell_i \in \{1, 2, 3\} \tag{6.16}$$

In contrast to η_i in (6.5), the residual $\bar{\eta}_i$ is purely algebraic and does not have a geometric interpretation.

The cost function J' is a 4th-order polynomial in the elements of $\mathbf{s}$. To algebraically find the global minimum of (6.15), we first determine all the critical points of J' by solving the optimality conditions $\frac{\partial J'}{\partial s_j} = 0$ for $j = 1, 2, 3$ and $N \geq 3$ [see (6.11b)]. These three relaxed optimality conditions are always *cubic* polynomials, regardless of the number of measurements, and generally generate a zero-dimensional ideal with 27 solutions that can be obtained from the corresponding multiplication matrix (see Section 4.4). Among these solutions (critical points of J'), we choose the ones that minimize J (and not J') as the relaxed estimates for the globally optimal solutions of $\mathcal{P}_2$.

6.3.3 Vanishing Points in a Camera with Unknown Focal Length

In the preceding section, we assumed the camera to be intrinsically calibrated, allowing us to model line measurements on the Gaussian sphere. If the focal length of the camera is unknown, the line measurements cannot be directly represented on the Gaussian sphere. Instead, we have to directly account for the focal length and estimate it along with the vanishing points.

Let us assume that the camera is partially calibrated such that its center point is known, and all the measured pixels are translated so that the effective center point is $[0 \;\, 0]^T$. Assuming that the camera's skew coefficient is zero, the calibration matrix is simply:

$$\mathbf{K} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.17}$$

where f is the unknown focal length of the camera. Attempting to estimate f directly along with the vanishing point leads to bad conditioning of the problem, since the value of f may be several orders of magnitude larger than the rest of unknowns. To mitigate this issue, we instead estimate the normalized focal length $\alpha = \frac{f_o}{f}$ where f_o is a constant nominal value (e.g., 500 pixels). In this case the camera calibration matrix can be expressed as

$$\mathbf{K} = \begin{bmatrix} \frac{f_o}{\alpha} & 0 & 0 \\ 0 & \frac{f_o}{\alpha} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.18}$$

If we denote $\bar{u}_i = \frac{u_i}{f_o}$ and $\bar{v}_i = \frac{v_i}{f_o}$ for $i = 1, 2$, the *scaled* moment of a line segment can be written as [see (6.1)]

$$\begin{aligned} {}^{C}\mathbf{m} = \mathbf{K}^{-1}\mathbf{p}_1 \times \mathbf{K}^{-1}\mathbf{p}_2 &= \alpha \begin{bmatrix} \bar{v}_1 - \bar{v}_2 \\ \bar{u}_2 - \bar{u}_1 \\ \alpha(\bar{u}_1\bar{v}_2 - \bar{u}_2\bar{v}_1) \end{bmatrix} \\ &= \alpha \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \alpha \end{bmatrix}}_{\mathbf{\Pi}(\alpha)} \underbrace{\begin{bmatrix} \bar{v}_1 - \bar{v}_2 \\ \bar{u}_2 - \bar{u}_1 \\ \bar{u}_1\bar{v}_2 - \bar{u}_2\bar{v}_1 \end{bmatrix}}_{{}^{C}\mathbf{m}'} \end{aligned} \tag{6.19}$$

Note that ${}^{C}\mathbf{m}'$ depends only on the normalized endpoints of the line segments, and not on the unknown focal length. Similar to Section 6.3.2, by considering that the vanishing point of a line lies on the lines' moment plane, we can form the following constraint for the vanishing point and the focal length:

$$\bar{\mathbf{v}}^T \, \mathbf{\Pi}(\alpha) {}^{C}\mathbf{m}' = 0 \tag{6.20}$$

where the α coefficient of the moment direction is dropped, since it is always nonzero in practice. Expressing the three orthogonal vanishing points of the Manhattan world as rows of a rotation matrix yields:

$$\bar{\mathbf{e}}_j^T \, {}^{G}_{C}\mathbf{C} \, \mathbf{\Pi}(\alpha) {}^{C}\mathbf{m}' = 0 \tag{6.21}$$

In the presence of noise, this measurement constraint is typically equal to a nonzero residual:

$$\bar{\mathbf{e}}_j^T \, {}^{G}_{C}\mathbf{C} \, \mathbf{\Pi}(\alpha) {}^{C}\hat{\mathbf{m}}' = \epsilon \tag{6.22}$$

Deterministic Solution

Similar to Section 6.3.2, a deterministic solution for the vanishing points and the focal length can be computed by ignoring the noise in ${}^C\hat{\mathbf{m}}'$ and setting $\epsilon = 0$. As expected, due to the inclusion of one more unknown, namely α, at least four line measurements are required to estimate the unknowns. Assuming that four line observations are available, we represent the rotation matrix ${}^G_C\mathbf{C}$ using CGR parameterization, and multiply both sides of (6.22) with $1 + \mathbf{s}^T\mathbf{s}$ to obtain:

$$f_i(\mathbf{s}, \alpha) = \bar{\mathbf{e}}_{\ell_i}^T \, \bar{\mathbf{C}}(\mathbf{s}) \, \mathbf{\Pi}(\alpha)^C\hat{\mathbf{m}}_i' = 0, \;\; i = 1, \ldots, 4, \;\; \ell_i \in \{1, 2, 3\} \tag{6.23}$$

As the components of $\bar{\mathbf{C}}(\mathbf{s})$ are quadratic in $\mathbf{s}$, (6.23) comprises four cubic polynomials in $\mathbf{s}$ and α. This system of polynomial equations generally has 32 solutions which can be computed using the method described in Section 4.4.

Note that, in the presence of noise, the equations in (6.23) are not equal to zero, and trying to solve them by assuming the residual is zero leads to inaccurate estimates of the unknowns.

Relaxed Least-Squares Solution

To counter the impact of the noise, we can choose to estimate the unknown vanishing points and the focal length in a least-squares framework, similar to that of Section 6.3.2. In contrast with the calibrated camera, however, we do not minimize the geometric residuals that correspond to the deviation of the angle between the vanishing point and the moment plane from $\pi/2$. This is due to the fact that ${}^C\mathbf{m}$ is not unit-norm [see (6.19)], and normalizing it will introduce the non-polynomial term $||\mathbf{\Pi}(\alpha)^C\mathbf{m}'||$ into the measurement constraints, and in turn, the optimality conditions. Instead, we minimize the purely algebraic residuals, ϵ_i, of (6.22):

$$\mathcal{P}_4 : \; {}^G_C\hat{\mathbf{s}}, \hat{\alpha} = \arg\min_{\mathbf{s},\alpha} \frac{1}{2} \sum_{i=1}^{N} \sigma_i^{-2} \left(\bar{\mathbf{e}}_i^T \, \bar{\mathbf{C}}(\mathbf{s}) \, \mathbf{\Pi}(\alpha)^C\hat{\mathbf{m}}_i' \right)^2 \tag{6.24}$$

Similar to Section 6.3.2, taking the derivative of the cost function in this equation with respect to the unknowns yields the optimality conditions:

$$h_j = \sum_{i=1}^{N} \left(\bar{\mathbf{e}}_i^T \, \bar{\mathbf{C}}(\mathbf{s}) \, \mathbf{\Pi}(\alpha)^C\hat{\mathbf{m}}_i' \right) \frac{\partial}{\partial s_j} \left(\bar{\mathbf{e}}_i^T \, \bar{\mathbf{C}}(\mathbf{s}) \, \mathbf{\Pi}(s_0)^C\hat{\mathbf{m}}_i' \right) = 0 \tag{6.25}$$

for $j = 0, \ldots, 3$, where for brevity we have redefined $s_0 \triangleq \alpha$. Regardless of the number of lines observed, these optimality conditions constitute a system of polynomial equation of degree five in four unknowns, and generally generate a zero-dimensional ideal with 243 solutions. We find these solutions from the corresponding multiplication matrix [see Section 4.4], and choose the one minimizing the cost function as the global optimum.

6.4 Existence and Multiplicity of Solutions

6.4.1 Case I: Calibrated Camera

Several works exist that study the conditions for having a finite number of solutions to the *deterministic* rotation-estimation problem, along with characterization of the maximum number of solutions. The most relevant to this paper is the work of Chen [26], where the camera's orientation estimation from line observations with known data association is discussed. In particular, he has shown that in order to have a finite number of solutions for the camera's orientation from observations of three 3D lines, the following scenarios should be avoided:

1. All three lines are parallel.

2. All three moment planes are parallel.

3. Two of the lines are parallel, and their moment planes are also parallel.

4. Two of the moment planes are parallel and perpendicular to the third moment plane; and the lines corresponding to the parallel moment planes are perpendicular to the third line.

5. Two of the lines are parallel and perpendicular to the third line; and the moment planes of the first two lines are perpendicular to that of the third line.

We are interested in the realization of these singular scenarios in a Manhattan world. Let us first argue that the 5th case rarely occurs in practice. Note that for this case to hold, the third moment plane should be perpendicular to the intersection of the first two moment planes, and therefore to the direction of the two parallel lines (i.e., their vanishing point). This means that for the observation of every two parallel lines, there is just *one* possible measurement (among infinitely many) for the third line that results in the singularity of the 5th kind. We conclude that the probability of this event is zero, and focus on the other four singular configurations.

The common property of the 2nd, 3rd, and 4th kind is that in all of them at least two of the measured moment planes are parallel. Since moment planes should all pass through the camera's center, if two moment planes are parallel, they are simply the same. This observation implies that these singularities occur if the observed line segments are collinear. One practical method to avoid these singularities is to ensure that the normal vector of the three moment planes that are used to estimate the orientation are not too close to each other by examining their inner products.

The most frequently encountered type of singularity is of the 1st type, arising from the observation of three parallel lines. This type of singularity cannot be reliably detected from the measurements, since depending on the location of the corresponding vanishing points, the measured line segments may appear in various configurations. This type of singularity, however, can be easily detected when attempting to solve the polynomial system in (6.6), as it leads to a rank-deficient $\mathbf{R}_1$ in (4.42).

In nonsingular scenarios, Chen has shown that up to eight deterministic solutions for the camera's orientation may exist [26]. Following the method proposed in [98], it is easy to show that regardless of the number of measurements, the solutions for a camera's orientation from observations of lines with known directions in a Manhattan world always appear in groups of four, corresponding to one distinct triplet of cardinal vanishing points. Specifically, the noise-free observation of $N \geq 3$ Manhattan lines provides the following constraints:

$$\bar{\mathbf{e}}_{\ell_i}^T \, {}^G_C\mathbf{C} \, {}^C\bar{\mathbf{m}}_i = 0, \quad i = 1, \ldots, N, \quad \ell_i \in \{1, 2, 3\} \tag{6.26}$$

Assuming that ${}^G_C\mathbf{C} = \mathbf{C}_1$ is one solution to this set of equations, it is easy to verify that ${}^G_C\mathbf{C} = \mathbf{R}_k \, \mathbf{C}_1, \ k = 1, \ldots, 4$, for

$$\mathbf{R}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \ \mathbf{R}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix},$$
$$\mathbf{R}_3 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \ \mathbf{R}_4 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.27}$$

also satisfies the same set of equations. Clearly, any of the choices for $\mathbf{R}_k$ only reverses the direction of two of the vanishing points from $\bar{\mathbf{v}}_j$ to $-\bar{\mathbf{v}}_j$ (corresponding to two rows of ${}^{G}_{C}\mathbf{C}$). Hence, all the four orientations correspond to the same set of cardinal vanishing points. Considering that (6.26) has at most eight roots, it yields at most two distinct solutions for the set of cardinal vanishing points.

The singular cases for the deterministic problem, when observing three 3D lines, are also singular for the least-squares configuration. This is justified by noting that singularities are in fact cases that the ideal defined by the noise-free measurement constraints in (6.6) is nonzero dimensional; in other words the solutions for $\mathbf{s}$ comprise a continuum, instead of a finite set. Since the least-squares cost function sums up the (square of) individual measurement constraints, it will attain zero (i.e., the global minimum) where the individual noise-free constraints are zero. Therefore, the continuum of solutions for the noise-free measurement constraints translates into a continuum of global minima for the least-squares cost function, where the value of $\mathbf{s}$ cannot be discretely identified.

6.4.2 Case II: Partially Calibrated Camera with Unknown Focal Length

A solution for this problem cannot be obtained if, even by assuming that the focal length is known, the vanishing points cannot be estimated. Hence, the singular scenarios for estimating vanishing points in a calibrated camera, should be avoided in a partially calibrated camera as well. Determining other possible singular cases and sufficient conditions for avoiding them remains part of our future work.

