UNIVERSITÄT BONN

Inaugural-Dissertation
zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
der Landwirtschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn
Institut für Geodäsie und Geoinformation

Extrinsic Calibration and Ego-Motion Estimation for Mobile Multi-Sensor Systems

von
Kaihong Huang
aus
Guangdong, China

Bonn 2018

Referent:
Prof. Dr. Cyrill Stachniss, Friedrich-Wilhelms-Universität Bonn

Korreferent:
Prof. Dr. Wolfgang Förstner, Friedrich-Wilhelms-Universität Bonn

Tag der mündlichen Prüfung: 07.12.2018

Angefertigt mit Genehmigung der Landwirtschaftlichen Fakultät der Universität Bonn

Abstract

Autonomous robots and vehicles are often equipped with multiple sensors to perform vital tasks such as localization or mapping. The joint system of various sensors with different sensing modalities can often provide better localization or mapping results than individual sensor alone in terms of accuracy or completeness. However, to enable improved performance, two important challenges have to be addressed when dealing with multi-sensor systems. Firstly, how to accurately determine the spatial relationship between individual sensor on the robot? This is a vital task known as extrinsic calibration. Without this calibration information, measurements from different sensors cannot be fused. Secondly, how to combine data from multiple sensors to correct for the deficiencies of each sensor, and thus, provides better estimations? This is another important task known as data fusion.

The core of this thesis is to provide answers to these two questions. We cover, in the first part of the thesis, aspects related to improving the extrinsic calibration accuracy, and present, in the second part, novel data fusion algorithms designed to address the ego-motion estimation problem using data from a laser scanner and a monocular camera.

In the extrinsic calibration part, we contribute by revealing and quantifying the relative calibration accuracy of three common types of calibration methods, so as to offer an insight into choosing the best calibration method when multiple options are available. Following that, we propose an optimization approach for solving common motion-based calibration problems. By exploiting the Gauss-Helmert model, our approach is more accurate and robust than the classical least squares model.

In the data fusion part, we focus on camera-laser data fusion and contribute with two new ego-motion estimation algorithms that combine complementary information from a laser scanner and a monocular camera. The first algorithm

utilizes camera image information to guide the laser scan-matching. It can provide accurate motion estimates and yet can work in general conditions without requiring a field-of-view overlap between the camera and laser scanner, nor an initial guess of the motion parameters. The second algorithm combines the camera and the laser scanner information in a direct way, assuming the field-of-view overlap between the sensors is substantial. By maximizing the information usage of both the sparse laser point cloud and the dense image, the second algorithm is able to achieve state-of-the-art estimation accuracy. Experimental results confirm that both algorithms offer excellent alternatives to state-of-the-art camera-laser ego-motion estimation algorithms.

Zusammenfassung

Autonome Roboter und Fahrzeuge sind oft mit mehreren Sensoren ausgerüstet, um essentielle Aufgaben wie Lokalisierung und Kartierung durchzuführen. Ein gemeinsames System verschiedener Sensoren mit unterschiedlichen Messprinzipien liefert oft eine erhöhte Genauigkeit und Zuverlässigkeit bezüglich der Lokalisierung und Kartierung im Vergleich zu Ansätzen mit nur einem Sensor. Um jedoch eine Verbesserung zu erreichen müssen zwei Herausforderungen bei der Verwendung von Mehrsensorsystemen bewältigt werden. Erstens, wie kann die relative Transformation zwischen den verschiedenen Sensoren bestimmt werden? Diese Aufgabe ist als extrinsische Kalibrierung bekannt. Ohne diese Information können die verschiedenen Sensorinformationen nicht kombiniert werden. Zweitens, wie sollen die Daten der verschiedenen Sensoren zur Korrektur der Defizite der einzelnen Sensoren kombiniert werden? Diese wichtige Aufgabe wird als Datenfusion bezeichnet.

Der Kern dieser Dissertation ist es Antworten auf diese zwei Fragestellungen zu geben. Im ersten Teil der Arbeit werden Aspekte zur Verbesserung der Genauigkeit der extrinsischen Kalibrierung behandelt und vorgestellt. Im zweiten Teil werden neuartige Algorithmen zur Fusion von Laser- und Kameradaten für die Schätzung der Eigenbewegung der Sensoren vorgestellt.

Im Teil zur extrinsischen Kalibrierung ist die Offenlegung und Quantifizierung der relativen Kalibrierungsgenauigkeit von drei verbreiteten Typen der Kalibrierung ein Beitrag, der Rückschlüsse zur Wahl einer bestimmten Methodik ermöglicht. Basierend auf diesen Erkenntnissen wird ein Optimierungsverfahren zur Lösung des gemeinhin als AX=XB bekannten Kalibrierungsproblems vorgeschlagen. Durch Verwendung des Gauss-Helmert Model ist unser Ansatz genauer und robuster als herkömmlich verwendete klassische kleinste Quadrate Ansätze.

Im Teil zur Sensordatenfusion fokussieren wir auf die Fusion von Kamera und Laserdaten und stellen zwei Beiträge zur Bewegungsschätzung der Eigenbewe-

gung der komplementären Sensorinformationen vor. Beim ersten Ansatz werden die Vorteile beider Sensoren ausgenutzt – die Fähigkeit der Kamera zur guten Bestimmung von Rotationen der Kamera und die Möglichkeit des Lasersensors zur Bestimmung der Skala durch Registrierung von dreidimensionalen Punktwolken. Solch ein Ansatz funktioniert mit beliebigen Sensorkonfigurationen, selbst wenn die Sichtfelder der Sensoren nicht überlappen. Der zweite Ansatz fusioniert die Sensorinformationen direkt, wobei eine Überlappung der Sichtfelder der Sensoren angenommen wird. Durch Maximierung der Ausnutzung der Sensorinformationen sowohl der dünnbesetzten Laserdaten als auch der dichten Kamerabilder ist unser Ansatz in der Lage hervorragende Ergebnisse bei der Schätzung der Eigenbewegung zu erreichen. Der experimentelle Vergleich mit aktuellen Methoden zeigt, dass vorgestellte Algorithmen eine gute Alternative darstellen.

Acknowledgements

First of all, I am most grateful to my supervisor Prof. Dr. Cyrill Stachniss for his devoted support during the years of my Ph.D. study and research. Without his guidance this thesis would not have been possible.

I furthermore want to express my gratitude to Prof. Dr. Wolfgang Förstner. I benefited a lot from his very knowledgeable photogrammetry book, as well as the daily discussions. I also very much appreciated his efforts in reviewing this thesis and providing me invaluable comments and advice.

All the members of our Photogrammetry and Robotics group are greatly acknowledged for they are nice and supportive. Special thanks are given to Thomas Läbe for his many technical support; also to my office mate, Johannes Schneider, for the engaging discussions we had during work.

I would also like to extend my thanks to the China Scholarship Council (CSC) for funding my Ph.D. studies in Germany.

Finally, I wish to thank my beloved family – my parents and sister, my girlfriend Milly and our son Eugine – for their endless love and support.

Contents

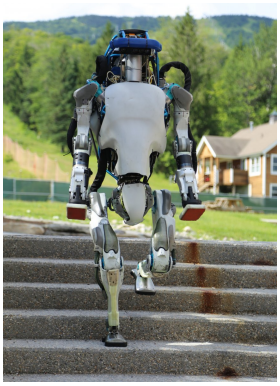
Chapter 1

Introduction

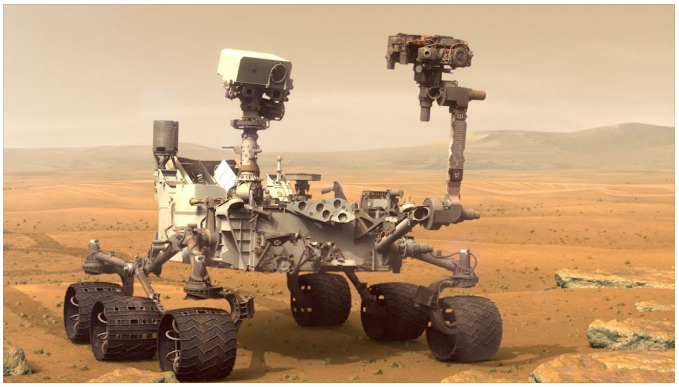
Robotics is certainly one of the key technologies of modern society. Many technology breakthroughs are happening right in this field. For examples, self-driving cars are now more of a reality than an imagination. Driver-less vehicles as shown in Figure 1.1a from the company Waymo [*Self-driving technology* 2018] have already run over 1 billion miles on public roads, up to the date of writing. With this emerging autonomous driving technology, everyone could get around more easily and safely, especially for children, the elderly, and disabled. Traffic collisions due to tired, drunk or distracted driving will be reduced. Time spent commuting could be time spent doing what people want, as the car handles all of the driving without the need for anyone in the driver's seat. Besides self-driving cars, there are also futuristic life-like robots being developed, such as the BigDog [*BigDog, the First Advanced Rough-Terrain Robot* 2018] and Atlas [*Atlas, the World's Most Dynamic Humanoid* 2018] from the company Boston Dynamics. The BigDog, as depicted in Figure 1.1b, is a quadruped robot that can carry heavy payloads for humans and follow them across snowy or rocky terrains, much like a pack mule but will not get tired. The Atlas, as depicted in Figure 1.1c, is the latest most advanced humanoid robot capable of performing surreal athletic actions such as backflips and parkour. A humanoid robot that is agile as such will be very useful in disaster-response operations such as looking for survivors or bodies in the rubble after earthquakes or mining accidents, or to shut down hazardous facilities in dangerous situations. In addition, there are robots deployed even on other planets beyond Earth. The Curiosity [*Curiosity Rover* 2018] as shown in Figure 1.1d is a car-sized robot rover that has been exploring Mars since August 2012 and is still in commission as of the date of writing. Its successful operation has provided invaluable information about the habitability of Mars, making important preparations for future human exploration and space colonization.

(a) Self-driving car

(b) BigDog



(c) Atlas



(d) Curiosity

Figure 1.1: Examples of state-of-the-art autonomous mobile robots. (a) Commercial self-driving cars from Waymo. (b) BigDog, a legged robot serves as a robotic pack mule. (c) Atlas, a humanoid robot with surreal motor skills, from the company Boston Dynamics. (d) Curiosity, a Mars rover performing robotic exploration of the red planet, from NASA. Images originated from (a) https://en.wikipedia.org/wiki/Waymo (b,c) https://www.bostondynamics.com (d) https://www.nasa.gov/mission_pages/msl/images/index.html, all accessed in October 2018.

All four robots mentioned here represent the state-of-the-art robotics technology. They are autonomous robots smart enough to operate in an environment that is inevitably dynamic and uncertain. And the key enabling factor for that lies in their sensory systems, which constantly perceive the environment and then provide necessary information for the robot to act accordingly. With a proper sensor system setup, robots can adapt to the environment and perform basic tasks such as navigation, obstacle avoidance, recognition, or manipulation, even in a challenging environment such as the surface of Mars.

To perform vital tasks such as localization and mapping, autonomous robots often utilize a joint system of multiple sensors with different modalities. This is because by fusing measurements from each sensor, a multi-sensor system is often able to

provide better perception results than individual sensor in terms of completeness or accuracy. Take the colored 3D map shown in Figure 1.2 for example. This map is generated from a multi-sensor system consists of a GPS receiver, a camera and a laser scanner, whose data are depicted in Figure 1.3 respectively. By comparing Figure 1.2 to individual data plot in Figure 1.3, we can see that the colored 3D map produced by a multi-sensor system is more informative and hence desirable.

However, before such a map can be generated, two important questions have to be answered. The first question is how to accurately determine the spatial relationship between individual sensor on the robot? This is an important task known as extrinsic calibration. Without this calibration information, measurements from different sensors cannot be fused. For example, to colorize the 3D laser points shown in Figure 1.3a, the laser points have to be mapped to the pixels of the image as shown in Figure 1.3b, which certainly can not be done if the laser scanner and camera has unknown position. Once the extrinsic calibration is done, the remaining question is then how to combine data from multiple sensors to better solve the task at hand? This is another important task known as data fusion.

1.1 Thesis Outline

This thesis focus on extrinsic calibration and data fusion problems of multi-sensor systems. We cover aspects related to improving the extrinsic calibration accuracy, and present novel data fusion algorithms designed to address the ego-motion estimation problem using data from a laser scanner and a monocular camera. The thesis is organized into eight chapters.

In the next chapter, Chapter 2, "Basic Techniques", we provide short introductions to basic concepts and techniques that are relevant to the thesis.

Chapter 3, "Accuracy Comparison of Common Calibration Models", marks the beginning of the Part I discussion on extrinsic calibration for multi-sensors systems. In Chapter 3, we analyze and quantify the calibration accuracy of three common types of calibration methods named AX=B, AX=YB and AX=XB, to answer the question of "which method is better and why?".

Chapter 4, "Estimation Approach for AX=XB Calibration Problems", continues the discussion on extrinsic calibration problem, especially on the AX=XB type of calibration problem. We discuss the overlooked defect of commonly used ordinary least squares approaches in this context and propose a better estimation approach using the the Gauss-Helmert framework.

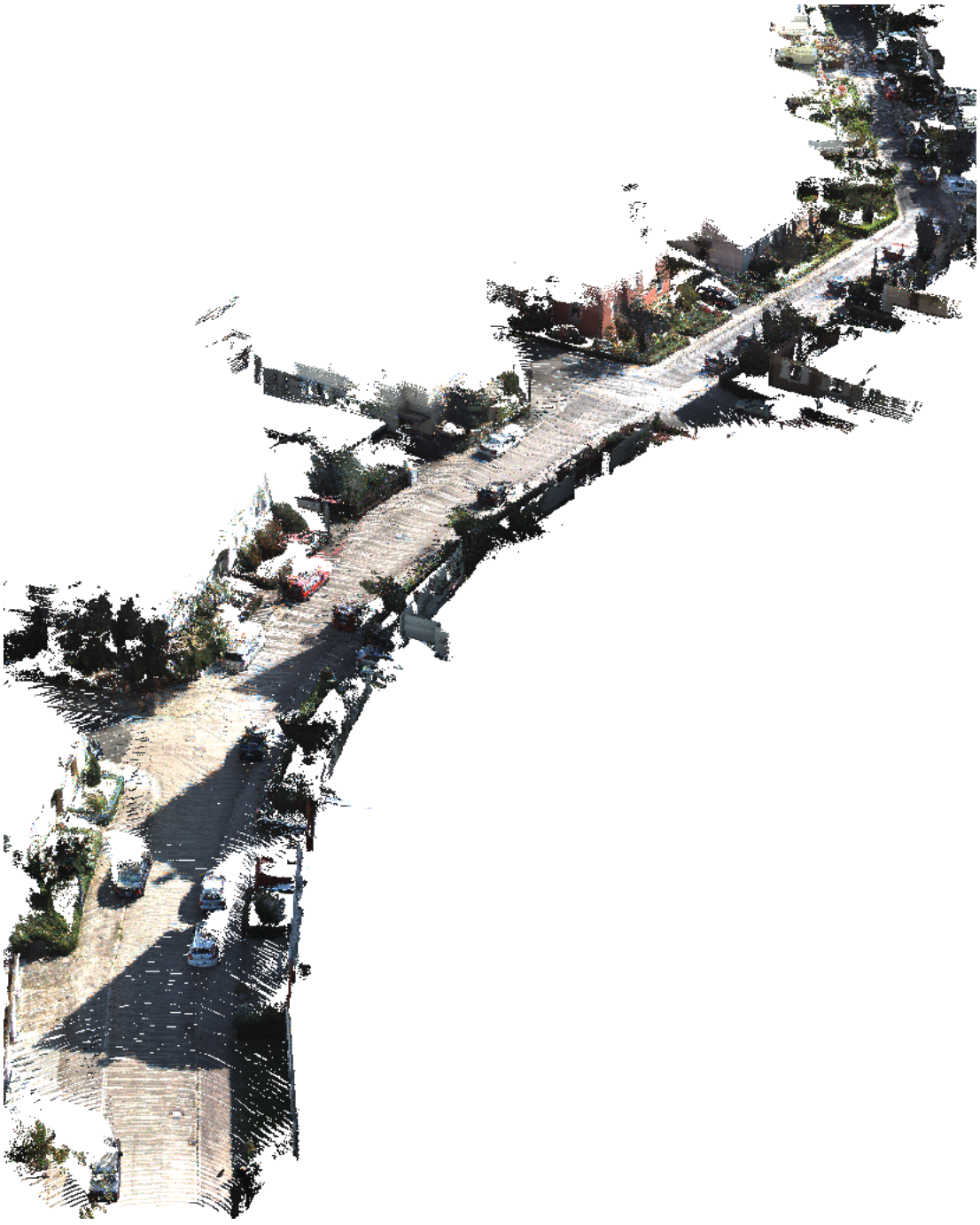
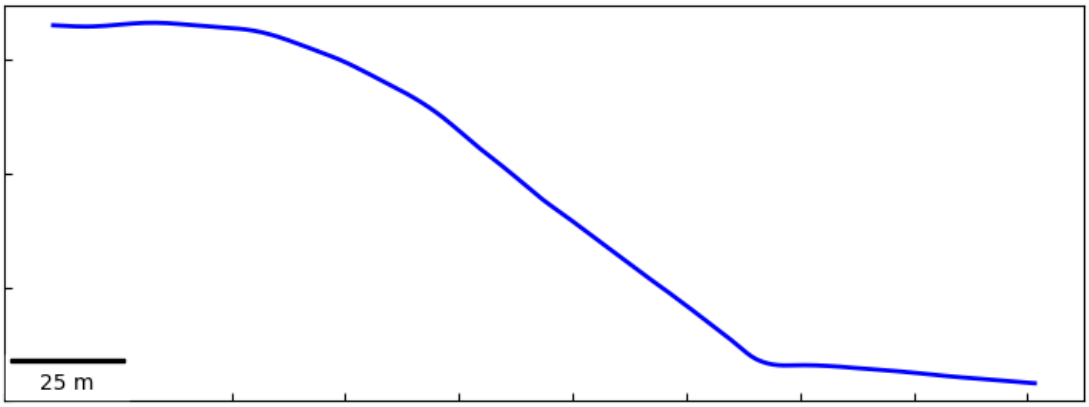


Figure 1.2: A colored 3D map generated by combing data from a GPS receiver, a laser scanner and a camera respectively.

(a) laser scanner data

(b) camera data



(c) GPS data

Figure 1.3: Plots of data from multiple sensors in KITTI dataset: (a) a 3D point cloud generated by a laser scanner; (b) a 2D image captured by a camera; (c) a set of poses measured by a GPS device.

Chapter 5, "Joint Ego-motion Estimation Through Relative Orientation Estimation and 1-DoF ICP", begins the Part II discussion on novel camera-laser fusion algorithms for the ego-motion estimation problem. We present the first approach that exploits image information to guide ICP-based laser scan-matching. It is able to improve the ego-motion estimation accuracy and yet does not require an initial guess of the motion parameters, nor a field-of-view overlap between the camera and the laser scanner.

Chapter 6, "Joint Ego-motion Estimation Through Relative Orientation Estimation and 1-DoF ICP", covers the second approach that fuses the camera and laser scanner information at the lowest level in a direct way, assuming the field-of-view overlap between the sensors is substantial. By exploiting planar information, performing occlusion prediction, and utilizing a two-stage registration, the second approach is able to estimate the ego-motion motion with high accuracy.

Chapter 7, "Related Work", reviews previous work on sensor extrinsic calibration problems and reports existing laser-camera fusion approaches. We discuss strengths and weaknesses of previous research, and explain their relations to our work presented in this thesis.

We finally conclude the thesis in Chapter 8, "Conclusion". We summarize the main insights of this thesis and provide prospects of future work.

1.2 Thesis Contributions

The contributions of the thesis are as follows:

- In Chapter 3, we present a systematic study about the calibration accuracy of three common types of calibration methods. We perform a rigorous study on their noise sensitivity from a novel geometric perspective. As a result, we can reveal and quantify the relative calibration accuracies of the three methods, thus answering the question of "which method is better and why?". Experimental results based on simulations validated our analysis. We are the first to offer such a comparison and it could give us an insight into choosing the best calibration method when multiple options are available.
- In Chapter 4, we present a novel approach for solving the AX=XB type of calibration problem involving multiple sensors. Our approach exploits constraints between the motions of individual sensors and formulates the resulting error minimization problem using the Gauss-Helmert model [Wolf, 1978]. By exploiting the Gauss-Helmert model, we estimate not only the unknown extrinsic parameters but also the pose observation errors, thus recovering

the underlying sensor movements that exactly fulfill the geometric constraints. Compared to the common ordinary least squares approaches that estimate only the parameters, our approach is more accurate and robust to pose measurement noise when estimating extrinsic calibration parameters for multiple sensors, with minor additional computation burden.

- In Chapter 5, we present a novel approach to joint laser-camera motion estimation. Our approach estimates the five-DoF relative orientation from image pairs through feature point correspondences and formulates the remaining scale estimation problem as a variant of the ICP problem with only one DoF. Our approach also exploits the camera information to effectively constrain the data association between laser point clouds. Our approach is able to work in general conditions, without requiring a field-of-view overlap between the camera and the laser scanner, nor an initial guess of the motion parameters.

- In Chapter 6, we propose a novel direct approach to the joint laser-camera motion estimation. Our approach is built upon photometric image alignment and designed to maximize the information usage of both the image and the laser scan to compute an accurate frame-to-frame motion estimate, under the assumption that the field-of-view overlap between the sensors is substantial. Our approach exploits planar information, performs occlusion prediction, and employs a two-stage registration. This allows us to estimate the ego-motions with high accuracy. Experiments on the KITTI and self-recorded datasets supported this claim.

Parts of this thesis have been published in the following peer-reviewed conference:

- K.H. Huang and C. Stachniss (2018a). "Joint Ego-motion Estimation Using a Laser Scanner and a Monocular Camera Through Relative Orientation Estimation and 1-DoF ICP". in: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*

- K.H. Huang and C. Stachniss (2018b). "On Geometric Models and Their Accuracy for Extrinsic Sensor Calibration". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*

- K.H. Huang and C. Stachniss (2017). "Extrinsic Multi-Sensor Calibration For Mobile Robots Using the Gauss-Helmert Model". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*

One work is currently under review:

- K.H. Huang and C. Stachniss (2019). "Accurate Direct Visual-Laser Odometry with Explicit Occlusion Handling and Plane Detection". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*

1.3 Notation

Throughout the thesis, we adopt the following mathematical notation conventions:

- Scalars are typed non-bold letters (such as s, i, j, n, N, E) to distinguish them from vectors and matrices.
- Vectors are typed slanted boldface with lower case letters (such as $\boldsymbol{r}, \boldsymbol{t}, \boldsymbol{\epsilon}$). A vector of all zeros is denoted as $\mathbf{0}$.
- Matrices are also typed slanted boldface but with capital letters (such as $\mathsf{R}, \mathsf{A}, \mathsf{B}$). The dimension of a matrix is indicated with $\mathsf{A} \in \mathbb{R}^{n \times m}$, meaning the matrix A has n rows and m columns. Identity matrices are denoted as I, or as I_n, where the subscript n indicates its dimension. Matrix transpose is denoted as A^T, while A^{-1} denotes a matrix inverse.
- Random variables are denoted with tilde accents (such as $\widetilde{\boldsymbol{t}}_a, \widetilde{\boldsymbol{t}}_b, \widetilde{\theta},$). The variance of a random scalar is denoted as V, while the covariance matrix of a random vector is $\boldsymbol{\Sigma}$.
- Braces $\{\cdot\}$ are used to define sets. For example, $\{\boldsymbol{a}_i\}_{i=1}^N$ denotes a point cloud of N points, with $\boldsymbol{a}_i$ being its elements. $\mathbb{R}$ denotes the set of real numbers. SO(3) denotes the set of 3D rotation matrices. SE(3) denotes the set of 3D rigid transformation matrices.
- $\|\cdot\|$ denotes the Euclidean vector norm.
- $\stackrel{\mathsf{def}}{=}$ indicates a definition, and $\equiv$ denotes the left and right optimization problems are equivalent.