In nonsingular scenarios, the number of solutions to the deterministic problem (see Section 6.3.3) is 32, which is equal to the cardinality of the normal set in integer instances of the problem. The number of solutions is typically smaller when more than four lines are available and we use least-squares formulation. Using the technique of the previous section, we can show that regardless of the number of measurements, the solutions for the vanishing points and the focal length appear in groups of four. Specifically, the noise-free observation of $N \geq 4$ Manhattan lines provides the following constraints:

$$\bar{\mathbf{e}}_{\ell_i}^T \, {}^{G}_{C}\mathbf{C} \, \mathbf{\Pi}(\alpha) \, {}^{C}\bar{\mathbf{m}}_i' = 0, \quad i = 1, \ldots, N, \quad \ell_i \in \{1, 2, 3\} \tag{6.28}$$

If $\{{}^{G}_{C}\mathbf{C} = \mathbf{C}_1, \alpha = \alpha_1\}$ is a solution to the above constraints, so are $\{{}^{G}_{C}\mathbf{C} = \mathbf{R}_k\mathbf{C}_1, \ \alpha = \alpha_1\}$ and $\{{}^{G}_{C}\mathbf{C} = \mathbf{R}_k\mathbf{C}_1\mathbf{R}_4, \ \alpha = -\alpha_1\}$ for $\mathbf{R}_k, \ k = 1, \ldots, 4$, defined in (6.27). Note that the latter sets of solutions, obtained by using $\alpha = -\alpha_1$ and post-multiplication with $\mathbf{R}_4$, are equivalent to simply multiplying the constraints with -1. The reasoning for pre-multiplication with $\mathbf{R}_k$ is similar to the previous section. Hence, all eight solutions correspond to the same set of canonical vanishing points and focal length. Considering that (6.28) has at most 32 roots, it yields at most four distinct solutions for the cardinal vanishing points and the focal length. We note, however, that in practice some of these solutions may be complex, and the number of real distinct solutions may become less than four.

6.5 Classification of Lines

6.5.1 Plain RANSAC-based Classification

In the previous sections, we assumed that the lines are already partitioned into groups, each parallel to one of the Manhattan directions. To relax this assumption, in this section we present a RANSAC-based line classifier that exploits the deterministic methods provided in the previous sections to generate hypotheses for the location of orthogonal vanishing points and the focal length (if applicable), and classify lines into parallel and mutually orthogonal groups.

The *existing* approaches for RANSAC-based classification of lines use the intersection of two image lines (or their extensions) to generate hypotheses for *individual* vanishing points [2, 135]. Since the vanishing points are detected sequentially (i.e., one after the other), at each step all lines are considered outliers unless they correspond to the dominant line direction. Once a dominant vanishing point is detected, all lines associated with it are removed from the image, and the procedure is repeated to detect the next dominant vanishing point. Besides assuming the existence of a dominant direction, these methods require a large number of hypotheses to compensate for the lines that do not pass through the dominant vanishing point (even if they are along other cardinal directions). Moreover, the vanishing points determined in this way are not generally orthogonal.

In this section, we propose a more efficient and robust RANSAC-based approach that generates hypotheses for all three orthogonal vanishing points and the focal length (if applicable) at once. Specifically, we randomly sample triplets or quadruplets of lines (depending on whether the focal length is known or not), and then consider all possible configurations for their directions. For each triplet of lines, one configuration assumes that each line is along one cardinal direction, and three configurations assume that two lines (out of three) are along one cardinal direction, while the third line is along another cardinal direction. For the quadruplets of lines we extend these configurations by assuming that the fourth line is along each of the cardinal directions, and obtain 12 possible configurations.

In the next step, given the possible configurations for each sample set, we employ the method described in Sections 6.3.2 and 6.3.3 to compute hypotheses for all three orthogonal vanishing points and the focal point (if applicable). For the case of the calibrated camera, each configuration of the sample set leads to at most two hypotheses for the vanishing points (see Section 6.4.1), and therefore each sample triplet will results in at most eight hypotheses. For the partially calibrated camera, each configuration of the sample quadruplets results in at most four hypotheses for the vanishing points and the focal length (see Section 6.4.2). Therefore, each sample quadruplet leads to at most 48 hypotheses for the focal lengths and vanishing points.[4]

After processing sufficient number of sample triplets or quadruplets, we will have M hypotheses for the three orthogonal vanishing points and the focal length (if applicable), denoted as $\bar{\mathbf{v}}_{i,\ell},\ i = 1, 2, 3$, and $\alpha_\ell,\ \ell = 1, \ldots, M$. For the calibrated camera, we measure the angle between the jth line's moment plane from each vanishing point as $\sin^{-1}(\bar{\mathbf{v}}_{i,\ell}^T {}^C\hat{\bar{\mathbf{m}}}_j)$. For the partially calibrated camera, we first have to normalize the moments using the estimated focal length. Therefore, we measure the angle as $\sin^{-1}\left(\bar{\mathbf{v}}_{i,\ell}^T \mathbf{\Pi}(\alpha_\ell)^C\hat{\bar{\mathbf{m}}}_j / ||\mathbf{\Pi}(\alpha_\ell)^C\hat{\bar{\mathbf{m}}}_j||\right)$. If this angle is smaller than a prespecified threshold for any of the three orthogonal vanishing points of the ℓth hypothesis and the focal length (if applicable), we label the jth line as inlier with respect to the ℓth hypothesis. In this case, within the ℓth hypothesis, we classify the jth line as belonging to the vanishing point that generated the smallest angle. The winner of the RANSAC algorithm is the hypothesis that results in the largest number of inliers. Once the inliers are determined and grouped into three perpendicular groups, we may employ the methodology described in Section 6.3.2 for calibrated cameras, and Section 6.3.3 for partially calibrated cameras to analytically estimate the unknowns.

6.5.2 Number of Required Sample Line Segments

When the camera is calibrated, the sample triplets that do not lead to any valid hypotheses must either include one (or more) line(s) with non-Manhattan direction, or follow one of the singular configurations listed in Section 6.4.1. To ensure that singular cases of the 2nd, 3rd, and 4th kind do not occur, we examine the direction of the moment planes for each sample triplet, and discard it if any two of them are too close to each other (without attempting to solve for the vanishing points). Considering that the

[4] Note that this is an upper bound on the number of generated hypotheses from a sample quadruplet. During our experiments, we observed that usually only one of the computed solutions for each configuration is real, and the rest are complex. Therefore, the total number of hypotheses for the vanishing points and focal lengths for a sample quadruplet is usually 12.

5th kind of singularity rarely happens in practice, the only common scenario where a sample triplet of lines with Manhattan directions does not lead to a valid hypothesis is when all three observed lines are parallel. We point out that the set of such triplets is, in general, significantly smaller than the set of samples generating invalid hypotheses for the intersection-based RANSAC classifiers [2, 135].

To clarify this, consider the following example: Assume we observe 10 lines per cardinal direction of a Manhattan environment and no non-Manhattan lines. Also, assume that none of the observed lines' moment planes are close to parallel. For the proposed RANSAC method, a sample triplet results in an invalid hypotheses if all the drawn lines are along the same direction. The portion of such triplets to all possible triplets is $\frac{3\times10\times9\times8}{30\times29\times28} = 9\%$. In other words, 91% of the possible triplets will lead to an inlier hypothesis. On the other hand, for the intersection-based RANSAC, a sampled couple will lead to a valid hypothesis for the vanishing point, only if both lines are along the same direction. Consequently, for the first vanishing point, only $\frac{3\times10\times9}{30\times29} = 31\%$ of the possible samples lead to a valid hypothesis. Moreover, with respect to a valid hypothesis, only $\frac{8}{28} = 29\%$ of the remaining lines are inliers when we use intersection-based RANSAC. In contrast, all the remaining lines are inliers with respect to a valid hypothesis generated by the proposed RANSAC method.

This analysis can be repeated for a scenario more favorable to the intersection-based RANSAC when there exist a dominant line direction. For example consider when out of the 30 observed lines, 20 are along the dominant direction, and 5 are along each other direction. In this case, only $\frac{20\times19}{30\times29} = 44\%$ of the possible couples lead to a valid hypothesis corresponding to the dominant vanishing point in the intersection-based RANSAC, while for the proposed RANSAC $1 - \frac{20\times19\times18+2\times5\times4\times3}{30\times29\times28} = 71\%$ of all possible triplets are valid. Moreover, for the intersection-based RANSAC, only $\frac{18}{28} = 64\%$ of the remaining lines are inliers with respect to a valid hypothesis for the dominant vanishing point, while for the proposed RANSAC all the remaining lines are inliers. This difference in the number of valid hypotheses and their corresponding inliers, significantly increases the chance of the proposed RANSAC correctly classifying the observed lines in the presence of noise and outliers, and thereby reduces the number of required sample triplets (or similarly quadruplets for partially calibrated cameras) to ensure that at least one inlier set is drawn. In practice, we do not know the number of outliers (i.e., non cardinal direction line segments), and therefore, we adaptively determine the number of sample sets by comparing the number of inliers at each stage with the total number of detected line segments [48].

6.5.3 Hybric RANSAC-based Classification

One potential issue of concern with RANSAC is the selection of the appropriate inlier residual threshold. Considering that the initial hypotheses for the vanishing points and the focal length (if applicable) are generated using a minimal number of line segments, they are quite sensitive to the noise in the measurements of the line segments' moments. If we choose a tight threshold to determine the inliers with respect to a hypothesis, many of the line segments that are indeed along cardinal directions may be counted as outliers, requiring us to draw many sample sets until one is selected with sufficient inlier support. Conversely, if a loose threshold is employed, we may mistakenly enlarge the inlier support of a "winning" hypothesis by including non-cardinal line segments that will significantly throw off the final estimate of the vanishing points and the focal length (if applicable).[5]

To address this issue, we propose a slightly modified RANSAC method that employs a pair of loose/tight thresholds for detecting inliers along with a hysteresis mechanism to provide more accurate line classification. Specifically, for each vanishing point hypothesis obtained from a minimal set of lines, we classify line segments using the loose residual threshold. Then, we use the analytical method of Section 6.3.2 to estimate the *refined* vanishing-points hypotheses based on the potential inliers. In the next step, we reclassify the line segments and determine the support of the current hypothesis using the refined vanishing-point estimates, but this time using the tight residual threshold. We update the winning hypothesis with the current one, given that it has a larger inlier support, and continue to draw new random sample sets until a sufficient number of them is drawn [48]. Once the winner hypothesis is determined, we proceed similar to the original RANSAC-based classification, and use methods of Sections 6.3.2 and 6.3.3 to estimate the unknowns.

6.6 Experiments

In this section, we present the experimental evaluation of the proposed algorithms and their comparison to existing methods. The evaluations are performed on the 102 outdoor and indoor images of the the York Urban Database (YUDB) [35]. The YUDB also

[5] Note that this problem occurs with all applications of the RANSAC algorithm (e.g., line-fitting). In general, one may reduce the impact of the outliers that are mistakenly labeled as inlier by employing robustified least-squares methods [48, 128]. These robust methods, however, employ non-polynomial cost functions (e.g., Huber cost function [56]) that are not adaptable to analytical solvers proposed in this thesis.

provides manually extracted line segments for every image, each labeled as being along one of the three cardinal directions. In addition, the cardinal vanishing points of every image are provided in the dataset as the intersections of the parallel lines along the cardinal directions. In practice, however, manually extracted and labeled parallel line segments are rarely available. For this reason, in the majority of the experiments described in this section, line segments are automatically extracted using the Canny edge detector, followed by edge linking and line-segment fitting adopted from Tardif [121]. The line segments that are shorter than a pre-determined length are discarded and the rest are employed in the evaluations. Note that the same set of extracted line segments, which often include outliers not parallel to any cardinal direction, are provided to all algorithms.

Proposed Methods' Acronyms

The acronym we employ for describing each method consists of two parts, separated by a dash line: (i) a prefix, denoting whether YUDB lines or automatically detected lines were used and how they were classified in the latter case, and (ii) a suffix that indicates which algorithm was tested and what quantities where estimated.

List of prefixes, indicating what line segments were used and how they were classified:

- **G-**: The **G**enie-aided experiments use manually extracted and labeled line segments of YUDB. They are provided for assessing the achievable performance of the proposed vanishing point and focal length estimators, given perfect line classification. These evaluations help to identify the performance loss due to inevitable errors of the RANSAC-based line classification.

- **R-**: The **R** experiments use the automatically detected line segments described above, and employ the **R**ANSAC-based classification scheme of Section 6.5.1 to classify them. This method adaptively chooses the number of samples before declaring a winner [48] (typically 5-15 sample sets are sufficient).