Chapter 2

Basic Techniques

This chapter covers basic concepts and techniques that are relevant to the thesis. We will introduce in Section 2.1 the concept of uncertainty propagation, which is heavily used in the first part of the thesis when discussing extrinsic calibration problems. In Section 2.2, we will cover both closed-form and iterative methods for rotation-matrix estimation. This is necessary because estimating rotation matrices is a basic yet non-straightforward problem which we will frequently encounter throughout the thesis. In the last section 2.3, we will describe the standard iterative closest point method (ICP) for 3D point clouds registration, so as to lay a foundation for the later discussion of laser scan-matching algorithms in the second part of the thesis.

Other concepts or methods related to specific topics will be introduced in the corresponding chapters.

2.1 Variance Propagation

The values of experimental measurements often contain uncertainties (i.e. random errors) due to measurement limitations. Such uncertainties will be propagated if functions are applied to these measurements related random variables. *Variance propagation* is a task that tries to find out the resulting variances of the output variables given the input measurement variances are known. Such a task is useful and sometimes also known as error propagation.

For readers that are not familiar with the principle of variance propagation, we present here a short derivation. A more detailed discussion can be found at the book of Förstner and Wrobel [2016] at pages 42–44.

First, let us consider a noisy multivariate measurement denoted as

$$\widetilde{\boldsymbol{x}} \stackrel{\text{def}}{=} \boldsymbol{\mu}_x + \boldsymbol{\delta}_x, \tag{2.1}$$

whose expectation $\mathbb{E}$ (i.e. mean) and variance $\mathbb{D}$ are

$$\mathbb{E}(\widetilde{\boldsymbol{x}}) = \boldsymbol{\mu}_x, \tag{2.2}$$

$$\mathbb{D}(\widetilde{\boldsymbol{x}}) \stackrel{\text{def}}{=} \mathbb{E}(\boldsymbol{\delta}_x \boldsymbol{\delta}_x^\mathsf{T}) = \boldsymbol{\Sigma}_{xx}. \tag{2.3}$$

According to variance propagation, the target random variable, $\widetilde{\boldsymbol{y}}$, which is computed through a nonlinear function

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}), \tag{2.4}$$

will approximately have the mean and variance of this form:

$$\mathbb{E}(\widetilde{\boldsymbol{y}}) = \boldsymbol{f}(\boldsymbol{\mu}_x) \tag{2.5}$$

$$\mathbb{D}(\widetilde{\boldsymbol{y}}) = \boldsymbol{J}\boldsymbol{\Sigma}_{xx}\boldsymbol{J}^\mathsf{T}, \tag{2.6}$$

where $\boldsymbol{J} \stackrel{\text{def}}{=} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}$ is the Jacobian of function $\boldsymbol{f}$ evaluated at $\boldsymbol{\mu}_x$.

This is due to the fact that

$$\widetilde{\boldsymbol{y}} \stackrel{\text{def}}{=} \boldsymbol{\mu}_y + \boldsymbol{\delta}_y \tag{2.7a}$$

$$\stackrel{\text{def}}{=} \boldsymbol{f}(\widetilde{\boldsymbol{x}}) \tag{2.7b}$$

$$= \boldsymbol{f}(\boldsymbol{\mu}_x + \boldsymbol{\delta}_x) \tag{2.7c}$$

$$\approx \boldsymbol{f}(\boldsymbol{\mu}_x) + \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{\mu}_x} \boldsymbol{\delta}_x + O(|\boldsymbol{\delta}_x|^2) \tag{2.7d}$$

$$\stackrel{\text{def}}{=} \boldsymbol{\mu}_y + \boldsymbol{J}\boldsymbol{\delta}_x. \tag{2.7e}$$

By equating Eq. (2.7a) and (2.7e), we have $\boldsymbol{\delta}_y = \boldsymbol{J}\boldsymbol{\delta}_x$ up to a first order approximation. Thus, the variance of $\widetilde{\boldsymbol{x}}$ (due to $\boldsymbol{\delta}_x$) is propagated to $\widetilde{\boldsymbol{y}}$ as

$$\boldsymbol{\Sigma}_{yy} \stackrel{\text{def}}{=} \mathbb{E}(\boldsymbol{\delta}_y \boldsymbol{\delta}_y^\mathsf{T}) \tag{2.8a}$$

$$= \mathbb{E}[(\boldsymbol{J}\boldsymbol{\delta}_x)(\boldsymbol{J}\boldsymbol{\delta}_x)^\mathsf{T}] \tag{2.8b}$$

$$= \boldsymbol{J}\,\mathbb{E}[\boldsymbol{\delta}_x \boldsymbol{\delta}_x^\mathsf{T}]\,\boldsymbol{J}^\mathsf{T} \tag{2.8c}$$

$$= \boldsymbol{J}\boldsymbol{\Sigma}_{xx}\boldsymbol{J}^\mathsf{T}. \tag{2.8d}$$

Equation (2.6) is hence proved.

Base on this principle, we will use variance propagation to perform noise sensitivity analysis for different calibration methods in Section 3.3, and to determine covariance matrices of intermediate random variables in Section 4.2.2.

2.2 Estimation of Rotation Matrices

In this section, we cover basic methods used to estimate a rotation matrix out of a set of corresponding vector pairs. Such a problem will appear in both the extrinsic calibration and ego-motion estimation problems.

Let us assume there are two corresponding vectors, $\{(\boldsymbol{a}_i, \boldsymbol{b}_i)\}_{i=1}^{N}$, where $\boldsymbol{b}_i$ is the rotated version of $\boldsymbol{a}_i$ but the rotation matrix, $R \in \mathrm{SO}(3)$, is unknown and to be determined. We model this as an optimization problem in which the optimal rotation matrix, R^*, minimizes the deviation between two vector sets:

$$R^* \overset{\text{def}}{=} \underset{R \in \mathrm{SO}(3)}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| R\boldsymbol{a}_i - \boldsymbol{b}_i \right\|^2. \tag{2.9}$$

Equation (2.9) admits both closed-form and iterative solutions. The closed form solution is useful as it does not require an initial guess of R and is efficient to compute. It will be used in Section 2.3 by the ICP point clouds registration algorithm, as well as in Section 4.2.1 for extrinsic calibration problems. The iterative approach, on the other hand, is more flexible and can be extended with useful features such as robust weighting, measurement weighting using the covariance matrix, or to incorporate other objectives for joint estimation of multiple parameters. It therefore serves as an important building block in the following chapters.

2.2.1 Closed Form Approach

To derive a closed form solution for Equation (2.9), we rewrite Equation (2.9) into an equivalent form using the Frobenius matrix norm:

$$R^* \overset{\text{def}}{=} \underset{R \in \mathrm{SO}(3)}{\operatorname{argmin}} \left\| RA - B \right\|_F^2, \tag{2.10}$$

in which $A \overset{\text{def}}{=} [\boldsymbol{a}_1 \ \dots \ \boldsymbol{a}_N]$ is a $\mathbb{R}^{3\times N}$ matrix whose columns are coordinates of $\boldsymbol{a}_i$, and B is similarly defined with $\boldsymbol{b}_i$. The resulting Equation (2.10) is known as the orthogonal Procrustes problem [Gower and Dijksterhuis, 2004] and can be solved by using singular value decomposition (SVD), which is

$$R^* = UV^\mathsf{T}, \tag{2.11a}$$

$$\text{with } UDV^\mathsf{T} \overset{\text{def}}{=} \operatorname{svd}(BA^\mathsf{T}) \tag{2.11b}$$

$$= \operatorname{svd}\left(\textstyle\sum_{i=1}^{N} \boldsymbol{b}_i \boldsymbol{a}_i^\mathsf{T}\right). \tag{2.11c}$$

To prove this, we first recall some of the properties of a matrix trace:

$$\operatorname{trace}(A) \stackrel{\text{def}}{=} \textstyle\sum_i A_{ii}, \tag{2.12}$$

$$\|A\|_F^2 = \operatorname{trace}(A^\mathsf{T} A), \tag{2.13}$$

$$\operatorname{trace}(A) = \operatorname{trace}(A^\mathsf{T}), \tag{2.14}$$

$$\operatorname{trace}(A+B) = \operatorname{trace}(A) + \operatorname{trace}(B), \tag{2.15}$$

$$\operatorname{trace}(ABC) = \operatorname{trace}(BCA) = \operatorname{trace}(CAB). \tag{2.16}$$

The last property (2.16) is of special interest and it means the trace is invariant under cyclic permutations.

The proof of Equation (2.11) begins with:

$$\begin{aligned}
&\operatorname*{argmin}_R \left\|RA - B\right\|_F^2 \\
&= \operatorname*{argmin}_R \operatorname{trace}((RA-B)^\mathsf{T}(RA-B)) && \text{(due to Eq. 2.13)} \\
&= \operatorname*{argmin}_R \operatorname{trace}(A^\mathsf{T}R^\mathsf{T}RA + B^\mathsf{T}B - A^\mathsf{T}R^\mathsf{T}B - B^\mathsf{T}RA) \\
&= \operatorname*{argmin}_R \operatorname{trace}(A^\mathsf{T}A + B^\mathsf{T}B) - 2\operatorname{trace}(A^\mathsf{T}R^\mathsf{T}B) && \text{(due to Eq. 2.14 \& 2.15)} \\
&= \operatorname*{argmax}_R \operatorname{trace}(A^\mathsf{T}R^\mathsf{T}B) \\
&= \operatorname*{argmax}_R \operatorname{trace}(R^\mathsf{T}BA^\mathsf{T}) && \text{(due to Eq. 2.16)} \\
&= \operatorname*{argmax}_R \operatorname{trace}(R^\mathsf{T}UDV^\mathsf{T}) && \text{(due to Eq. 2.11b)} \\
&= \operatorname*{argmax}_R \operatorname{trace}(\underbrace{V^\mathsf{T}R^\mathsf{T}U}_{X} D) && \text{(due to Eq. 2.16)} \\
&\equiv \operatorname*{argmax}_X \operatorname{trace}(XD) \\
&= \operatorname*{argmax}_X \sum_{i=1}^{3} X_{ii}D_{ii}. && (2.20)
\end{aligned}$$

Notice that i) D is a diagonal matrix from the SVD decomposition, its diagonal elements D_{ii} are all non-negative by definition, ii) $X \stackrel{\text{def}}{=} V^\mathsf{T}R^\mathsf{T}U$ is an orthonormal matrix, its diagonal elements X_{ii} are therefore in the range of $[-1, 1]$. By combining these two facts, we can conclude that the objective in the last equation (2.20) is maximized when all $X_{ii} = 1$, in other words:

$$V^\mathsf{T}R^\mathsf{T}U \stackrel{\text{def}}{=} X = I. \tag{2.17}$$

Thus, the solution is $R^* = UV^\mathsf{T}$. In case UV^T has a of determinant -1 instead of 1, to ensure a proper rotation matrix, the solution R^* is set to

$$R^* = UD'V^\mathsf{T}, \tag{2.18}$$

where $D' \stackrel{\text{def}}{=} \text{diag}(1, 1, -1)$ is a diagonal matrix that negates the column of V corresponding to the smaller singular value, see [Umeyama, 1991] for a detailed discussion.

2.2.2 Iterative Least Squares Approach

We can also solve Equation (2.9) with an iterative approach, assuming an initial guess of R is available and to be refined. Such iterative approach has two key components: multiplicative update and rotation parameterization.

First, let us assume at the k-th iteration, we would like to apply a multiplicative update to the current rotation estimate, R^k, with an incremental rotation, $\Delta R \in \text{SO}(3)$, i.e.

$$R^{k+1} \stackrel{\text{def}}{=} \Delta R \, R^k, \tag{2.19}$$

so that R^{k+1} remains a proper rotation matrix after the update.