- **hR-**: The **hR** experiments use the automatically detected line segments, but employ the **h**ybrid **R**ANSAC-based classification scheme of Section 6.5.3 to classify them. Similar to the standard RANSAC, the hybrid RANSAC also adaptively chooses the number of samples before determining the winner hypothesis (typically 5-15 sample sets are sufficient).

Table 6.1: The acronyms for the experimental evaluations of the proposed methods. Note that the results from **R-ALSx** and **hR-ALSx** are omitted, as they are only slightly inferior to those of **R-ALS** and **hR-ALS**, respectively.

	G-	**R-**	**hR-**
-ALS	**G-ALS**	**R-ALS**	**hR-ALS**
-ALSx	**G-ALSx**	*R-ALSx*	*hR-ALSx*
-PCal	**G-PCal**	**R-PCal**	**hR-PCal**

List of suffixes, indicating what quantities were estimated and how:

- **-ALS**: Stands for **A**nalytical **L**east **S**quares and represents the case of a fully calibrated camera whose vanishing points are estimated by analytical minimization of (6.8) using the method described in Section 6.3.2.

- **-ALSx**: Similar to **-ALS**, but instead analytically minimizes the rela**x**ed cost function of (6.14) using the method described in Section 6.3.2, hence the additional **x** appended to the acronym. The camera is assumed to be fully calibrated in **-ALSx**.

- **-PCal**: The experiments that assume the camera is **P**artially **Cal**ibrated, and simultaneously estimate the focal length and the cardinal vanishing points. This is achieved by minimizing the cost function in (6.24) using the method described in Section 6.3.3.

For example, **hR-PCal** is the experiment that uses hybrid RANSAC-based classification and the line segments that were automatically extracted to estimate the focal length and cardinal vanishing points of the camera, by minimizing the cost function in (6.24). The complete list of the experimental evaluations of the proposed methods is provided in Table 6.1. Among all possible evaluations, we have omitted the results of **R-ALSx** and **hR-ALSx**, as they are only slightly inferior to those of **R-ALS** and **hR-ALS**.

Benchmarks' and Existing Methods' Acronyms

In addition to the experimental evaluation of the proposed methods, the following benchmarks and existing methods are evaluated:

- **GT**: Standing for **G**round **T**ruth, the vanishing point estimates labeled by **GT** are obtained by iterative minimization of the cost function (6.8) using the Gauss-Newton method [63], and initialized by the values provided in YUDB. Note that the real ground-truth values for the vanishing points are unknown and **GT** merely indicates the best achievable estimates for them given the manually extracted and labeled line segments of YUDB. The ground-truth value for the focal length of the camera is provided by YUDB as 675 pixels.

- **EM**: The **E**xpectation-**M**aximization method proposed in [68], and implemented by Hoiem *et al.* [54] for estimating the vanishing points (and the focal length in the case of a partially calibrated camera) from the automatically extracted line segments. The iterations are started heuristically with detection of line segments that are fairly parallel in the image space [67].

- **JPT**: The analytical method proposed and implemented by Tardif [121] for estimating the vanishing points (and the focal length in the case of a partially calibrated camera) from the automatically extracted line segments.

In all the aforementioned methods, if we do not estimate the focal length of the camera, we use the intrinsic camera calibration parameters provided by the YUDB. If we denote the estimated orientation of the camera by the **GT** method and any algorithm **X** as $\mathbf{C}_{GT}$ and $\mathbf{C}_X$, respectively, we measure the error in the estimated orientation with the norm of the tilt-angle error $||\tilde{\mathbf{s}}||$ corresponding to $\mathbf{C}(\tilde{\mathbf{s}}) = \mathbf{C}_X^T \mathbf{C}_{GT}$ [see (6.2)].

Experimental Evaluations for a Calibrated Camera

In Fig. 6.2, the tilt-angle errors of the calibrated camera's orientation estimates compared to **GT** are depicted for various evaluated algorithms. We first note that **G-ALS** achieves exactly the same results as those of the **GT**. This is not surprising as **G-ALS** uses the same manually extracted and labeled line segments as **GT**, and therefore it is guaranteed to find the globally optimal estimates for the camera's orientation and hence the vanishing points.

The **G-ALS** superior performance is followed closely by the performance of **G-ALSx** which minimizes a relaxed cost function to estimate the vanishing points (see Section 6.3.2). It can be observed that, provided with correct line extractions and classifications, the **G-ALSx** error is less than 1.5° for 98% of the images. This results confirms suitability of the relaxed minimization method, if computational resources are limited.

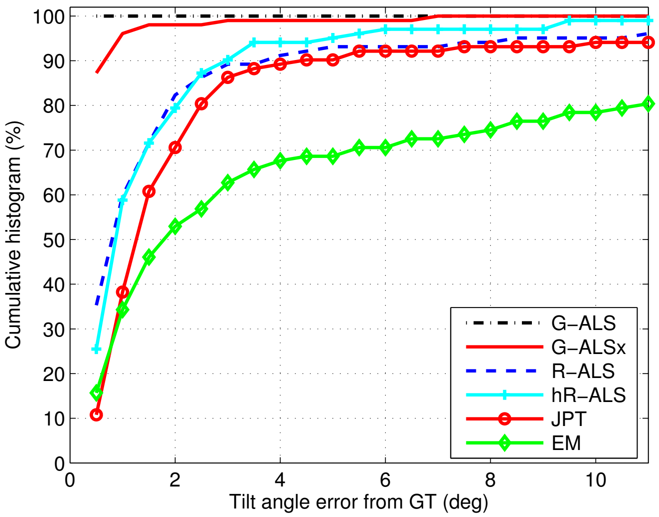


Figure 6.2: The cumulative histogram of the tilt angle error in the orientation of the fully calibrated camera as estimated by various evaluated algorithms.

The rest of the results shown in Fig. 6.2 are from algorithms that do not have access to the manually extracted and labeled line segments of YUDB, and instead rely on automatically extracted line segments. It is important to note that the same set of line segments were provided to all of these algorithms. The best performing method in this group is our hybrid RANSAC-based classification, followed by analytical least squares, namely **hR-ALS**, which is able to estimate the orientation of the camera in 94% of the cases with accuracy of 3.5° or better. The performance of **hR-ALS** is closely replicated by **R-ALS** in the lower 90th percentile. However, in the last 10th percentile, **R-ALS**'s performance is clearly inferior to that of **hR-ALS**. This indicates that in about 90% of the cases, the standard RANSAC algorithm is able to find the correct classification, as accurately as the hybrid RANSAC. However, in about 10% of the cases, RANSAC-based classification has too few inliers or has mistakenly labeled ourlier line segments as cardinal, leading to an inaccurate estimation of the vanishing points and thus the camera orientation. On the other hand, the **hR-ALS** which employs a hysteresis mechanism to detect inliers and outliers, achieves a higher accuracy even in the last 10th percentile.

The best competing method is **JPT** that closely follows **R-ALS** in the upper 10th percentile. In the lower percentiles, however, **JPT** is slightly worse than both **R-ALS** and **hR-ALS**. For example, **JPT** estimates the orientation with 2° or less error in only 70% of the cases, while **R-ALS** and **hR-ALS** estimate the orientation with such accuracy in more than 80% of the images. This can be explained by the fact that **JPT** in effect minimizes a different (relaxed) cost function than the least squares. In addition,

JPT does not employ the fact that the vanishing points of interest are perpendicular to each other during their estimation. Instead, it finds various vanishing points in the image and merely reports the ones that are closest to being perpendicular to each other. This leads to suboptimal estimates for the cardinal vanishing points, in contrast to **R-ALS** and **hR-ALS** that enforce orthogonality of the vanishing points throughout the estimation process.

The widely used **EM** algorithm, is a distant second competing algorithm, that achieves 10° or better accuracy in about 80% of the cases. Due to the iterative nature of **EM** and in the absence of accurate initialization, it is commonly trapped in local minima far from **GT**. This problem is often amplified further when **EM** incorrectly labels outlier line segments as inliers, leading to increasingly inaccurate estimates of orientation and vanishing points in the upper 20th percentile.

In Figs. 6.5-6.6, a sample set of images in YUDB are provided. In these figures, the left-most columns show the original image and the automatically extracted line segments. The other three columns show the labeled line segments in blue, yellow, and green. The outliers detected by each of the algorithms are marked red. Additionally for each of the algorithms, a clean version of the image with only labeled-as-cardinal line segments extended through vanishing points are presented, to help visually identify the location of the estimated vanishing point. The results of **R-ALS** are omitted, as they were virtually indistinguishable from those of **hR-ALS** for these images.

In Fig. 6.5, three examples are shown where all methods estimate the vanishing points with reasonable accuracy. Although all of the algorithms fail to identify some of the truly cardinal line segments, this issue is more pronounced with **EM** which always misses some of the line segments that were correctly labeled as cardinal by the two other algorithms.

On the other hand, Fig. 6.6 illustrates three cases where one or more of the algorithms fail to correctly detect the vanishing points. The **EM** algorithm fails in all three examples. In particular, **EM** fails in the image of row (a), where the two other algorithms succeed in accurately labeling the lines segments and estimating the vanishing points. The reason in this case is that **EM** is an iterative approach and requires an accurate initial estimate to succeed. Obtaining such an estimate, however, for vanishing points that fall in the center of an image is very challenging, since the employed heuristic-based initialization requires detection and grouping of the line segments with similar orientation in the image [68]. As evident from this image, the parallel line segments that have their vanishing point in the middle of the image appear with widely

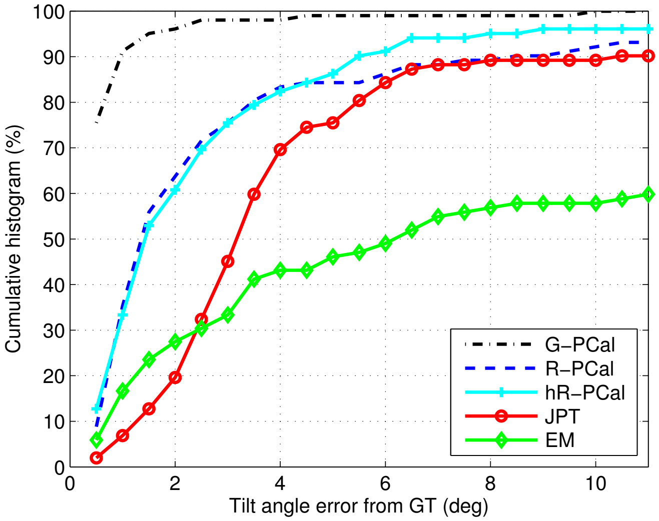


Figure 6.3: The cumulative histogram of the tilt angle error for the estimated orientation of the partially calibrated camera using various evaluated algorithms.

varying orientations in the image, which lead to failure of the heuristics for grouping them.[6]

Another notable example of failure in this set is the third image (the row denoted by (c) in Fig. 6.6) where there is a significant number of parallel lines on the staircase handrails which are not aligned with any of the cardinal directions. Even though in this case the **hR-ALS** estimates the vanishing points fairly accurately, it miss-classifies some of the handrail's line segments as being along one of the cardinal directions. Similarly, both **JPT** and **EM** mistakenly classify the handrails as being parallel to the cardinal directions and fail in labeling the actual cardinal line segments and consequently in estimating the camera's orientation. One interesting observation is that even in failure examples, almost all algorithms label two of the cardinal directions fairly reasonably, although due to incorrectly classifying the outlier lines as being along the third cardinal direction, they often end up reporting camera orientations with tens of degrees of error.

Experimental Evaluations for Partially Calibrated Camera

Figure 6.3 shows the performance of vanishing point estimation for the case of a partially calibrated camera. In this case, the focal length of the camera is assumed to be unknown, while the other intrinsic calibration parameters of the camera are set to the values

[6] Initialization via Hough transform is similarly challenging. In addition to being slow compared to heuristic methods that search for semi-aligned line segments in the image space, the Hough transform often gets numerous false hits in the middle of image, where many outlier line segments happen to cross each other.