Then, we parameterize the rotation ΔR with a vector $\Delta \boldsymbol{r} \in \mathbb{R}^3$, in which the rotation axis is parallel to $\Delta \boldsymbol{r}$, the rotation magnitude is $\theta \stackrel{\text{def}}{=} \|\Delta \boldsymbol{r}\|$, and

$$\begin{aligned} \Delta R &\stackrel{\text{def}}{=} \exp([\Delta \boldsymbol{r}]_\times) && \text{(2.20a)} \\ &= I_3 + [\Delta \boldsymbol{r}]_\times + \frac{1}{2!}[\Delta \boldsymbol{r}]_\times^2 + \frac{1}{3!}[\Delta \boldsymbol{r}]_\times^3 + \cdots && \text{(2.20b)} \\ &= I_3 + \frac{\sin\theta}{\theta}[\Delta \boldsymbol{r}]_\times - \frac{1 - \cos\theta^2}{\theta}[\Delta \boldsymbol{r}]_\times^2. && \text{(2.20c)} \end{aligned}$$

Equation (2.20c) is given by Rodriguez [1840], and $[\Delta \boldsymbol{r}]_\times$ is a skew symmetric matrix induced by vector $\Delta \boldsymbol{r}$:

$$[\Delta \boldsymbol{r}]_\times \stackrel{\text{def}}{=} \begin{bmatrix} 0 & -\Delta \boldsymbol{r}_3 & \Delta \boldsymbol{r}_2 \\ \Delta \boldsymbol{r}_3 & 0 & -\Delta \boldsymbol{r}_1 \\ -\Delta \boldsymbol{r}_2 & \Delta \boldsymbol{r}_1 & 0 \end{bmatrix}. \tag{2.21}$$

It is important to note that the product of $[\Delta \boldsymbol{r}]_\times$ and a $\mathbb{R}^3$ vector (e.g. $\boldsymbol{a}$) induces a vector cross product of $\Delta \boldsymbol{r}$ and $\boldsymbol{a}$, which means

$$[\Delta \boldsymbol{r}]_\times \boldsymbol{a} = \Delta \boldsymbol{r} \times \boldsymbol{a} = -\boldsymbol{a} \times \Delta \boldsymbol{r} = [-\boldsymbol{a}]_\times \Delta \boldsymbol{r}. \tag{2.22}$$

Using the rotation parameterization in Equation (2.20b), we can linearize the estimation problem with a first order approximation:

$$\Delta R \approx I_3 + [\Delta \boldsymbol{r}]_\times. \tag{2.23}$$

Such an approximation is reasonable if we assume the incremental rotation magnitude θ is small.

Putting Equations (2.9), (2.19) and (2.23) altogether, the iterative rotation estimation problem becomes:

$$
\begin{aligned}
&\underset{R^{k+1}\in \mathrm{SO}(3)}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| R^{k+1}\boldsymbol{a}_i - \boldsymbol{b}_i \right\|^2 \\
=&\underset{\Delta R\in \mathrm{SO}(3)}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| \Delta R\, R^{k}\boldsymbol{a}_i - \boldsymbol{b}_i \right\|^2 && \text{(due to Eq. 2.19)} \\
\equiv&\underset{\Delta \boldsymbol{r}\in \mathbb{R}^3}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| (I_3 + [\Delta \boldsymbol{r}]_\times) R^{k}\boldsymbol{a}_i - \boldsymbol{b}_i \right\|^2 && \text{(due to Eq. 2.23)} \\
=&\underset{\Delta \boldsymbol{r}\in \mathbb{R}^3}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| [\Delta \boldsymbol{r}]_\times R^{k}\boldsymbol{a}_i + R^{k}\boldsymbol{a}_i - \boldsymbol{b}_i \right\|^2 \\
=&\underset{\Delta \boldsymbol{r}\in \mathbb{R}^3}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| \underbrace{[-R^{k}\boldsymbol{a}_i]_\times}_{J_i} \Delta \boldsymbol{r} + \underbrace{R^{k}\boldsymbol{a}_i - \boldsymbol{b}_i}_{\boldsymbol{\epsilon}_i} \right\|^2 && \text{(due to Eq. 2.22)} \\
\overset{\text{def}}{=}&\underset{\Delta \boldsymbol{r}\in \mathbb{R}^3}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| J_i \Delta \boldsymbol{r} + \boldsymbol{\epsilon}_i \right\|^2. && (2.24)
\end{aligned}
$$

We end up with a unconstrained linear least squares problem in Equation (2.24), and the solution to it is

$$
\Delta \boldsymbol{r}^* = -\Big(\sum_{i=1}^{N} J_i^{\mathsf{T}} J_i\Big)^{-1} \sum_{i=1}^{N} J_i^{\mathsf{T}} \boldsymbol{\epsilon}_i. \tag{2.25}
$$

Once $\Delta \boldsymbol{r}^*$ is determined, we recover the rotation matrix ΔR by using Equation (2.20c), as well as update the parameter R^{k+1} using Equation (2.19). With the newly estimated R^{k+1}, we repeat the process again until the result converges.

2.3 Iterative Closest Point Algorithm

In this section, we will describe the basic iterative closest point algorithm (ICP). ICP is a popular choice for aligning point clouds, hence often used to estimate the transformation between two laser scans.

Let us assume there are two point clouds, $\{\boldsymbol{a}_i \in \mathbb{R}^3\}_{i=1}^N$ and $\{\boldsymbol{b}_j \in \mathbb{R}^3\}_{j=1}^M$. We would like to register the two point clouds in order to determine the relative transformation between the two scanning locations. The relative transformation consists of a rotation $R \in \mathrm{SO3}$, and a translation, $\boldsymbol{t} \in \mathbb{R}^3$. If a point pair $(\boldsymbol{a}_i, \boldsymbol{b}_j)$ belong to the same scene point are correctly registered, we will have the relation

$$R\boldsymbol{a}_i + \boldsymbol{t} = \boldsymbol{b}_j. \tag{2.26}$$

However, both the point correspondences and the transformation are usually unknown and need to be estimated.

Given an initial guess of R and $\boldsymbol{t}$, the ICP will first try to associate the two point clouds by finding for every point $\boldsymbol{a}_i$ its closest point in the other point cloud $\{\boldsymbol{b}\}$. If the matching result is denoted as $\boldsymbol{b}'_i$, then

$$\boldsymbol{b}'_i \overset{\text{def}}{=} \underset{\boldsymbol{b}_j \in \{\boldsymbol{b}\}}{\operatorname{argmin}} \left\| R\boldsymbol{a}_i + \boldsymbol{t} - \boldsymbol{b}_j \right\|^2, \tag{2.27}$$

where R and $\boldsymbol{t}$ are held constant. The minimization in Equation (2.27) is often performed as a k-d-tree based nearest-neighbor search [Bentley, 1975].

Once the point correspondences $(\boldsymbol{a}_i, \boldsymbol{b}'_i)$ are determined, the ICP performs a second step to estimate the transformation, by minimizing the point-to-point matching error:

$$\underset{R,\boldsymbol{t}}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| R\boldsymbol{a}_i + \boldsymbol{t} - \boldsymbol{b}'_i \right\|^2. \tag{2.28}$$

The minimization problem in Equation (2.28) admits a closed from solution. To derive it, we first focus on the parameter $\boldsymbol{t}$ and define $\boldsymbol{\epsilon}_i \overset{\text{def}}{=} R\boldsymbol{a}_i - \boldsymbol{b}'_i$ as well as

$$\Phi(\boldsymbol{t}) \overset{\text{def}}{=} \sum_{i=1}^{N} \left\| R\boldsymbol{a}_i + \boldsymbol{t} - \boldsymbol{b}'_i \right\|^2 \tag{2.29a}$$

$$= \sum_{i=1}^{N} \boldsymbol{t}^\mathsf{T}\boldsymbol{t} + 2\boldsymbol{\epsilon}_i^\mathsf{T}\boldsymbol{t} + \boldsymbol{\epsilon}_i^\mathsf{T}\boldsymbol{\epsilon}_i. \tag{2.29b}$$

To attain the minimum of $\Phi(\boldsymbol{t})$, let $(\frac{\partial \Phi}{\partial \boldsymbol{t}})^\mathsf{T} = \boldsymbol{0}_3$, which is

$$\sum_{i=1}^{N} \boldsymbol{t} + \boldsymbol{\epsilon}_i = \boldsymbol{0}_3. \tag{2.30}$$

Algorithm 1 Standard Point-to-Point ICP

Input:
- Point clouds $\boldsymbol{a}$, $\boldsymbol{b}$
- Initial transformation parameters $(R, \boldsymbol{t})$

Output: Estimated transformation parameters $(R, \boldsymbol{t})$.

repeat
  Transform point cloud $\boldsymbol{a}' \leftarrow R\boldsymbol{a} + \boldsymbol{t}$;
  Associate point cloud $\boldsymbol{b}' \leftarrow \operatorname{argmin}_{\boldsymbol{b}} \|\boldsymbol{a}' - \boldsymbol{b}\|$;
  Compute center of mass $\boldsymbol{c}_a \leftarrow \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{a}_i$;
  Compute center of mass $\boldsymbol{c}_b \leftarrow \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{b}'_i$;
  Compute decomposition $UDV^\mathsf{T} \leftarrow \text{svd}\left(\sum_i (\boldsymbol{b}'_i - \boldsymbol{c}_b)(\boldsymbol{a}_i - \boldsymbol{c}_a)^\mathsf{T}\right)$;
  Determine matrix $D' \leftarrow \text{diag}(1, 1, \det(UV^\mathsf{T}))$;
  Update $R \leftarrow UD'V^\mathsf{T}$;
  Update $\boldsymbol{t} \leftarrow \boldsymbol{c}_b - R\boldsymbol{c}_a$;
until converge **or** maximum iterations reached
return $(R, \boldsymbol{t})$

Therefore, the optimal $\boldsymbol{t}$ should be

$$\boldsymbol{t}^* = -\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\epsilon}_i \tag{2.31a}$$

$$= -\frac{1}{N}\sum_{i=1}^{N} R\boldsymbol{a}_i - \boldsymbol{b}'_i \tag{2.31b}$$

$$= \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{b}'_i - R\left(\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{a}_i\right). \tag{2.31c}$$

Now, let us denote the center-of-mass of point cloud $\{\boldsymbol{a}\}$ as $\boldsymbol{c}_a \overset{\text{def}}{=} \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{a}_i$, and the center-of-mass of point cloud $\{\boldsymbol{b}'\}$ as $\boldsymbol{c}_b \overset{\text{def}}{=} \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{b}'_i$, then from Equation (2.31c), we can recognize that $\boldsymbol{t}^*$ is the difference between the two centers of mass with respect to the reference frame of point cloud $\boldsymbol{b}$, i.e.

$$\boldsymbol{t}^* = \boldsymbol{c}_b - R\boldsymbol{c}_a. \tag{2.32}$$

With that, we can now solve for the rotation R. By substituting $\boldsymbol{t}$ with Equation (2.32), the estimation problem in Equation (2.28) becomes:

$$\operatorname*{argmin}_{R} \sum_{i=1}^{N} \left\| R(\boldsymbol{a}_i - \boldsymbol{c}_a) + (\boldsymbol{b}'_i - \boldsymbol{c}_b) \right\|^2, \tag{2.33}$$

which is an orthogonal Procrustes problem as described in Section 2.2.1. It has a closed form solution as

$$R^* = UD'V^\mathsf{T}, \tag{2.34}$$

with $UDV^\mathsf{T} \stackrel{\text{def}}{=} \text{svd}\left(\sum_i (\boldsymbol{b}'_i - \boldsymbol{c}_b)(\boldsymbol{a}_i - \boldsymbol{c}_a)^\mathsf{T}\right)$ and $D' \stackrel{\text{def}}{=} \text{diag}(1, 1, \det(UV^\mathsf{T}))$.

Once the transformation parameters R and $\boldsymbol{t}$ are updated, the ICP process will start over again and re-associate the two point clouds. It will repeat the whole process until convergent, as summarized in Algorithm 1.

The standard ICP algorithm described here is first developed by Besl and McKay [1992]. There are other more advanced ICP variants nowadays. For a detailed review and comparison for other ICP variants, we refer readers to the work of Rusinkiewicz and Levoy [2001] and Pomerleau et al. [2015].