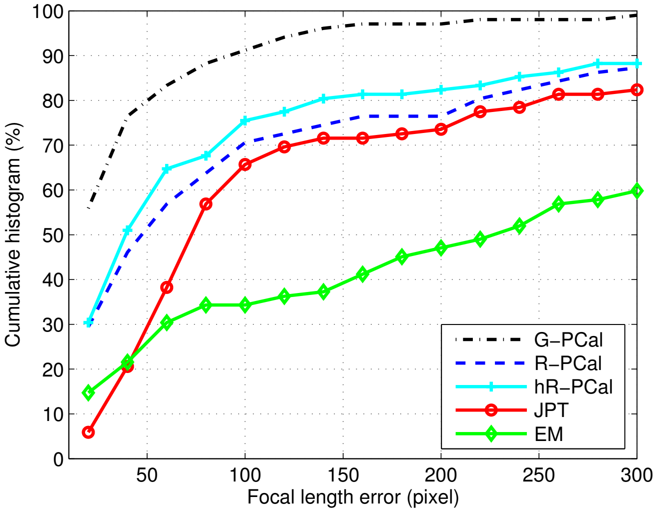


Figure 6.4: The cumulative histogram of the error in the estimated focal length of the partially calibrated camera, following the line classification using various evaluated algorithms.

provided by YUDB. Similar to the case of a calibrated camera, **EM**, **JPT**, **R-PCal**, and **hR-PCal** use the line segments obtained from the Canny's edge detector followed by edge-linking, line fitting, and removal of short segments. The **G-PCal**, however, uses the line segments and their classifications provided in YUDB and acts as a benchmark on the achievable performance of the analytical least-squares methods when the line classifications are known. It can be observed that in this case, the orientation of the camera is estimated with 2.5° or better accuracy in 98% of the images.

Similar to the case of a calibrated camera, **hR-PCal** achieves the highest accuracy, closely followed by **R-PCal** in the lower 85th percentile. In the last 15th percentile, however, **R-PCal** clearly falls behind **hR-PCal**. This difference is due to **hR-PCal**'s ability to avoid incorrectly labeling outlier lines by using a hysteresis mechanism for inlier detection (see Section 6.5.3). The **JPT** performance matches that of **R-PCal** in the last 15th percentile, but is clearly inferior in the lower 80th percentile. One explanation is that since **JPT** essentially minimizes a cost function other than least-squares, it does not even try to reach the **GT** estimates that are approximately optimal in the least-squares sense. Finally, **EM** estimates the camera orientation in less than 60% of the images with 10° or better accuracy. We suspect that this inferior performance is due to the iterative nature of **EM** and its sensitivity to initialization.

In Fig. 6.4, the histogram of the error in the focal length estimates for the evaluated methods is shown. The performance of the **G-PCal** which uses manually extracted and labeled line segments of YUDB may appear worse than expected. Specifically, **G-PCal** has estimated the focal length with less than 50 pixels of error in only 80% of the images. This result can be explained by considering that the ground-truth value for the focal length provided by YUDB is obtained by combining information from 10 images [35]. But what **G-PCal** approximates is the best achievable estimate of the focal length given the line segments of *only one* image.

Among the algorithms that use automatically extracted line segments, **hR-PCal** is slightly better than **R-PCal**, as similar to the calibrated camera case, the hysteresis mechanism helps to detect more inlier line segments without including too many outliers. The **JPT** performance is slightly worse that **R-PCal**. The main explanation for this is that **JPT**, in contrast to **R-PCal** and **hR-PCal**, first detects the orthogonal vanishing points assuming an initial estimate (575 pixels) for the focal length and then estimates the focal length in the second step. Therefore, if the orthogonal vanishing points are not correctly detected due to the error in the initial estimate for the focal length, the final estimate for the focal length may significantly deviate from the ground truth. This behavior is in contrast to **R-PCal** and **hR-PCal** that do not need any initial value for the focal length of the camera. Finally, **EM**, performs considerably worse due to its sensitivity to initialization, and is only able to estimate the focal length with less than 200 pixels of error in less than half of the images.

In Fig. 6.7, representative results of various evaluations are provided. Similar to the case of a calibrated camera, all algorithms miss several of the cardinal line segments. However, **EM** misses more line segments compared to others. For example, in row (c) of Fig. 6.7, it can be seen that several cardinal line segments along the staircase are labeled as outliers by **EM**.

Finally, Fig. 6.8 illustrates examples where one or more of the algorithms have failed. In particular, **EM** fails in row (a) due to inaccurate initialization while **JPT** fails in row (b) due to selecting the wrong group of parallel lines (yellow lines) as cardinal due to imprecise initial estimate of the focal length. In row (c) all three algorithms fail due to the strong group of parallel line segments along the handrails that do not correspond to any of the cardinal directions.

6.7 Summary

In this chapter, we presented a unified framework for analytically estimating the orthogonal vanishing points of a calibrated or partially calibrated camera (i.e., with unknown focal length), in a Manhattan world. Specifically, we studied both the minimal and optimal (in the least-squares sense) solvers, and employed the multiplication matrix to solve the multivariate polynomial systems resulting from either the minimal measurement constraints, or the optimality conditions of the corresponding constrained least-squares problem. For the case of the optimal solver, the solutions to the optimality conditions constitute the critical points, amongst which, the ones that minimize the cost function are the globally optimal estimates of the orthogonal vanishing points and the focal length. Additionally, we introduced a robust and efficient RANSAC-based line classifier that employs the minimal solver to generate hypotheses for all three orthogonal points (and the focal length) from triplets (or quadruplets) of line observations. Finally, we presented experimental validation of the proposed method on the existing test datasets to demonstrate its suitability for practical applications.

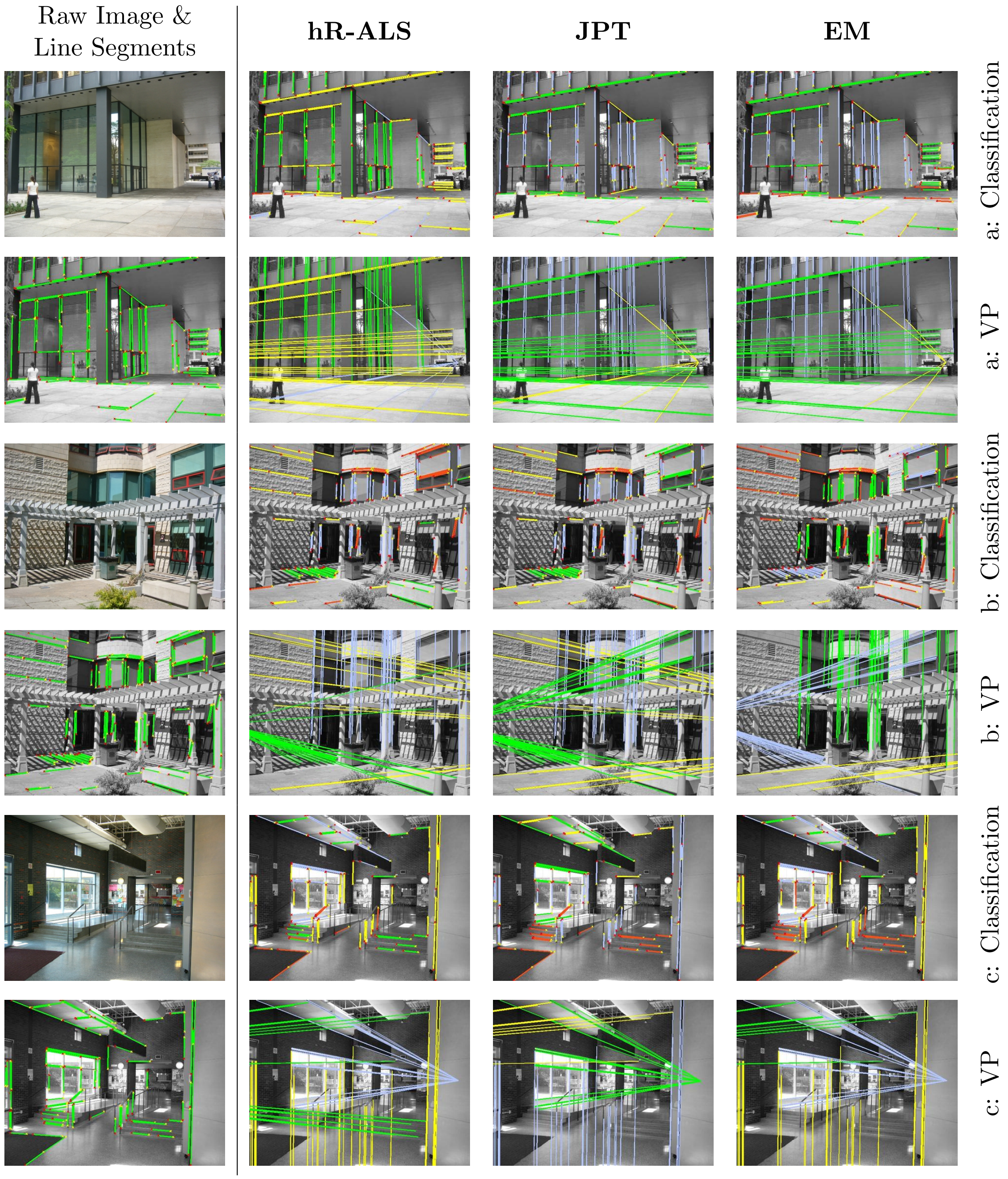


Figure 6.5: Successful vanishing point recovery in a calibrated camera: Examples of images where all the competing algorithms result in reasonable estimates of the orthogonal vanishing points. The left-most column shows the original images and the automatically extracted line segments. The other three columns show the line classification and orthogonal vanishing points as estimated by each algorithm. The results of **R-ALS** are not shown as they are very similar to **hR-ALS** in the selected examples.

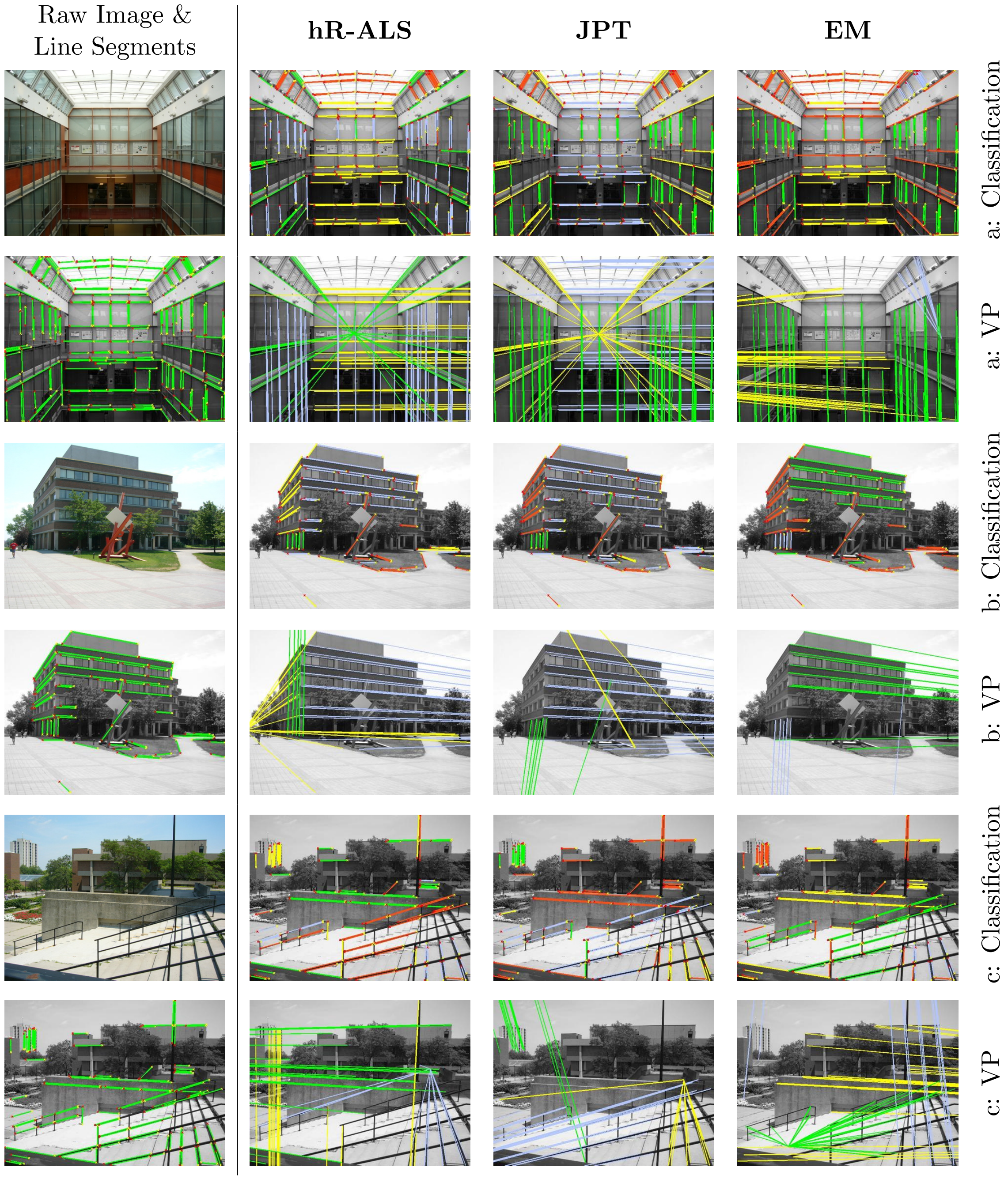


Figure 6.6: Failed estimation of vanishing points in a calibrated camera: Examples of images where one or more of the algorithms fail to reasonably estimate the orthogonal vanishing points. The left-most column shows the original images and the automatically extracted line segments. The other three columns show the line classification and orthogonal vanishing points as estimated by each algorithm. The results of **R-ALS** are not shown as they are very similar to **hR-ALS** in the selected examples.