Part I

Extrinsic Calibration

Chapter 3

Accuracy Comparison of Common Calibration Models

Extrinsic calibration is an important matter for multi-sensor systems, as measurements from different sensors cannot be fused correctly without accurate calibration information. There are various ways to perform the calibration task, but it is not straightforward to tell which method is more accurate and should be preferred when multiple options are available.

In this chapter, we will answer this question by investigating the calibration accuracy of three common types of calibration methods, each represented by the equations AX=B, AX=YB, and AX=XB respectively. We will discuss the advantages and disadvantages of the three calibration models in Section 3.2, and perform a rigorous study on their noise sensitivity from a novel geometric perspective in Section 3.3. As a result, we can quantify and compare the relative calibration accuracy of the three methods, thus answering the question of "which method is better and why?" in Section 3.4. To validate our analytical findings, we conduct numerical simulation experiments in Section 3.5.

3.1 Problem Formulation

In this chapter, we consider the calibration problem between only two sensors, as calibration involving multiple sensors can be formulated in a pair-wise manner without loss of generality (see more discussion in Section 4.3).

Let us assume there are two sensors (named as a and b) that are rigidly attached to a robot. Our calibration task is to estimate the unknown relative transformation matrix (denoted as X) between sensor a and b.

To estimate X, the general calibration procedure involves first setting up an environment in which the robot can move and the sensors can estimate their own poses. Then, we move the robot and record the poses (or trajectories) of each sensor. The obtained pose data (denoted as $\{A_i\}$ for sensor a and $\{B_i\}$ for sensor b) are used to estimate X based on some models (or formulations) as described in the next section. Depending on which model is used, the pose data $\{A_i\}$ and $\{B_i\}$ could be incremental motions relative to past ego-centric reference frames of the sensors, or absolute poses with respect to some fixed coordinate systems. We further denote

$$X \overset{\text{def}}{=} \left[\begin{array}{c|c} O & \boldsymbol{\xi} \\ \hline \mathbf{0} & 1 \end{array}\right] \in \mathrm{SE}(3) \quad \text{and} \quad A_i, B_i \overset{\text{def}}{=} \left[\begin{array}{c|c} R_i & \boldsymbol{t}_i \\ \hline \mathbf{0} & 1 \end{array}\right] \in \mathrm{SE}(3),\ i = 1, \ldots, N \tag{3.1}$$

where O and R are SO(3) rotation matrices, $\boldsymbol{\xi}$ and $\boldsymbol{t}$ are $\mathbb{R}^3$ translation vectors.

Once the calibration is done, the transformation X estimated from $\{A_i\}$ and $\{B_i\}$ can be used to fuse the information from the two sensors. For instance, a scene point $\boldsymbol{p}_b \in \mathbb{R}^3$ originally observed in sensor b's frame can now be transferred to sensor a's frame with the equation

$$\boldsymbol{p}_a = O\boldsymbol{p}_b + \boldsymbol{\xi}. \tag{3.2}$$

3.2 Three Calibration Models

There are three models commonly used in the extrinsic calibration task, namely AX=B, AX=YB, and AX=XB.

3.2.1 Model AX=B

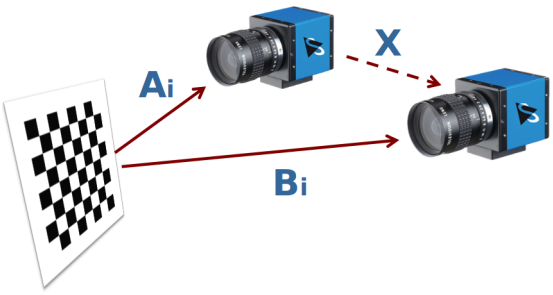


Figure 3.1: Stereo camera calibration using model AX=B. A_i and B_i are absolute poses measured with respect to a common reference frame (i.e. the checkerboard).

In the first model, AX=B, there exists a reference frame common to both sensors, and the pose observations $\{A_i\}$ and $\{B_i\}$ are made with respect to that

global frame. A typical example of this setup is the stereo-camera calibration as illustrated in Figure 3.1. In this case, both cameras look at a checkerboard and a common reference frame is built using this checkerboard. As the physical dimension of the checkerboard is known, the camera poses with respect to the checkerboard (i.e. A_i and B_i) can be estimated by solving the Perspective-N-Point problem. Once the camera poses are provided, their relative transformation X can then be estimated under the formulation

$$A_i X = B_i, \quad \forall i. \tag{3.3}$$

The camera to LiDAR calibration approaches by Khosravian et al. [2017] and Pandey et al. [2015] also fit into this model. In their approaches, the ego-centric frame of either the LiDAR or the camera is chosen as the common global frame and held fix, i.e. $A = I_4$. Then, by performing image-to-point-cloud registration, the other sensor poses is estimated and serves as the intended extrinsic parameter X, i.e. $X = B$, assuming $N = 1$.

The estimation of X with this model is straight forward. A pair of poses (i.e. $N = 1$) is enough to determine the parameter and the estimation can be made explicitly from measurements, i.e. $X = A^{-1}B$. If multiple pairs are available (i.e. $N > 1$), then $X = \text{averaged}(A_i^{-1}B_i)$. As we will see in Section 3.3 as well as in the experiments, model AX=B has a good and stable estimation accuracy compared to the other two models.

The setup of a reference frame common to all sensors often requires control points, landmarks, or reference objects with known geometry. Hence, we refer to calibration methods based on model AX=B as *marker-based approach*. The requirement of a common frame is, however, a major disadvantage of model AX=B as they are hard or even impossible to achieve in some cases. For example, consider calibration problems involving encoders. The encoder of a robot arm or wheel measures nothing other than its own rotation, therefore setting up a direct shared measurement frame with other sensors is impossible. This goes the same for IMU calibrations. Another example for model AX=B hard to apply is camera-to-camera calibration where the cameras have non-overlapping views (as illustrated in Figure 3.2). In this case, a single checkerboard is not sufficient and a more elaborate infrastructure with multiple checkerboards is required. If more than one calibration objects are involved when using model AX=B, we not only need to determine their relative transformations in advance but also need to make sure such information are sufficiently accurate. Otherwise, the estimation result will be biased and contain systematic errors.

3.2.2 Model AX=YB

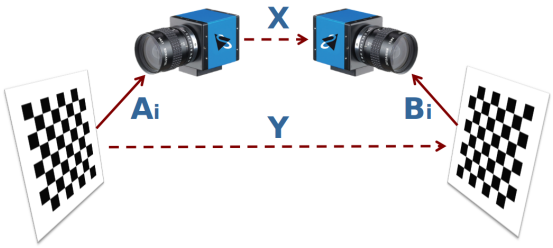


Figure 3.2: Camera-to-camera calibration with model AX=YB. A_i and B_i are absolute poses measured with respect to individual checkerboards.

The applicability problem of model AX=B is mainly due to the requirement of a single reference frame. If we allow the sensors to have their own global reference frames, then we can overcome the applicability problem as well as simplify the calibration process. Such relaxation leads us to the second model, AX=YB, which introduces another unknown parameter Y to represent the transformations between the reference frames.

Figure 3.2 depicts how to use model AX=YB to formulate the previous example of camera-to-camera calibration with non-overlapping camera views. In this case, each camera estimates its poses A_i or B_i with respect to their own checkerboards. The cameras are related to each other by the transformation X, while the checkerboards are related to each other by the transformation Y, which cloud be completely unknown or known but with uncertainties. The pose pair, A_i and B_i, together with the transformations, X and Y, form a quadrilateral and the geometric relation reads

$$A_i X = Y B_i, \quad \forall i. \tag{3.4}$$

Estimating both X and Y requires at least two sets of poses, i.e. $N > 1$.

Using this model, we can record the sensor poses (i.e. $\{A_i\}$ and $\{B_i\}$) independently except time synchronization, hence, the calibration process is simplified and allows for calibrating all kinds of sensors including IMUs and encoders.

3.2.3 Model AX=XB

An alternative to model AX=YB is the third model AX=XB, which addresses the applicability problem of model AX=B by using relative motions as pose measurements instead of absolute ones. In this model, A_i and B_i are incremental motions relative to past ego-centric frames of the sensors. As illustrated in

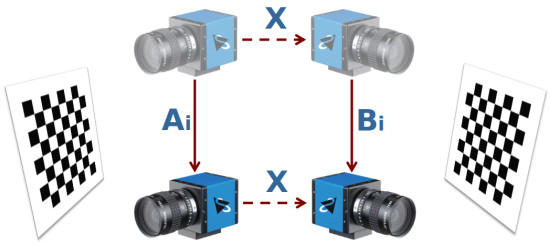


Figure 3.3: Camera-to-camera calibration with model AX=XB. A_i and B_i are incremental motions relative to past ego-centric frames of the sensors.

Figure 3.3, the past and current sensor frames constitute a quadrilateral and the geometric relation becomes

$$A_i X = X B_i, \quad \forall i. \tag{3.5}$$

An obvious advantage of this model over model AX=YB is that there is no need to introduce the additional transformation Y. Besides, using relative motions is sometimes a more attractive option than using absolute poses, because absolute poses are not always available or can be subjected to drift. For instances, the wheel-odometry for ground vehicles provides more accurate instant incremental motion information than the cumulated absolute ones; absolute poses estimated by simultaneous localization and mapping algorithms over long trajectories are inevitable to drift often even with loop-closing. Since model AX=XB utilizes mainly motion information, we refer to it as *motion-based method*.

Similar to model AX=YB, model AX=XB can also be used for calibrating all kinds of sensors and therefore has been widely studied. Previous work by such as Fassi and Legnani [2005]; Park and Martin [1994] proved that it requires at least two sets of poses (i.e. $N > 1$) with non-parallel rotation axes to determine a unique solution for X.

3.3 Noise Sensitivity Analysis

The three models discussed in the previous section can cover probably most of the extrinsic calibration problems we can encounter for mobile robots. For a common calibration problem, we have at least two models to choose from, i.e. the model AX=YB and AX=XB. For calibrations between cameras and LiDARs, we could also use model AX=B and thus have three options. Given multiple options, it is natural to ask "Which one is the best and should be preferred?" To be more specific, we ask the question of "Which model will provide the output X with less uncertainty, assuming that the input noise (or uncertainty) level in A and B are the same for the three models."

To answer this question, we analyze the noise sensitivity of each calibration model. We will first identify the nonlinear function that relates the unknown parameters (i.e. O and $\boldsymbol{\xi}$ of X) and the noisy measurements (i.e. R and $\boldsymbol{t}$ of A and B), and then apply variance propagation (see Section 2.1) to that function to obtain a theoretical lower bound of the estimation uncertainty.

For translation parameter $\boldsymbol{\xi}$, we propose to analyze the scalar entity $\|\boldsymbol{\xi}\|$ instead of carrying out an exhaustive variance propagation to each $\boldsymbol{\xi}$ component. This is because the focus on $\|\boldsymbol{\xi}\|$ allows us to make intuitive interpretation and analysis for each model base on a single equation with only three variables, instead of three complicated equations with a mixture of twelve variables, which apparently cannot be compared directly.

For the estimation of the orientation parameter O, several studies exist and hence will not be covered here. An in-depth discussion of such topic can be found at [Hartley et al., 2013].

In the following discussion, we use tilde accents to denote noisy measurements (e.g. $\widetilde{\boldsymbol{t}}_a, \widetilde{\boldsymbol{t}}_b, \widetilde{\theta}, \dots$) and use V to denote the corresponding variance of additive noise. Other entities that appear in variance propagation without accents are meant to be noise-free latent values. Their values depend on the physical and spatial configuration of the sensors.

3.3.1 Analysis of Model AX=B

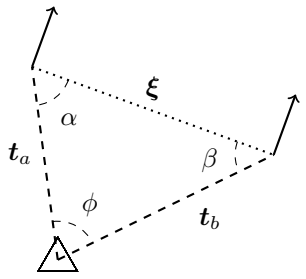


Figure 3.4: Model AX=B

We start with the analysis of model AX=B. The translation part in equation $AX = B$ reads

$$R_a\boldsymbol{\xi} = \boldsymbol{t}_b - \boldsymbol{t}_a. \tag{3.6}$$

Here $(R_a, \boldsymbol{t}_a)$ are from A, and $(R_b, \boldsymbol{t}_b)$ are from B. We can infer from Equation (3.6) that

$$\|\boldsymbol{\xi}\| = \|\boldsymbol{t}_b - \boldsymbol{t}_a\| \tag{3.7}$$

$$= \sqrt{\|\boldsymbol{t}_b\|^2 + \|\boldsymbol{t}_a\|^2 - 2\|\boldsymbol{t}_a\|\|\boldsymbol{t}_b\|\cos\phi}, \tag{3.8}$$

where $\phi \overset{\text{def}}{=} \angle(\boldsymbol{t}_a,\ \boldsymbol{t}_b)$ is the angle between vector $\boldsymbol{t}_a$ and $\boldsymbol{t}_b$, as depicted in Figure 3.4. The (noisy) estimated $\|\widetilde{\boldsymbol{\xi}}\|$ is therefore

$$\|\widetilde{\boldsymbol{\xi}}\| = \sqrt{\|\widetilde{\boldsymbol{t}_b}\|^2 + \|\widetilde{\boldsymbol{t}_a}\|^2 - 2\|\widetilde{\boldsymbol{t}_a}\|\|\widetilde{\boldsymbol{t}_b}\|\cos\phi}. \tag{3.9}$$

The uncertainty of $\|\widetilde{\boldsymbol{\xi}}\|$ is obtained by applying variance propagation to Equation (3.9) and reads

$$V^A_{\|\boldsymbol{\xi}\|} = \left[\frac{\|\boldsymbol{t}_a\| - \|\boldsymbol{t}_b\|\cos\phi}{\|\boldsymbol{\xi}\|}\right]^2 V_{\|\boldsymbol{t}_a\|} + \left[\frac{\|\boldsymbol{t}_b\| - \|\boldsymbol{t}_a\|\cos\phi}{\|\boldsymbol{\xi}\|}\right]^2 V_{\|\boldsymbol{t}_b\|} + \mathcal{O}(V_\phi), \tag{3.10}$$

assuming the noise of $\|\widetilde{\boldsymbol{t}_a}\|$ and $\|\widetilde{\boldsymbol{t}_b}\|$ are uncorrelated.

If we consider the other two angles $\alpha \overset{\text{def}}{=} \angle(\boldsymbol{\xi},\ -\boldsymbol{t}_a)$ and $\beta \overset{\text{def}}{=} \angle(\boldsymbol{\xi},\ \boldsymbol{t}_b)$ within the vector triangle of Figure 3.4, we can obtain

$$\|\boldsymbol{t}_a\| = \cos\phi\|\boldsymbol{t}_b\| + \cos\alpha\|\boldsymbol{\xi}\| \tag{3.11}$$

$$\|\boldsymbol{t}_b\| = \cos\phi\|\boldsymbol{t}_a\| + \cos\beta\|\boldsymbol{\xi}\|, \tag{3.12}$$

and then Equation (3.10) can be simplified to

$$V^A_{\|\boldsymbol{\xi}\|} = \cos^2\alpha\ V_{\|\boldsymbol{t}_a\|} + \cos^2\beta\ V_{\|\boldsymbol{t}_b\|}. \tag{3.13}$$

We can conclude that the more $\boldsymbol{\xi}$ is perpendicular to $\boldsymbol{t}_a, \boldsymbol{t}_b$, the less sensitive is $\widetilde{\boldsymbol{\xi}}$ to noise. $V_{\|\boldsymbol{\xi}\|}$ is bounded by

$$V^A_{\|\boldsymbol{\xi}\|\,\max} = V_{\|\boldsymbol{t}_a\|} + V_{\|\boldsymbol{t}_b\|} \tag{3.14}$$

when $(\alpha, \beta, \phi) = \{(\pi, 0, 0), (0, \pi, 0), (0, 0, \pi)\}$, i.e. $\boldsymbol{t}_a, \boldsymbol{t}_b$ being collinear.

Computing the lower bound for $V_{\|\boldsymbol{\xi}\|}$ is not as straight forward and it depends on the ratio of $V_{\|\boldsymbol{t}_a\|}$ and $V_{\|\boldsymbol{t}_b\|}$. To give a rough idea, we can assume $V_{\|\boldsymbol{t}_a\|} = V_{\|\boldsymbol{t}_b\|}$, then

$$V^A_{\|\boldsymbol{\xi}\|\,\min} = \frac{1}{2}\frac{\|\boldsymbol{\xi}\|^2}{\|\boldsymbol{t}_a\|^2}V_{\|\boldsymbol{t}_a\|} \tag{3.15}$$

which is the case if $\alpha = \beta$. For example, assume $\|\boldsymbol{\xi}\| = 20\,cm$, $\|\boldsymbol{t}_a\| = 1\,m$, $\|\boldsymbol{t}_b\|$ varies but $V_{\|\boldsymbol{t}_a\|} = V_{\|\boldsymbol{t}_b\|} = 1\,cm^2$, then the $\widetilde{\boldsymbol{\xi}}$ estimated by model AX=B will have a standard deviation of $0.014\,cm \sim 1.4\,cm$ in its length.

3.3.2 Analysis of Model AX=XB

We discuss model AX=XB first and leave model AX=YB for last, because there are similarities between the two models and model AX=XB is simpler to start with. The translation part in equation $AX = XB$ reads

$$R_a\boldsymbol{\xi} + \boldsymbol{t}_a = O\boldsymbol{t}_b + \boldsymbol{\xi}. \tag{3.16}$$

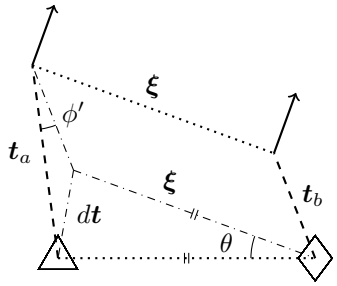


Figure 3.5: Model AX=XB

Unlike in the model AX=B, the two pose pairs now form a vector quadrilateral instead of a triangle. Our target is to find the extrinsic parameters, $\boldsymbol{\xi}$ and O, that "closes" the quadrilateral. Equation (3.16) in its original form does not provide clear clues for this task. Therefore, we introduce an intermediate entity $d\boldsymbol{t}$ and rewrite Equation (3.16) as

$$\underbrace{\boldsymbol{t}_a - O\boldsymbol{t}_b}_{d\boldsymbol{t}} = \boldsymbol{\xi} - R_a\boldsymbol{\xi}. \tag{3.17}$$

The motivation behind is illustrated in Figure 3.5. By assuming a non-zero rotation and shifting the upper pose pair to the lower pair, we can transform the quadrilateral into two triangles that share one side. These two triangles correspond to the left and right term of Equation (3.17) respectively. The shared side, $d\boldsymbol{t}$, is the translation difference between the sensors a and b, as defined by the left term[1]. On the other hand, the right term, $\boldsymbol{\xi} - R\boldsymbol{\xi}$, forms an isosceles triangle. It relates the unknown entity $\boldsymbol{\xi}$ to its two equal sides, with the included angle being the rotation magnitude θ of R_a. From the isosceles triangle, we can infer that

$$\|d\boldsymbol{t}\| = 2\sin(\theta/2)\|\boldsymbol{\xi}\|. \tag{3.18}$$

If $\theta = 0$, Equation (3.18) still hold because $\sin(\theta/2) = 0$ and $\|d\boldsymbol{t}\| = 0$, but $\boldsymbol{\xi}$ is no longer unique and could take any values given fixed $\boldsymbol{t}_a$ and $\boldsymbol{t}_b$. If $\theta \neq 0$, then $\|\widetilde{\boldsymbol{\xi}}\|$ reads

$$\|\widetilde{\boldsymbol{\xi}}\| = \frac{1}{2\sin(\widetilde{\theta}/2)}\|\widetilde{d\boldsymbol{t}}\|. \tag{3.19}$$

By applying variance propagation to Equation (3.19), we obtain the uncertainty of $\|\widetilde{\boldsymbol{\xi}}\|$:

$$V^C_{\|\boldsymbol{\xi}\|} = \left[\frac{\cos(\theta/2)}{4\sin^2(\theta/2)}\|d\boldsymbol{t}\|\right]^2 V_\theta + \left[\frac{1}{2\sin(\theta/2)}\right]^2 V_{\|d\boldsymbol{t}\|} \tag{3.20}$$

or

$$V^C_{\|\boldsymbol{\xi}\|} = \left[\frac{\cos(\theta/2)}{2\sin(\theta/2)}\|\boldsymbol{\xi}\|\right]^2 V_\theta + \left[\frac{1}{2\sin(\theta/2)}\right]^2 V_{\|d\boldsymbol{t}\|} \tag{3.21}$$

[1]When the robot pivots around sensor b, i.e. $\boldsymbol{t}_b = 0$, then $\boldsymbol{\xi}$ and O are completely decoupled, and $d\boldsymbol{t}$ is solely the translation measurement of sensor a because $d\boldsymbol{t} = \boldsymbol{t}_a$.

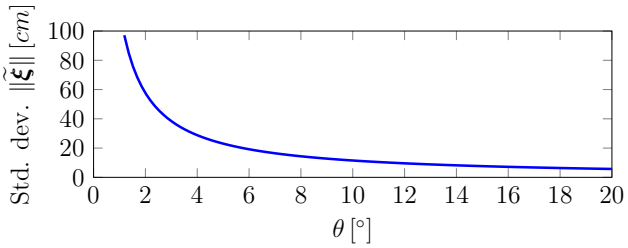


Figure 3.6: Standard deviation of $\|\widetilde{\boldsymbol{\xi}}\|$ in relation to the rotation θ executed by the robot with an assumed noise in the inputs of $V_{\|d\boldsymbol{t}\|} = (1\,cm)^2, V_\theta = (1\,°)^2$. As can be seen, rotations of less than 10° do not allow for determining $\|\boldsymbol{\xi}\|$ well.

due to $\|d\boldsymbol{t}\| = 2\sin(\theta/2)\|\boldsymbol{\xi}\|$.

Furthermore, since $\widetilde{d\boldsymbol{t}}$ is a composition of vectors $\widetilde{\boldsymbol{t}_a}$ and $\widetilde{\boldsymbol{t}_b}$:

$$\widetilde{d\boldsymbol{t}} = \widetilde{\boldsymbol{t}_a} - \widetilde{O\boldsymbol{t}_b}, \tag{3.22}$$

its uncertainty $V_{\|d\boldsymbol{t}\|}$ has a similar form to $V^A_{\|\boldsymbol{\xi}\|}$ in Equation (3.10), which is

$$V_{\|d\boldsymbol{t}\|} = \left[\frac{\|\boldsymbol{t}_a\| - \|\boldsymbol{t}_b\|\cos\phi'}{\|d\boldsymbol{t}\|}\right]^2 V_{\|\boldsymbol{t}_a\|} + \left[\frac{\|\boldsymbol{t}_b\| - \|\boldsymbol{t}_a\|\cos\phi'}{\|d\boldsymbol{t}\|}\right]^2 V_{\|\boldsymbol{t}_b\|}, \tag{3.23}$$

but with $\phi' \overset{\text{def}}{=} \angle(\boldsymbol{t}_a,\ O\boldsymbol{t}_b)$. Substituting $\|d\boldsymbol{t}\|$ with $2\sin(\theta/2)\|\boldsymbol{\xi}\|$, Equation (3.23) can be rewrote as

$$V_{\|d\boldsymbol{t}\|} = \frac{1}{4\sin^2(\theta/2)}\left(\left[\frac{\|\boldsymbol{t}_a\| - \|\boldsymbol{t}_b\|\cos\phi'}{\|\boldsymbol{\xi}\|}\right]^2 V_{\|\boldsymbol{t}_a\|} + \left[\frac{\|\boldsymbol{t}_b\| - \|\boldsymbol{t}_a\|\cos\phi'}{\|\boldsymbol{\xi}\|}\right]^2 V_{\|\boldsymbol{t}_b\|}\right). \tag{3.24}$$

From Equation (3.20), we can see that $V^C_{\|\boldsymbol{\xi}\|}$ consists of two parts. One part is from translational uncertainty ($\|\widetilde{d\boldsymbol{t}}\|$) and the other is from rotational one ($\widetilde{\theta}$). The relative angle θ of the pose pairs plays an important role in both parts. In situations where θ takes small values (e.g. $\theta < 5°$), the factor $\frac{1}{\sin(\theta/2)}$ (and its power) will be large, meaning any noise in the measurements will be significantly amplified. We refer to this as a "degeneration zone". In extreme cases around $\theta = 0$, the uncertainty (or variance) approaches infinity, because the solution is not unique and can take any values. Figure 3.6 illustrates this fact. For example, assume the uncertainty of the pose measurements are $V_\theta = (1\,°)^2$ and $V_{\|d\boldsymbol{t}\|} = (1\,cm)^2$. Then, for a bad configuration where $\theta < 5°$, the uncertainty of the outcome $\|\widetilde{\boldsymbol{\xi}}\|$ will be more than $20\,cm$, which is huge compared to the uncertainty of measurement $\boldsymbol{t}$ $(1\,cm)$.