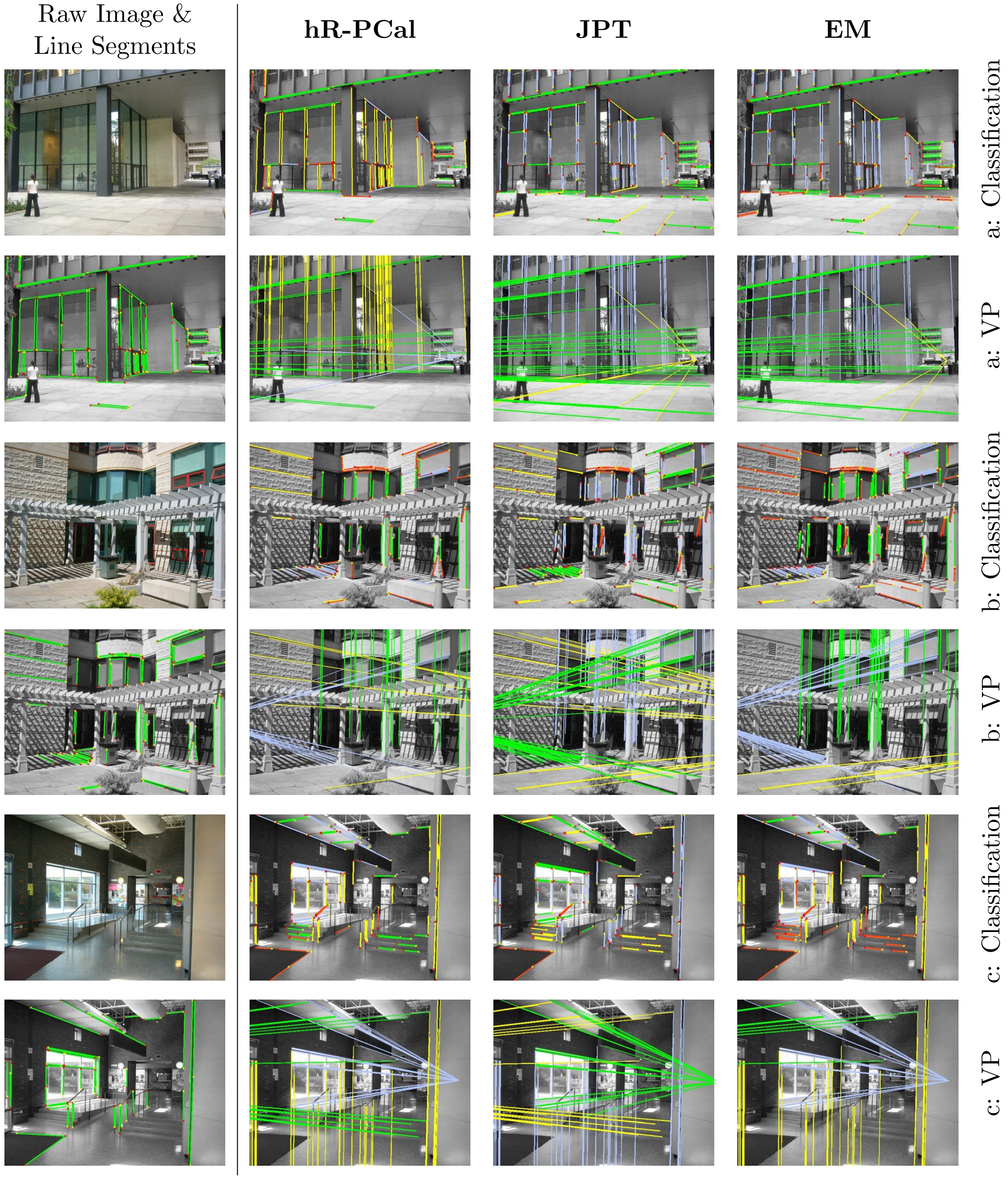


Figure 6.7: Successful estimation of vanishing points and focal length in a partially calibrated camera: Examples of images where all the competing algorithms result in reasonable estimates of the orthogonal vanishing points and focal length. The left-most column shows the original images and the automatically extracted line segments. The other three columns show the line classification and orthogonal vanishing points as estimated by each algorithm. The results of **R-PCal** are not shown as they are very similar to **hR-PCal** in the selected examples.

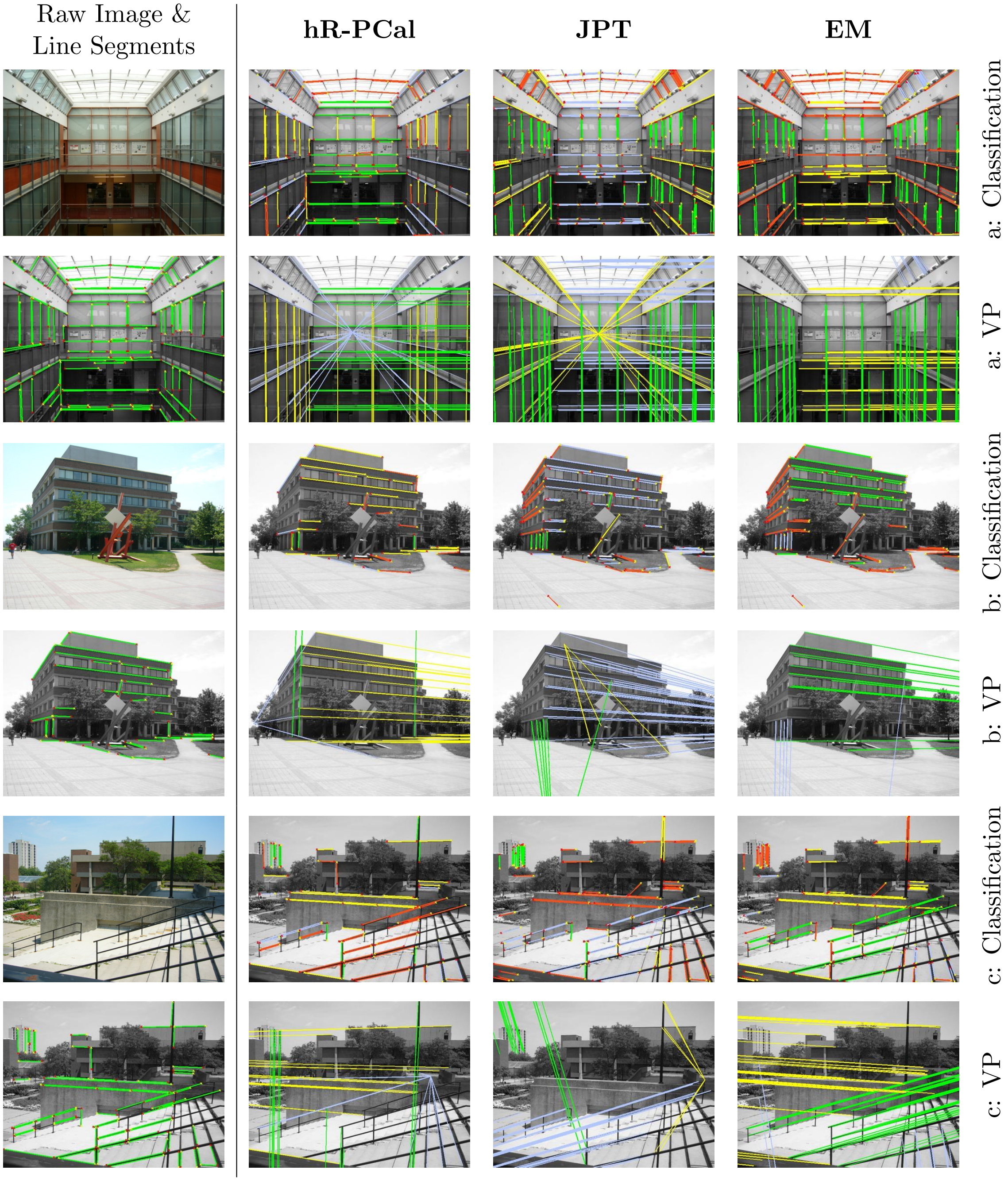


Figure 6.8: Failed estimation of vanishing points or focal length in a partially calibrated camera: Examples of images where one or more of the algorithms fail to reasonably estimate the orthogonal vanishing points or the focal length. The left-most column shows the original images and the automatically extracted line segments. The other three columns show the line classification and orthogonal vanishing points as estimated by each algorithm. The results of **R-PCal** are not shown as they are very similar to **hR-PCal** in the selected examples.

Chapter 7

Conclusion

7.1 Summary of Contributions

The work presented in this thesis focused on introducing new methods for calibrating some of the sensors that are commonly used in robotics and computer vision, and addressing challenging issues regarding the *observability* and *optimality* of the corresponding systems and estimators. The key contributions of this work are summarized in the following:

Gyroscope-Odometer Calibration:

Chapter 2 provided a novel Lie-derivative-based observability analysis for the gyroscope-odometer calibration system, and showed that as long as the rotational velocity of the robot that carries the sensors is not deterministically zero, the alignment between the two sensors can be estimated based on their own measurements. Subsequently, a novel method for performing the calibration, while taking the difference in the sensors' sampling mechanisms into account, was presented. Gyroscope-odometer sensor pairs have the potential to be widely used on ground vehicles and mobile robots, not only for decreasing the rate of localization uncertainty growth, but also for detecting single-sensor failures. Moreover, unlike other sensor combinations that include an exteroceptive sensor (e.g., a camera or a laser scanner), the gyroscope-odometers' performance does not depend on the lighting conditions or the presence of easy-to-detect-and-track features in the surrounding environment. We expect that the cheap price of gyroscope-odometer sensor pairs will soon allow their commercial usage on all vehicles and bring autonomy (e.g., collision prediction and avoidance) to the roads without increasing the financial burden on the consumers.

IMU-Camera Calibration:

Chapter 3 presented a novel EKF-based method for the IMU-camera calibration that does not require any special hardware except a calibration pattern (i.e., a checker board). A Lie-derivative-based analytical proof for the observability of this systems was provided, showing that when at least four known landmarks (e.g., corners of squares on the checker board) are observed, rotating the IMU-camera rig around two different axes is sufficient for estimating the unknown calibration parameters. A practical aspect of this proof was that no translational motion is required in order to calibrate any IMU-camera pair. IMU-camera pairs are one of the most widely available sensor combinations, found on cell phones, gaming consoles, and cars. By providing a method for fast and relatively easy calibration of these sensor pairs, this work sets the foundation for optimally exploiting them in precise localization and mapping applications (e.g., navigation-assistant devices for the visually impaired), where even small uncertainties can pose serious hazards (e.g., missing steps or obstacles) to their users.

3D LIDAR-Camera Calibration:

Chapter 4 provided an observability analysis for the 3D LIDAR-camera calibration system, showing that the observations of a calibration pattern (i.e., a checker board) at three configurations are sufficient to estimate the calibration parameters. To estimate the hundreds of parameters of the 3D LIDAR-camera calibration system, prior work had focused on iterative estimators (e.g., nonlinear least-squares method), whose accuracy depends on their initialization. This work presented the first method ever for providing an accurate initial estimate for the calibration parameters, by dividing the optimization problem into smaller ones, and solving each of them separately using the Analytical Nonlinear Least-Squares (ANLS) method. To this end, each individual problem was formulated as a polynomial minimization, whose optimality conditions form a system of multivariate polynomial equations. The solutions to this polynomial system comprise the set of critical points, among which, the one that minimizes the cost function is the guaranteed global minimizer. The provided initialization method is in particular important, as several of the calibration parameters (such as the offset and scale of the laser beams) cannot be directly measured from the technical drawings.