In contrast to that, if we have a good configuration with a large θ around 180°, the influence of measurement noise will be reduced. The minimum $V_{\|\boldsymbol{\xi}\|}$ is attained

for $\theta = 180°$ with

$$V^C_{\|\boldsymbol{\xi}\|\,\min} = \frac{1}{16}\left(\left[\frac{\|\boldsymbol{t}_a\| - \|\boldsymbol{t}_b\|\cos\phi'}{\|\boldsymbol{\xi}\|}\right]^2 V_{\|\boldsymbol{t}_a\|} + \left[\frac{\|\boldsymbol{t}_b\| - \|\boldsymbol{t}_a\|\cos\phi'}{\|\boldsymbol{\xi}\|}\right]^2 V_{\|\boldsymbol{t}_b\|} \right). \quad (3.25)$$

Our simulation experiment in Section 3.5 confirms this bound.

3.3.3 Analysis of Model AX=YB

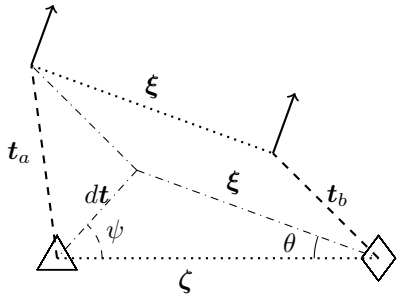


Figure 3.7: Model AX=YB

The analysis of model AX=YB is similar to model AX=XB in that they both form a vector quadrilateral, but model AX=YB has more unknowns (namely X and Y). Let us denote O' as the rotation part of Y, and $\boldsymbol{\zeta}$ as the translation part. Then, the translation part in $AX = YB$ reads

$$R_a\boldsymbol{\xi} + \boldsymbol{t}_a = O'\boldsymbol{t}_b + \boldsymbol{\zeta}, \quad (3.26)$$

which we rewrite as

$$\underbrace{\boldsymbol{t}_a - O'\boldsymbol{t}_b}_{d\boldsymbol{t}} = \boldsymbol{\zeta} - R_a\boldsymbol{\xi}. \quad (3.27)$$

In this model, the triangle that relates $\boldsymbol{\xi}$ and $d\boldsymbol{t}$ is no longer isosceles and an extra unknown side $\boldsymbol{\zeta}$ is present, as depicted in Figure 3.7. The geometric relation becomes

$$\sin\theta\|\boldsymbol{\xi}\| = \sin\psi\|d\boldsymbol{t}\| \quad (3.28)$$

with $\psi \stackrel{\text{def}}{=} \angle(\boldsymbol{\zeta},\ d\boldsymbol{t})$.

If $\sin\theta = 0$, (i.e. $\theta = 0$ or π, $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$ are collinear), we have $\sin\psi = 0$. Equation (3.28) still holds but the system becomes degenerated in the sense that the solution to $\widetilde{\boldsymbol{\xi}}$ is not unique, so does $\widetilde{\boldsymbol{\zeta}}$ because $\widetilde{\boldsymbol{\zeta}} = \widetilde{d\boldsymbol{t}} + R_a\widetilde{\boldsymbol{\xi}}$. Assuming $\sin\theta \neq 0$, then $\|\widetilde{\boldsymbol{\xi}}\|$ relates to $\|\widetilde{d\boldsymbol{t}}\|$, $\widetilde{\psi}$ and $\widetilde{\theta}$ with

$$\|\widetilde{\boldsymbol{\xi}}\| = \frac{\sin\widetilde{\psi}}{\sin\widetilde{\theta}}\|\widetilde{d\boldsymbol{t}}\|. \quad (3.29)$$

Table 3.1: Overview of $V_{\|\boldsymbol{\xi}\|}$ for the three models.

Model	$V_{\|\boldsymbol{\xi}\|}$	Degeneration zones
AX=B	$\left[2\sin(\theta/2)\right]^2 V_{\|d\boldsymbol{t}\|}$	0
AX=XB	$\left[\frac{\cos(\theta/2)}{2\sin(\theta/2)}\|\boldsymbol{\xi}\|\right]^2 V_\theta + \left[\frac{1}{2\sin(\theta/2)}\right]^2 V_{\|d\boldsymbol{t}\|}$	1
AX=YB	$\left[\frac{\cos\psi}{\sin\psi}\|\boldsymbol{\xi}\|\right]^2 V_\psi + \left[\frac{\cos\theta}{\sin\theta}\|\boldsymbol{\xi}\|\right]^2 V_\theta + \left[\frac{\sin\psi}{\sin\theta}\right]^2 V_{\|d\boldsymbol{t}\|}$	2

By applying variance propagation to Equation (3.29) and omitting the correlation terms, the uncertainty of $\|\widetilde{\boldsymbol{\xi}}\|$ is

$$V^B_{\|\boldsymbol{\xi}\|} = \left[\frac{\cos\theta\sin\psi}{\sin^2\theta}\|d\boldsymbol{t}\|\right]^2 V_\theta + \left[\frac{\cos\psi}{\sin\theta}\|d\boldsymbol{t}\|\right]^2 V_\psi + \left[\frac{\sin\psi}{\sin\theta}\right]^2 V_{\|d\boldsymbol{t}\|} \tag{3.30}$$

which can be rewrote as

$$V^B_{\|\boldsymbol{\xi}\|} = \left[\frac{\cos\theta}{\sin\theta}\|\boldsymbol{\xi}\|\right]^2 V_\theta + \left[\frac{\cos\psi}{\sin\psi}\|\boldsymbol{\xi}\|\right]^2 V_\psi + \left[\frac{\sin\psi}{\sin\theta}\right]^2 V_{\|d\boldsymbol{t}\|} \tag{3.31}$$

due to $\|\boldsymbol{\xi}\| = \frac{\sin\psi}{\sin\theta}\|d\boldsymbol{t}\|$.

The uncertainty $V^B_{\|\boldsymbol{\xi}\|}$ consists of one translation and two rotation parts. We can identify the two degeneration zones around $\theta = 0$ and $\theta = \pi$ from Equation (3.31), which are due to the squared cotangent factors of V_ψ and V_θ.

The translation part is not as straight forward as in model AX=XB, but we can show that, assuming $\|\boldsymbol{\zeta}\| > \|\boldsymbol{\xi}\|$, the factor $\frac{\sin\psi}{\sin\theta}$ is bounded by

$$\frac{\|\boldsymbol{\xi}\|}{\|\boldsymbol{\zeta}\| + \|\boldsymbol{\xi}\|} < \frac{\sin\psi}{\sin\theta} < \frac{\|\boldsymbol{\xi}\|}{\|\boldsymbol{\zeta}\| - \|\boldsymbol{\xi}\|}, \tag{3.32}$$

because of Equation (3.28) and $\|\boldsymbol{\zeta}\| - \|\boldsymbol{\xi}\| < \|d\boldsymbol{t}\| < \|\boldsymbol{\zeta}\| + \|\boldsymbol{\xi}\|$. From the perspective of Equation (3.32), we can in theory increase the calibration accuracy by making $\|\boldsymbol{\zeta}\|$ larger, e.g, separating the global frames further away from each other.

3.4 Accuracy Comparison

Given Equations (3.10), (3.21) and (3.31), see also Table 3.1, we are now able to compare the three models. For this comparison, we assume the same trajectory and measurement noise.

First, we compare model AX=YB and model AX=XB. From the geometry perspective, model AX=XB can be seen as a special case of model AX=YB with $\boldsymbol{\zeta} = \boldsymbol{\xi}$. In this case, we can exploit this equality for the comparison of the models, as it implies

$$\psi = (\pi - \theta)/2. \tag{3.33}$$

This results in

$$\frac{\sin\psi}{\sin\theta} = \frac{\sin(\pi/2 - \theta/2)}{\sin\theta} = \frac{\cos(\theta/2)}{2\sin(\theta/2)\cos(\theta/2)} = \frac{1}{2\sin(\theta/2)}. \tag{3.34}$$

Given that, the second term of Equation (3.21) and the third term of Equation (3.31) are equal. Additionally exploiting $\cot\theta = \frac{\cos\theta}{\sin\theta}$ and assuming $V_\psi = \frac{1}{4}V_\theta$, we simplify $V^B_{\|\boldsymbol{\xi}\|} - V^C_{\|\boldsymbol{\xi}\|}$ for our comparison as follows:

$$\begin{aligned}
&V^B_{\|\boldsymbol{\xi}\|} - V^C_{\|\boldsymbol{\xi}\|} \\
&= \|\boldsymbol{\xi}\|^2 \left[V_\theta \cot^2\theta + V_\psi \cot^2\psi - \frac{V_\theta}{4}\cot^2\frac{\theta}{2} \right] && (3.35)\\
&= \|\boldsymbol{\xi}\|^2 \left[V_\theta \cot^2\theta + \frac{V_\theta}{4}\tan^2\frac{\theta}{2} - \frac{V_\theta}{4}\cot^2\frac{\theta}{2} \right] && (3.36)\\
&= V_\theta \|\boldsymbol{\xi}\|^2 \underbrace{\left[\cot^2\theta + \frac{1}{4}\tan^2\frac{\theta}{2} - \frac{1}{4}\cot^2\frac{\theta}{2} \right]}_{g(\theta)}. && (3.37)
\end{aligned}$$

The term $g(\theta)$ in Equation (3.37) provides us the insight that, under which circumstances model AX=XB is better than model AX=YB, or vice versa. We have

$$\begin{cases} g(\theta) < 0, & \text{if } \theta < \pi/2 \\ g(\theta) = 0, & \text{if } \theta = \pi/2 \\ g(\theta) > 0, & \text{otherwise.} \end{cases} \tag{3.38}$$

See also Figure 3.8 for a plot of this term. The term $g(\theta)$ is smaller than zero (between -0.5 and 0) for $\theta \in [0°, 90°)$, meaning model AX=YB is (slightly) better than model AX=XB in that range. For $\theta \in [90°, 180°]$, this term is larger than zero and even approaches infinity, such that model AX=XB is substantially better than model AX=YB here.

Second, we compare model AX=B to model AX=XB. Equation (3.10) shows that $V^A_{\|\boldsymbol{\xi}\|}$ is independent of θ, which is an advantage of model AX=B over AX=YB and AX=XB. Because the terms related to θ can lead to large uncertainty or degeneration zones as we saw before. By comparing $V^A_{\|\boldsymbol{\xi}\|}$ to the theoretical minimum value of $V^C_{\|\boldsymbol{\xi}\|}$ from Equation (3.25), we can see that model AX=XB can in theory outperform model AX=B when θ is large. However, as we will observe in the experimental evaluation, model AX=B is less sensitive to degenerate cases

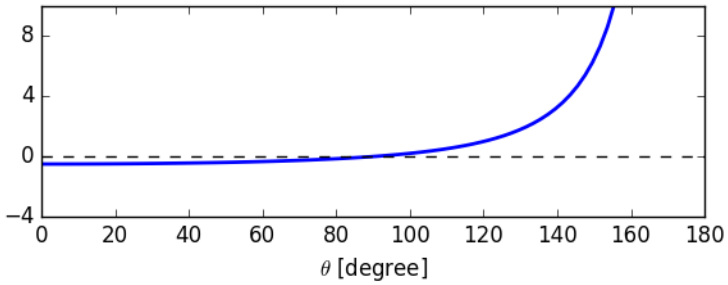


Figure 3.8: Illustration of the function $g(\theta)$ in Equation (3.37) depending on the value of θ. As can be seen, for $\theta < 90°$, model AX=YB is slightly better than model AX=XB, but for value larger than 90°, model AX=YB degenerate quickly. In contrast, model AX=XB performs well (compare also Figure 3.9).

and thus, should be preferred over model AX=YB and model AX=XB in most practical situation. Only for large values of $\theta > 60°$, model AX=XB is better than model AX=B. Furthermore, model AX=YB never outperforms model AX=B.