3D LIDARs, along with cameras and other sensors, have been recently used for autonomous navigation of vehicles (e.g., on the Google car and in the DARPA Urban Challenge). We expect that the presented calibration procedure will allow to optimally fuse measurements from these two sensors and open up new opportunities for using them in a wide variety of applications (e.g., photorealistic 3D reconstruction, surveillance, remote facility monitoring, etc.). For instance, by constructing photorealistic 3D maps of indoor environments (i.e., where GPS signals are not reliable such as airports and shopping malls) and storing them on remote servers, we can enable users to accurately determine their location by taking one picture with their smart phones. In this way, personal navigation and location-based services can be provided to the consumers in the places where they were not available before.

Extrinsic Camera Calibration:

Chapters 5 and 6 presented novel methods for extrinsic calibration of intrinsically calibrated monocular cameras from observation of line segments. Assuming that the coordinates of the observed lines are known *a priori*, a polynomial least-squares cost function was derived and minimized using the ANLS method. The key contribution of this algorithm is its ability to find the guaranteed globally optimal (in a least-squares sense) estimates for the orientation of the camera without requiring initialization or iterations. Furthermore, assuming that the coordinates of the observed lines are known, we derived a linear least-squares solution for the position of the camera.

In Chapter 6, for urban environments where the majority of lines are along the three cardinal axes, we relaxed the requirements of knowing the line-segments' coordinates and complete intrinsic calibration of the camera. Specifically, we presented a novel RANSAC-based classifier to first partition the line-segment observations into three orthogonal groups. Subsequently, the ANLS method was used to estimate the camera's vanishing points and focal length and provide a finite number of hypotheses for the orientation of the camera with respect to its surrounding. By estimating the camera's focal length and orientation with respect to its surroundings, the presented algorithm acts as a local compass that works reliably in indoor environments where electromagnetic interference prevents usage of magnetometers. This will be an extremely valuable tool for many applications that depend on the orientation estimation to work properly. Augmented reality is an example where determining the orientation of the camera is essential in order to apply perspective distortion to the virtual objects that are superimposed on the image.

7.2 Future Research Directions

A direct extension of our work would be to include in the calibration procedure, the intrinsic sensor parameters that were excluded for the purpose of simplifying the problem. In particular, for the 3D LIDAR-camera calibration problem, it would be desirable to estimate the elevation angle of the individual laser beams. For this purpose, one would need to first extend the observability analysis to ensure that estimating the extra unknowns given only the sensors' own measurements is possible. If that is the case, one would need to investigate the possible methods for analytically obtaining an accurate initial estimate for the elevation angle from the sensor measurements. Additionally, in the presented work, when exploiting measurements from multiple sensors, we assumed that their clocks are already synchronized. An interesting future direction would be to investigate if the system's state vector can be augmented with the time-delay between the sensors' clocks, and whether the resulting system remains observable.

Although the observability analysis provides sufficient conditions on the control inputs (e.g., motion of the sensor pairs) for estimating the unknown calibration parameters, it does not provide any guidance on what control inputs to choose in order to achieve the highest possible accuracy from a given number of measurements. An exciting future research direction would be to formulate optimization problems for minimizing the uncertainty of the calibration parameters' estimates over the control inputs and obtain the optimal control strategies [141]. Moreover, when external hardware for precisely controlling the inputs to the system (e.g., a 2 d.o.f. turn table) are available, further research has to be conducted on how to utilize them so as to improve the performance of the calibration procedure.

Regarding optimality of sensor calibrations, the presented ANLS method has the potential to be used for many calibration problems in robotics and computer vision, due to the relative ease of converting geometric constraints into polynomial equations. One of our ongoing research efforts is to apply this technique to the 5-point algorithm [102] in order to obtain the guaranteed globally optimal estimate for the motion of the camera from two overlapping images.

A more general issue of concern that the future research must address is with regard to the computational complexity of the ANLS method. Specifically, applying the ANLS method to many practical problems in computer vision and robotics (e.g., 5-point algorithm), will involve solving polynomial systems of high degree in a relatively large number of unknowns. In order to make these algorithms suitable for real-time applications, we would need to investigate avenues for reducing their processing requirements. In particular, with the anticipated slowdown in the growth of computational power, we need to exploit the widespread availability of multi-core processors. This may be achieved, in part, by employing distributed implementations of the QR decomposition [12] required in the computation of the multiplication matrix. Furthermore, if we are required to perform RANSAC in order to reject ouliers or classify observations, we can distribute the processing of individual hypothesis among the available cores.

An alternative approach is to reduce the computational complexity of solving systems of polynomial equations through intelligent expansion of the polynomials to obtain the multiplication matrix. Currently, in the method that was presented in Section 4.4, all the polynomials are expanded up to a certain degree to ensure that the monomials which reside outside the normal set can be mapped back into the set. This blind form of expanding the polynomials, however, may result in adding many polynomials that do not contribute to such mapping due to their linear dependence on the existing (already expanded) polynomials. Recently, a method was suggested to avoid including such expanded polynomials for simple problems [96]. An efficient approach for performing this task, however, is not yet available and requires further research.

In order to guarantee the optimality of the estimated calibration parameters, in this work we assumed that outliers do not exist among the measurements. In a future research, one would need to investigate how to account for outliers, and perform robust and optimal estimation of the unknown calibration parameters. One possible lead is to leverage the recent advances in ℓ_1 minimization and robust sensing [64] to force the majority of the individual measurement residuals to be lower than an outlier threshold.

Finally, in this work we employed the minimal representation of unknown parameters (e.g., CGR for rotation) in order to ensure that the minimization problems include the smallest possible number of variables, and their total degrees are as low as possible. However, these minimal representations often include singular points, or they may need a wide range of real numbers (e.g., $[-\infty, +\infty]$ for CGR parameters) to represent the space of possible values, leading to occurrences of bad conditioned matrices in the algorithm. To address these issues, one may need to use nonminimal representations of the unknowns (e.g., unit quaternions instead of CGR parameters for rotation), which are often accompanied by one or more constraints (e.g., unit-norm constraint for unit quaternions). Subsequently, these constraints have to be enforced in the polynomial optimization problem in order to obtain a feasible solution. To this end, if we attempt to directly solve the optimality conditions (e.g., using the multiplication matrix) and find the critical points, we must employ the method of Lagrange multipliers, which may lead to a prohibitively large number of variables. An alternative approach would be to leverage recent advances in the Sum-of-Squares (S.o.S.) relaxations [100, 66] for polynomial optimization which allow us to directly enforce constraints, without explicitly forming the Lagrange multipliers.

References

[1] T. Abbas, M. Arif, and W. Ahmed, “Measurement and correction of systematic odometry errors caused by kinematics imperfections in mobile robots,” in *SICE-ICASE International Joint Conference*, Busan, Oct. 2006, pp. 2073–2078.

[2] D. G. Aguilera, J. G. Lahoz, and J. F. Codes, “A new method for vanishing points detection in 3D reconstruction from a single view,” in *Proc. ISPRS Commission V Workshop Virtual Reconst. Visual. Complex Archit.*, Mestre-Venice, Italy, Aug. 22–24 2005.

[3] A. Ansar and K. Daniilidis, “Linear pose estimation from points or lines,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 578–589, May 2003.

[4] M. Antone and S. Teller, “Automatic recovery of relative camera rotations for urban scenes,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Hilton Head, SC, Jun. 13–15 2000, pp. 282–289.

[5] G. Antonelli, F. Caccaale, F. Grossi, and A. Marino, “Simultaneous calibration of odometry and camera for a differential drive mobile robot,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, AK, May 3–7 2010, pp. 5417–5422.

[6] G. Antonelli and S. Chiaverini, “Linear estimation of the physical odometric parameters for differential-drive mobile robots,” *Auton. Robots*, vol. 23, no. 1, pp. 59–68, Jul. 2007.

[7] L. Armesto and J. Tornero, “Robust and efficient mobile robot self-localization using laser scanner and geometrical maps,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 3080 – 3085.

[8] W. Auzinger and H. J. Stetter, “An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations,” in *Proc. Int. Conf. on Numer. Math.*, Singapore, 1988, pp. 11–30.

[9] A. Azarbayejani and A. Pentland, “Recursive estimation of motion, structure, and focal length,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 6, pp. 562–575, Jun. 1995.

[10] S. T. Barnard, “Interpreting perspective images,” *Artif. Intell.*, vol. 21, no. 4, pp. 435–462, Nov. 1983.

[11] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Scientific, 2004.

[12] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation:Numerical Methods.* Prentice Hall, 1989.

[13] P. Bonnifait and G. Garcia, “Design and experimental validation of an odometric and goniometric localization system for outdoor robot vehicles,” *IEEE Trans. Robot. Autom.*, vol. 14, no. 4, pp. 541–548, Aug. 1998.

[14] J. Borenstein and L. Feng, “Measurement and correction of systematic odometry errors in mobile robots,” *IEEE Trans. Robot. Autom.*, vol. 12, no. 6, pp. 869–880, Dec. 1996.

[15] J.-Y. Bouguet, “Camera calibration toolbox for matlab,” 2006. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calibdoc/

[16] W. L. Brogan, *Modern Control Theory.* Prentice Hall, 1990.

[17] J. B. Burns, A. R. Hanson, and E. M. Riseman, “Extracting straight lines,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 4, pp. 425 – 455, Jul. 1986.

[18] L. Busé, I. Emiris, B. Mourrain, O. Ruatta, and P. Trébuchet, “Multires, a Maple package for multivariate resolution problems,” 2002. [Online]. Available: http://www-sop.inria.fr/galaad/logiciels/multires

[19] M. Byröd, K. Josephson, and K. Åström, “Improving numerical accuracy of Gröbner basis polynomial equation solvers,” in *Proc. Int. Conf. Comput. Vision*, Rio de Janeiro, Brazil, Oct. 14–21 2007, pp. 1–8.

[20] ——, “A column-pivoting based strategy for monomial ordering in numerical Gröbner basis calculations,” in *Proc. Euro. Conf. Comput. Vision*, Marseille, France, Oct. 12–18 2008, pp. 130–143.

[21] M. Byröd, Z. Kukelova, K. Josephson, T. Pajdla, and K. Åström, “Fast and robust numerical solutions to minimal problems for cameras with radial distortion,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, Jun. 23–28, 2008, pp. 1–8.

[22] J. Canny, “A computational approach to edge detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, 1986.

[23] A. Censi, L. Marchionni, and G. Oriolo, “Simultaneous maximum-likelihood calibration of odometry and sensor parameters,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May 19–23 2008, pp. 2098–2103.

[24] Y. L. Chang and J. K. Aggarwal, “Calibrating a mobile camera’s extrinsic parameters with respect to its platform,” in *Proc. IEEE Int. Symp. Intell. Control*, Aug. 1991, pp. 443–448.

[25] A. B. Chatfield, *Fundamentals of High Accuracy Inertial Navigation.* Reston, VA: American Institute of Aeronautics and Astronautics, Inc., 1997.

[26] H. H. Chen, "Pose determination from line-to-plane correspondences: existence condition and closed-form solutions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 6, pp. 530–541, Jun. 1991.

[27] J. Chou and M. Kamel, "Finding the position and orientation of a sensor on a robot manipulator using quaternions," *Int. J. Robot. Res.*, vol. 10, no. 3, pp. 240–254, Jun. 1991.

[28] H. Chung, L. Ojeda, and J. Borenstein, "Accurate mobile robot dead-reckoning with a precision-calibrated fiber-optic gyroscope," *IEEE Trans. Robot. Autom.*, vol. 17, no. 1, pp. 80–84, Feb. 2001.

[29] J. M. Coughlan and A. L. Yuille, "Manhattan world: Orientation and outlier detection by bayesian inference," *Neural Comput.*, vol. 15, no. 5, pp. 1063–1088, May 2003.

[30] D. Cox, J. Little, and D. O'Shea, *Using Algebraic Geometry.* Springer, 2004.

[31] K. Daniilidis, "Hand-eye calibration using dual quaternions," *Int. J. Robot. Res.*, vol. 18, no. 3, pp. 286–298, Mar. 1999.

[32] P. David, D. DeMenthon, R. Duraiswami, and H. Samet, "Simultaneous pose and correspondence determination using line features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Madison, WI, Jun. 18–20, 2003, pp. 424–431.

[33] T. A. Davis, *Direct Methods for Sparse Linear Systems.* Philadelphia: SIAM, 2006.

[34] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications.* Springer-Verlag, 2008.

[35] P. Denis, J. Elder, and F. Estrada, "Efficient edge-based methods for estimating Manhattan frames in urban imagery," in *Proc. Euro. Conf. Comput. Vision*, Marseille, France, Oct. 12-18 2008, pp. 197–210.

[36] J. Dias, M. Vinzce, P. Corke, and J. Lobo, "Special issue: 2nd workshop on integration of vision and inertial sensors," in *Int. J. Robot. Res.* Sage Publications, Jun. 2007, vol. 26, no. 6.

[37] G. Dissanayake, S. Sukkarieh, E. Nebot, and H. Durrant-Whyte, "The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications," *IEEE Trans. Robot. Autom.*, vol. 17, no. 5, pp. 731–747, Oct. 2001.

[38] F. Dornaika and C. Garcia, "Pose estimation using point and line correspondences," *Real-Time Imaging*, vol. 5, no. 3, pp. 215–230, Jun. 1999.

[39] S. A. Dyer, *Wiley Survey of Instrumentation and Measurement.* Wiley-IEEE Press, 2001.

[40] A. Etemadi, "Robust segmentation of edge data," in *Proc. Intl Conf. Image Processing and Its Applications*, Apr. 1992, pp. 311–314.

[41] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Comm. of the ACM*, vol. 24, p. 381395, June 1981.

[42] W. Förstner, "Optimal vanishing point detection and rotation estimation of single images from a legoland scene," in *Proc. ISPRS Commission III Symp. Photogramm. Comput. Vis. Image Anal.*, Sep. 2010, pp. 157–162.

[43] E. M. Foxlin, "Generalized architecture for simultaneous localization, autocalibration, and map-building," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Lauzanne, Switzerland, Sep. 30 – Oct. 4 2002, pp. 527–533.

[44] C. Glennie and D. D. Lichti, "Static calibration and analysis of the velodyne hdl-64e s2 for high accuracy mobile scanning," *Remote Sens.*, vol. 2, no. 6, pp. 1610–1624, 2010.

[45] G. H. Golub and C. F. van Loan, *Matrix Computations.* Baltimore: Johns Hopkins University Press, 1996.

[46] C. X. Guo, F. M. Mirzaei, and S. I. Roumeliotis, "An analytical least-squares solution to the odometer-camera extrinsic calibration problem," in *Proc. IEEE Int. Conf. Robot. Autom. (to appear)*, St. Paul, MN, May 14-18 2012.

[47] R. Haralick, "Pose estimation from corresponding point data," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 19, no. 6, pp. 1426–1446, Nov.–Dec. 1989.

[48] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[49] M. Hebert, "Active and passive range sensing for robotics," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, San Francisco, CA, Apr. 24–28 2000, pp. 102–110.

[50] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 17–19 1997, pp. 1106–1112.

[51] R. Hermann and A. Krener, "Nonlinear controlability and observability," *IEEE Trans. Autom. Control*, vol. AC-22, no. 5, pp. 728–740, Oct. 1977.

[52] J. A. Hesch, F. M. Mirzaei, G. L. Mariottini, and S. I. Roumeliotis, "A laser-aided inertial navigation system (L-INS) for human localization in unknown indoor environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, AK, May 3–8, 2010, pp. 5376–5382.

[53] J. A. Hesch and S. I. Roumeliotis, "A direct least-squares (dls) solution for PnP," in *Proc. of the Int. Conf. on Computer Vision*, Barcelona, Spain, Nov. 6–13, 2011.

[54] D. Hoiem, A. Efros, and M. Hebert, "Geometric context from a single image," in *Proc. Int. Conf. Comput. Vision*, Beijing, China, 17–21 Oct. 2005.

[55] B. K. P. Horn, "Closed-form solution of the absolute orientation using quaternions," *J. Opt. Soc. Amer. A*, vol. 4, pp. 629–642, Apr. 1987.

[56] P. J. Huber, "Robust regression: Asymptotics, conjectures, and monte carlo," *Ann. Statist.*, vol. 1, pp. 799–821, 1973.

[57] *IEEE standard specification format guide and test procedure for linear, singleaxis, nongyroscopic accelerometers*, The Institute of Electrical and Electronics Engineers, New York, NY, 1999, IEEE Std 1293-1998.

[58] *IEEE standard specification format guide and test procedure for single-axis laser gyros*, The Institute of Electrical and Electronics Engineers, New York, NY, 2006, IEEE Std 647-2006.

[59] A. Isidori, *Nonlinear Control Systems*, 3rd ed. London: Springer–Verlag, 1995.

[60] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, ser. Mathematics in Science and Engineering. Academic Press, 1970, vol. 64.

[61] A. E. Johnson, J. F. Montgomery, and L. H. Matthies, "Vision guided landing of an autonomous helicopter in hazardous terrain," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, Apr. 18–22 2005, pp. 3966–3971.

[62] A. E. Johnson, R. Willson, J. Goguen, J. Alexander, and D. Meller, "Field testing of the mars exploration rovers descent image motion estimation system," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, Apr. 18–22 2005, pp. 4463–4469.

[63] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory.* Prentice Hall, 1993.

[64] V. Kekatos and G. B. Giannakis, "From sparse signals to sparse residuals for robust sensing," *IEEE Trans. Signal Process.*, vol. 59, no. 7, pp. 3355–3368, Jul. 2011.

[65] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *Int. J. Robot. Res.*, vol. 30, no. 1, pp. 56–79, 2011.

[66] S. Kim and M. Kojima, "Solving polynomial least squares problems via semidefinite programming relaxations," *J. of Global Optim.*, vol. 46, pp. 1–23, 2010.

[67] J. Košecká and W. Zhang, "Efficient computation of vanishing points," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, 2002, pp. 223–228.

[68] ——, "Video compass," in *Proc. Euro. Conf. Comput. Vision*, Copenhagen, Denmark, May 27 - Jun. 2 2002, pp. 29–32.

[69] R. Kumar and A. R. Hanson, "Robust methods for estimating pose and a sensitivity analysis," *Comput. Vis. Graph. Image Process.*, vol. 60, no. 3, pp. 313–342, Nov. 1994.

[70] P. Lamon and R. Siegwart, "Inertial and 3D-odometry fusion in rough terrain – towards real 3D navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 2, Sendai, Japan, Sep. 28 – Oct. 2 2004, pp. 1716 – 1721.

[71] P. Lang and A. Pinz, "Callibration of hybrid vision/inertial tracking systems," in *InerVis*, Barcelona, Spain, Apr. 2005.

[72] K. W. Lee, W. S. Wijesoma, and J. I. Guzman, "On the observability and observability analysis of SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Beijing, China, Oct. 9–15 2006, pp. 3569–3574.

[73] E. J. Lefferts, F. L. Markley, and M. D. Shuster, "Kalman filtering for spacecraft attitude estimation," *J. Guid. Control Dynam.*, vol. 5, no. 5, pp. 417–429, Sep.–Oct. 1982.

[74] Y. Liu, T. S. Huang, and O. D. Faugeras, "Determination of camera location from 2-d to 3-d line and point correspondences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 1, pp. 28–37, Jan. 1990.

[75] J. Lobo and J. Dias, "Relative pose calibration between visual and inertial sensors," in *InerVis*, Barcelona, Spain, Apr. 2005.

[76] ——, "Relative pose calibration between visual and inertial sensors," *Int. J. Robot. Res.*, vol. 26, no. 6, pp. 561–575, Jun. 2007.

[77] C.-P. Lu, G. D. Hager, and E. Mjolsness, "Fast and globally convergent pose estimation from video images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 6, pp. 610–622, 2000.

[78] E. Lutton, H. Maitre, and J. Lopez-Krahe, "Contribution to the determination of vanishing points using Hough transform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 4, pp. 430–438, Apr. 1994.

[79] S. Maeyama, N. Ishikawa, and S. Yuta, "Rule based filtering and fusion of odometry and gyroscope for a fail safe dead reckoning system of a mobile robot," in *Proc. IEEE/SICE/RSJ Int. Conf. on Multisens. Fusion Integr. Intell. Syst.*, Washington, DC, Dec. 8–11 1996, pp. 541–548.

[80] G. L. Mariottini, G. Pappas, D. Prattichizzo, and K. Daniilidis, "Vision-based localization of leader-follower formations," in *Proc. IEEE Conf. on Decision Control*, Seville, Spain, Dec. 12-15 2005, pp. 635–640.

[81] A. Martinelli, D. Scaramuzza, and R. Siegwart, "Automatic self-calibration of a vision system during robot motion," in *Proc. IEEE Int. Conf. Robot. Autom.*, Orlando, FL, May 15–19 2006, pp. 43–48.

[82] A. Martinelli and R. Siegwart, "Observability analysis for mobile robot localization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Edmonton, Canada, Aug. 2-6 2005, pp. 1471–1476.

[83] P. S. Maybeck, *Stochastic Models, Estimation and Control.* New York: Academic Press, 1979, vol. 1.

[84] ——, *Stochastic Models, Estimation and Control.* New York: Academic Press, 1979, vol. 2.

[85] M. Minimair, "MR: Macaulay resultant package for Maple," 2005. [Online]. Available: http://minimair.org/mr

[86] F. M. Mirzaei, D. G. Kottas, and S. I. Roumeliotis, "Analytical least-squares solution for 3d lidar-camera calibration," in *Proc. Int. Symp. Robot. Res.*, Flagstaff, AZ, Aug. 28 - Sep. 1 2011.

[87] F. M. Mirzaei and S. I. Roumeliotis, "Optimal estimation of vanishing points in a manhattan world," in *Proc. Int. Conf. Comput. Vision*, Barcelona, Spain, Nov. 2011.

[88] ——, "IMU-camera calibration: Bundle adjustment," Dept. of Computer Science & Engineering, University of Minnesota, Minneapolis, MN, Tech. Rep., Aug. 2007. [Online]. Available: http://mars.cs.umn.edu/tr/reports/Mirzaei07a.pdf

[89] ——, "IMU-camera calibration: Observability analysis," Dept. of Computer Science & Engineering, University of Minnesota, Minneapolis, MN, Tech. Rep., Jan. 2007. [Online]. Available: http://mars.cs.umn.edu/tr/reports/Mirzaei07.pdf

[90] ——, "A Kalman filter-based algorithm for IMU-camera calibration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Diego, CA, Oct. 29 – Nov. 2 2007, pp. 2427–2434.

[91] ——, "Globally optimal pose estimation from line-to-plane correspondences," University of Minnesota, Department of Computer Science, Tech. Rep., 2010.

[92] ——, "Globally optimal pose estimation from line correspondences," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 9–13 2011.

[93] ——, "A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1143–1156, Oct. 2008.

[94] A. I. Mourikis and S. I. Roumeliotis, "A dual-layer estimator architecture for long-term localization," in *Proc. of the Workshop on Visual Localization for Mobile Platforms*, Anchorage, AK, Jun. 2008, pp. 1–8.

[95] H. Myung, H.-K. Lee, K. Choi, S.-W. Bang, Y.-B. Lee, and S.-R. Kim, "Constrained kalman filter for mobile robot localization with gyroscope," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Beijing, China, Oct. 9-15 2006, pp. 442–447.

[96] O. Naroditsky and K. Daniilidis, “Optimizing polynomial solvers for minimal geometry problems,” in *Proc. Int. Conf. Comput. Vision*, Nov. 2011.

[97] O. Naroditsky, A. Patterson IV, and K. Daniilidis, “Automatic alignment of a camera with a line scan lidar system,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 9–13, 2011.

[98] N. Navab and O. Faugeras, “Monocular pose determination from lines: critical sets and maximum number of solutions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, New York City, NY, Jun. 1993, pp. 254 –260.

[99] V. Nguyen, A. Harati, A. Martinelli, R. Siegwart, and N. Tomatis, “Orthogonal SLAM: a step toward lightweight indoor autonomous navigation,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Beijing, China, Oct. 9–15, 2006, pp. 5007–5012.

[100] J. Nie and J. Demmel, “Sparse sos relaxations for minimizing functions that are summations of small polynomials,” *SIAM Journal On Optimization*, vol. 19, no. 4, pp. 1534–1558, 2008.

[101] H. Nijmeijer and A. van der Schaft, *Nonlinear Dynamical Control Systems.* New York: Springer–Verlag, 1990.

[102] D. Nister, “An efficient solution to the five-point relative pose problem,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 18–20 2003, pp. 195–202.

[103] G. Pandey, J. McBride, S. Savarese, and R. Eustice, “Extrinsic calibration of a 3D laser scanner and an omnidirectional camera,” in *Proc. IFAC Symp. on Intell. Auton. Veh.*, Lecce, Italy, Sep. 6–8, 2010.

[104] T. Pfister, K. L. Kriechbaum, S. I. Roumeliotis, and J. Burdick, “Weighted range sensor matching algorithms for mobile robot displacement estimation,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Washington D.C., May 11-15 2002, pp. 1667–1674.

[105] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C.* Cambridge: Cambridge University Press, 1992.

[106] L. Quan and Z.-D. Lan, “Linear n-point camera pose determination,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 8, pp. 774–780, 1999.

[107] G. Reid and L. Zhi, “Solving polynomial systems via symbolic-numeric reduction to geometric involutive form,” *J. Symb. Comput.*, vol. 44, no. 3, pp. 280 – 291, Mar. 2009.

[108] C. Rother, “A new approach to vanishing point detection in architectural environments,” *Image Vis. Comput.*, vol. 20, no. 9–10, pp. 647 – 655, 2002.

[109] A. Rudolph, “Quantification and estimation of differential odometry errors in mobile robotics with redundant sensor information,” *Int. J. Robot. Res.*, vol. 22, no. 2, pp. 117–128, Feb. 2003.

[110] W. J. Rugh, *Linear System Theory*, 2nd ed. Prentice Hall, 1996.

[111] S. Sastry, *Nonlinear Systems: Analysis, Stability, and Control.* New York: Springer–Verlag, 1999.

[112] G. G. Scandaroli, P. Morin, and G. Silveira, "A nonlinear observer approach for concurrent estimation of pose, imu bias and camera-to-imu rotation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2011, pp. 3335 –3341.

[113] D. Scaramuzza, A. Harati, and R. Siegwart, "Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Diego, USA, Oct. 29–Nov. 2, 2007, pp. 4164 –4169.

[114] G. Schindler and F. Dellaert, "Atlanta world: an expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Washington, DC, Jun. 27 – Jul. 2 2004, pp. 203–209.

[115] S. Se, D. Lowe, and J. Little, "Global localization using distinctive visual features," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Lausanne, Switzerland, Oct. 2002, pp. 226–231.

[116] J. Shufelt, "Performance evaluation and analysis of vanishing point detection techniques," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 3, pp. 282–288, Mar. 1999.

[117] M. D. Shuster, "A survey of attitude representations," *J. Astronaut. Sci.*, vol. 41, no. 4, pp. 439–517, Oct.–Dec. 1993.

[118] I. Stamos, L. Liu, C. Chen, G. Wolberg, G. Yu, and S. Zokai, "Integrating automated range registration with multiview geometry for the photorealistic modeling of large-scale scenes," *Int. J. Comput. Vision*, vol. 78, no. 2-3, pp. 237–260, Nov. 2008.

[119] H. Stewnius, C. Engels, and D. Nistr, "Recent developments on direct relative orientation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 60, no. 4, pp. 284–294, 2006.

[120] D. Sun and J. L. Crassidis, "Observability analysis of six-degree-of-freedom configuration determination using vector observations," *J. Guid. Control Dynam.*, vol. 25, no. 6, pp. 1149–1157, Nov. 2002.

[121] J.-P. Tardif, "Non-iterative approach for fast and accurate vanishing point detection," in *Proc. Int. Conf. Comput. Vision*, Kyoto, Japan, Sep. 29 – Oct. 2 2009, pp. 1250–1257.

[122] G. A. Thompson, "Inertial measurement unit calibration using full information maximum likelihood optimal filtering," Master's thesis, Massachusetts Institute of Technology. Dept. of Aeronautics and Astronautics., Sep. 2005.

[123] D. Titterton and J. Weston, Eds., *Strapdown Inertial Navigation Technology.* IEE, 2005.

[124] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D pose estimation," Dept. of Computer Science & Engineering, University of Minnesota, Minneapolis, MN, Tech. Rep., Mar. 2005.

[125] ——, "On the global optimum of planar, range-based robot-to-robot relative pose estimation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, AK, May 3 - 8 2010.

[126] N. Trawny, X. S. Zhou, K. X. Zhou, and S. I. Roumeliotis, "Interrobot transformations in 3-D," *IEEE Trans. Robot.*, vol. 26, no. 2, pp. 226 –243, Apr. 2010.

[127] N. Trawny, X. S. Zhou, and S. I. Roumeliotis, "3D relative pose estimation from six distances," in *Proc. Robot. Sci. Syst.*, Seattle, WA, Jun. 28– Jul. 1, 2009.

[128] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment - a modern synthesis," in *Vision Algorithms: Theory and Practice*, B. Triggs, A. Zisserman, and R. Szeliski, Eds. Springer–Verlag, 2000.

[129] R. Tsai, "An efficient and accurate camera calibration technique for 3d machine vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami Beach, FL, 1986, pp. 364–374.

[130] ——, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal of Robotics and Automation*, vol. 3, no. 4, pp. 323–344, Aug. 1987.

[131] R. Tsai and R. Lenz, "A new technique for fully autonomous and efficient 3d robotics hand/eye calibration," *IEEE Trans. Robot. Autom.*, vol. 5, no. 3, pp. 345–358, Jun. 1989.

[132] R. Unnikrishnan and M. Hebert, "Fast extrinsic calibration of a laser rangefinder to a camera," Carnegie Mellon University, Robotics Institute, Tech. Rep., July 2005.

[133] I. Velodyne Lidar, "Hdl-64e velodyne lidar," Available: http://velodynelidar.com/lidar/lidar.aspx.

[134] R. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 722 –732, Apr. 2010.

[135] H. Wildenauer and M. Vincze, "Vanishing point detection in complex man-made worlds," in *Proc. Int. Conf. Image Anal. Process.*, Modena, Italy, Sep. 2007, pp. 615 –622.

[136] C. Ye and J. Borenstein, "Characterization of a 2-D laser scanner for mobile robot obstacle negotiation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Washington D.C., May 11-15 2002, pp. 2512–2518.

[137] S. Yi, R. M. Haralick, and L. G. Shapiro, "Error propagation in machine vision," *Mach. Vision Appl.*, vol. 7, no. 2, pp. 93–114, Apr. 1994.

[138] X. Zezhong, L. Jilin, and X. Zhiyu, "Map building and localization using 2d range scanner," in *Proc. of the IEEE International Symposium of Computational Intelligence in Robotics and Automation*, Jul. 16-20 2003, pp. 848–853.

[139] Q. Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 3, Sep. 28–Oct. 2, 2004, pp. 2301–2306.

[140] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proc. Int. Conf. Comput. Vision*, vol. 1, Sep. 20–27, 1999, pp. 666–673.

[141] K. Zhou and S. Roumeliotis, "Optimal motion strategies for range-only constrained multi-sensor target tracking," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1168–1185, Oct. 2008.

[142] X. Zhou and S. Roumeliotis, "Robot-to-robot relative pose estimation from range measurements," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1379 –1393, Dec. 2008.

[143] ——, "Determining the robot-to-robot 3d relative pose using combinations of range and bearing measurements: 14 minimal problems and closed-form solutions to three of them," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 2983 –2990.

[144] ——, "Determining the robot-to-robot 3d relative pose using combinations of range and bearing measurements (part ii)," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 4736 –4743.

Appendix A

Bundle Adjustment for IMU-Camera Calibration

In order to compare the results of the proposed EKF algorithm for estimating the 6 d.o.f. IMU-camera transformation with the best achievable (off-line) estimates, we compute the batch least-squares estimate, also known as bundle adjustment [128]. For this purpose we minimize the following linearized cost function:

$$\begin{aligned}\mathcal{C}_M^i &= \frac{1}{2}\left(\bar{\mathbf{x}}_0 - \hat{\mathbf{x}}_0^i\right)^T \mathbf{P}_0^{-1}\left(\bar{\mathbf{x}}_0 - \hat{\mathbf{x}}_0^i\right) \\ &+ \frac{1}{2}\sum_{k=0}^{M-1}\left(\widetilde{\mathbf{z}}_k - \mathbf{H}_k^i\widetilde{\mathbf{x}}_k^i\right)^T \mathbf{R}_k^{-1}\left(\widetilde{\mathbf{z}}_k - \mathbf{H}_k^i\widetilde{\mathbf{x}}_k^i\right) \\ &+ \frac{1}{2}\sum_{k=0}^{M-1}\left(\widetilde{\mathbf{x}}_{k+1}^i - \boldsymbol{\Phi}_k^i\widetilde{\mathbf{x}}_k^i\right)^T \mathbf{Q}_k^{-1}(\widetilde{\mathbf{x}}_{k+1}^i - \boldsymbol{\Phi}_k^i\widetilde{\mathbf{x}}_k^i)\end{aligned} \tag{A.1}$$

where $\boldsymbol{\Phi}_k^i$ [see (3.17)], $\mathbf{H}_k^i$ [see (3.20)] and $\widetilde{\mathbf{z}}_k = \mathbf{r}_k$ [see (3.22)] are evaluated at $\hat{\mathbf{x}}_k^i = \hat{\mathbf{x}}_k^{i-1} + \widetilde{\mathbf{x}}_k^{i-1}$, the i-th iterate of the system's state-vector estimates at time-step k, $k = 0 \ldots M$ [see (3.2) and (3.14)]. Additionally, $\mathbf{R}_k$ represents the measurement noise covariance matrix (see Section 3.3.3) and $\mathbf{Q}_k$ is the discrete-time system noise covariance matrix [see (3.18)]. Furthermore, $\bar{\mathbf{x}}_0$ and $\mathbf{P}_0$ represent the initial state estimate and its covariance, respectively. Minimizing this cost function requires solving the following system of equations iteratively:

$$\mathbf{M}\underline{\widetilde{\mathbf{x}}}^i = \boldsymbol{\epsilon} \tag{A.2}$$

where $\mathbf{M}$ and $\boldsymbol{\epsilon}$ are computed as functions of $\hat{\mathbf{x}}_k^i$, $\mathbf{z}_k$, $\mathbf{\Phi}_k^i$, $\mathbf{H}_k^i$, $\mathbf{P}_0$, $\mathbf{R}_k$, $\mathbf{Q}_k$, and $\underline{\widetilde{\mathbf{x}}}^i = [\widetilde{\mathbf{x}}_0^{iT} \ \ldots \ \widetilde{\mathbf{x}}_N^{iT}]^T$ [88]. In our implementation, we have initialized $\hat{\mathbf{x}}_k^0, k = 0 \ldots M$ with the results from the EKF proposed in Chapter 3, and employed the sparse Cholesky factorization with symmetric approximate minimum degree permutation to solve (A.2) [33]. The resulting estimates are used as benchmarks in Section 3.5.

Appendix B

Bundle Adjustment for 3D LIDAR-Camera Calibration

One approach for enforcing the constraints in (4.28) is to use the method of Lagrange multipliers. It is possible, however, to re-parameterize the cost function and minimally express it over the optimization variables. In this way, the constraints in (4.28) will be automatically satisfied. Specifically, we consider the k-th *intrinsically corrected* point measured by the i-th laser scanner from the j-th configuration of the calibration plane as

$$
{}^{L}\mathbf{p}_{ijk} = \begin{bmatrix} \alpha_i(\rho_{ijk} + \rho_{oi}) \cos\phi_i \cos(\theta_{ijk} + \theta_{oi}) \\ \alpha_i(\rho_{ijk} + \rho_{oi}) \cos\phi_i \sin(\theta_{jik} + \theta_{oi}) \\ \alpha_i(\rho_{ijk} + \rho_{oi}) \sin\phi_i + h_i \end{bmatrix}. \tag{B.1}
$$

This relationship is obtained by substituting (4.1) in (4.2), and then transforming the result to the LIDAR's frame of reference $\{L\}$. Note that the intrinsic LIDAR parameters are already expressed in their minimal form; thus the constraints in (4.28) are redundant and can be removed. In particular, (4.14) is satisfied since h_i is added to the z component of the point's measurement and (4.13) is satisfied since θ_{oi} is added to the azimuth of the point's measurement. Also, we set $h_1 = \theta_{o1} = 0$, since we have assumed $\{L_1\} \equiv \{L\}$. Based on (B.1), we define the following *unconstrained* minimization problem:

$$
\min \sum_{i,j,k} \frac{\left({}^{C}\bar{\mathbf{n}}_j^T \, {}^{C}_{L}\mathbf{C} \, {}^{L}\mathbf{p}_{ijk} + {}^{C}\bar{\mathbf{n}}_j^T \, {}^{C}\mathbf{t}_L - d_j \right)^2}{\sigma^2_{\epsilon_{ijk}}} \tag{B.2}
$$

over ${}^{C}\mathbf{t}_{L},\ {}^{C}_{L}\mathbf{C},\ \alpha_i,\ \rho_{oi},\ \theta_{oi},\ h_i,\ i = 2, \ldots, K,\ \alpha_1$, and ρ_{o1}. Finally, we minimize this cost function using the Levenberg-Marquardt algorithm [105].