3.5 Experimental Evaluation

We conduct Monte Carlo simulations to validate our analytical analysis. We generate trajectories with controlled configuration parameters for $\|\boldsymbol{t}\|, \theta$, etc. We add zero-mean Gaussian noise to the simulated pose measurements and estimate the extrinsic parameters, O and $\boldsymbol{\xi}$, with a nonlinear least-squares approach which will be described later in the Chapter 4. Our estimation approach is based on the Gauss-Helmert paradigm and is able to provide a statistical optimal solution up to the Cramer-Rao bound. Meanwhile, all solutions are initialized with ground-truth values to rule out possible effects of local minima. In the end, we compute the error as the difference in the length of the vectors $\boldsymbol{\xi}$ and $\widetilde{\boldsymbol{\xi}}$, i.e., $\|\boldsymbol{\xi}\| - \|\widetilde{\boldsymbol{\xi}}\|$, and calculate the root mean square (RMSE) for each model. The common setup for the simulations are $\|\boldsymbol{\zeta}\| = \|\boldsymbol{\xi}\| = 1\,m$, $\|\widetilde{\boldsymbol{t}_a}\| = 10\,m$, and noise variance are set to $V_{\|\boldsymbol{t}_a\|} = V_{\|\boldsymbol{t}_b\|} = (0.01\,m)^2$, $V_\theta = (0.001 \text{ rad})^2$. We generated 1000 trials/trajectories per value of θ and each trajectory consists of 100 poses, all evaluated for the three models. The result of our Monte Carlo simulations is depicted in Figure 3.9. It shows the RMSE for each model with varying values of θ.

For model AX=B, the RMSE plot is almost straight and with minimal variations as expected. The curve of model AX=YB is dominated by the shape of the squared cotangent function. We can visually identify the two degeneration zones around $\theta = 0°$ and $\theta = 180°$, which are due to the squared cotangent factor

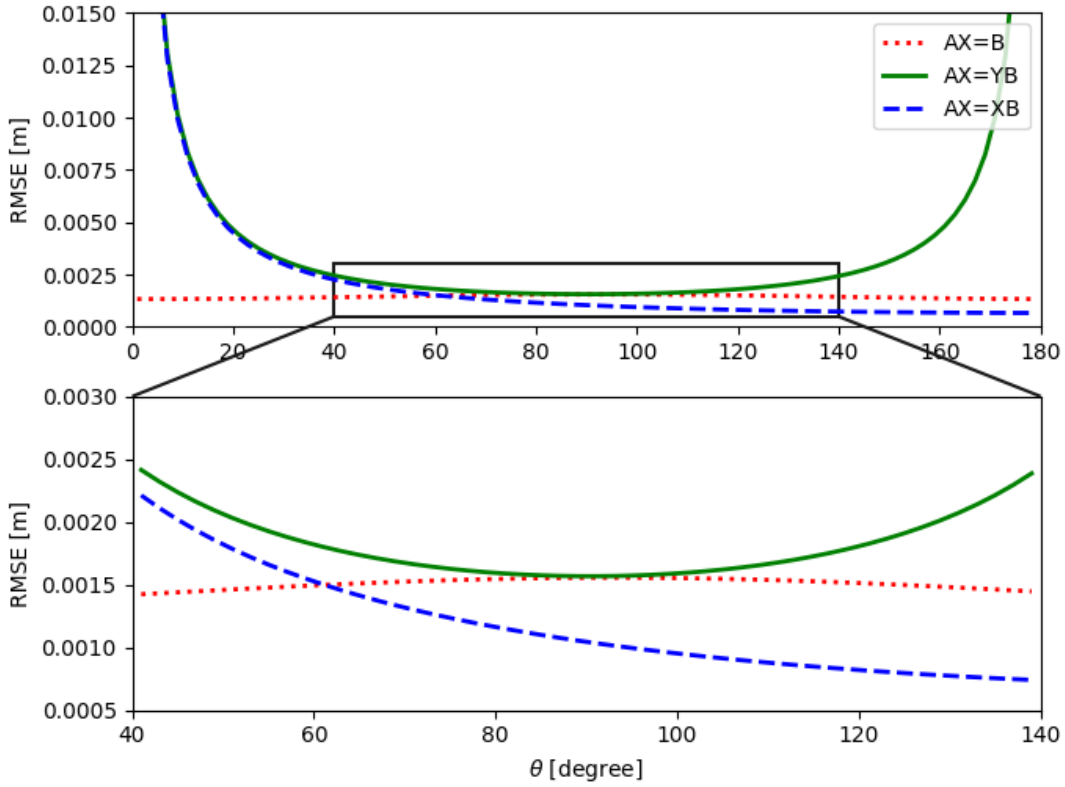


Figure 3.9: Result from a simulation experiment for calibrating two sensors with all three models along the same trajectories. The plot shows the RMSE of the three models depending on θ with a close-up view for the interval $\theta \in [40°, 140°]$. As can be seen, model AX=B performs well, independent of θ. Model AX=XB is better than model AX=B for $\theta > 60°$. Model AX=YB never outperforms model and has practically no advantages over model AX=XB.

of V_ψ and V_θ, see Equation (3.31). For model AX=XB, the RMSE plot shows almost no difference to that of model AX=YB for values $\theta < 40°$, but it is strictly decreasing with θ and has only one degeneration zone around $\theta = 0$.

Comparing the three models, we see the situation changes as the value of θ increases. Model AX=B's near constant performance remains the best until θ reaches 60°, then model AX=XB becomes the best and eventually its RMSE takes only half of model AX=B's RMSE when $\theta = 180°$. Model AX=YB performs the worst among the three models in this experiment, only in a narrow range (from 80° to 100°) it is close to model AX=B.

The noticeable deviation of model AX=YB's and AX=XB's curve begins at $\theta = 40°$ instead of expected $\theta = 90°$. A possible explanation is because the correlation terms we omitted in Equation (3.31) also have a substantial influence on $V^B_{\|\boldsymbol{\xi}\|}$. But this does not change our conclusion about the accuracy comparison.

To sum up, the results of the simulation experiment support our analytical analysis and suggest that model AX=B is the preferred solution. For controlled cases with $\theta > 60°$, one can consider model AX=XB instead. The use of model AX=YB, however, should be avoided.

3.6 Summary

In this chapter, we presented a systematic study about the calibration accuracy of three kinds of calibration methods, namely AX=B, AX=XB and AX=YB. We discussed the advantages and disadvantages of each model and performed a rigorous study on their noise sensitivity. We showed how the sensor configuration and calibration setup influence the calibration accuracy and answered the question of "which model is better?". Contradict to the common conception that marker-based methods (i.e. model AX=B) are always superior, we showed that in some cases the motion-based methods (i.e. model AX=XB) can be better than marker-based approaches. In summary, we conclude that if the calibration setup allow for using model AX=B, it is a good choice and should be used. For controlled settings with $\theta > 60°$, one should also consider model AX=XB as it can provide better estimates of the parameters and typically requires less calibration infrastructure. If model AX=B cannot be applied, model AX=XB is more appropriate than model AX=YB, but for small values of θ both models degenerate.

Chapter 4

Estimation Approach for AX=XB Calibration Problems

In the last chapter, we concluded that calibration methods based on model AX=B and AX=XB should be preferred after making an accuracy comparison. The estimation with model AX=B is relatively straightforward, but the one with model AX=XB is unfortunately not as easy and requires special care to ensure high calibration accuracy. Yet, model AX=XB is the only option in many cases. Therefore, in this chapter, we will focus on the estimation approach for AX=XB calibration problem.

We first introduce the detailed formulation of the problem in Section 4.1 as well as its simple closed form solution in Section 4.2.1. We then discuss the traditional least squares estimation approach in Section 4.2.2 and point out its overlooked defect in the context of AX=XB estimation problem. In Section 4.2.3, we present our solution based on the Gauss-Helmert paradigm, which estimates not only the extrinsic parameters but also the pose observation errors, thus recovering the underlying sensor movements that exactly fulfill the motion constraints. The reason and advantage of doing so is explained. In Section 4.3, we extend our approach to cover multi-sensors cases. We implemented our approach and tested it on real robot. The experiment results are shown in Section 4.4 and confirm that our approach is able to accurately determine the extrinsic configuration of each sensor and can largely improve the accuracy when the noise level is high.

4.1 Problem Formulation

In the following discussion, we utilize the angle-axis representation for rotation parameterization. Specifically, we use $\boldsymbol{r}$ to denote the angle-axis vector of the rotational measurements in A and B, i.e. R

$$R \stackrel{\text{def}}{=} \exp([\boldsymbol{r}]_\times), \tag{4.1}$$

and use $\boldsymbol{\eta}$ to denote the unknown rotational parameters in X, i.e. O:

$$O \stackrel{\text{def}}{=} \exp([\boldsymbol{\eta}]_\times). \tag{4.2}$$

Here $[\cdot]_\times$ is the skew operation as described in Section 2.2.2.

Let us assume there are a number of N motion segments, $\{A_i, B_i\}_{i=1}^N$, being used for the calibration. For each pair of A_i and B_i, we have a constraint equation $A_iX = XB_i$, which can be split into two parts:

$$R_{ai}\boldsymbol{\xi} + \boldsymbol{t}_{ai} = O\boldsymbol{t}_{bi} + \boldsymbol{\xi}, \tag{4.3}$$

$$R_{ai}O = OR_{bi}. \tag{4.4}$$

The upper Equation (4.3) is for the translation and the lower Equation (4.4) is for the rotation, which can be further simplified to

$$\boldsymbol{r}_{ai} = O\boldsymbol{r}_{bi}, \tag{4.5}$$

due to the angle-axis representation.

When discussing the least squares estimation approaches, we will refer to the unknown parameters collectively as $\boldsymbol{x}$ and the measurements collectively as $\boldsymbol{l}_i$:

$$\boldsymbol{x} \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{\xi} \end{bmatrix}, \quad \boldsymbol{l}_i \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{r}_{ai} \\ \boldsymbol{r}_{bi} \\ \boldsymbol{t}_{ai} \\ \boldsymbol{t}_{bi} \end{bmatrix}, \quad i = 1, \ldots, N. \tag{4.6}$$

Also, we define two error functions, $\boldsymbol{g}_t$ and $\boldsymbol{g}_r$, from Equation (4.5) and Equation (4.3):

$$\boldsymbol{g}_t(\boldsymbol{x}, \boldsymbol{l}_i) \stackrel{\text{def}}{=} [R(\boldsymbol{r}_{ai}) - I_3]\boldsymbol{\xi} + \boldsymbol{t}_{ai} - O(\boldsymbol{\eta})\boldsymbol{t}_{bi}, \tag{4.7}$$

$$\boldsymbol{g}_r(\boldsymbol{x}, \boldsymbol{l}_i) \stackrel{\text{def}}{=} \boldsymbol{r}_{ai} - O(\boldsymbol{\eta})\boldsymbol{r}_{bi}. \tag{4.8}$$

These two error functions are at the core of our estimation problem. In the noise-free case, a true solution $\boldsymbol{x}^*$ will fulfill

$$\boldsymbol{g}(\boldsymbol{x}^*, \boldsymbol{l}_i) \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{g}_t(\boldsymbol{x}^*, \boldsymbol{l}_i) \\ \boldsymbol{g}_r(\boldsymbol{x}^*, \boldsymbol{l}_i) \end{bmatrix} = \boldsymbol{0}, \quad \forall i. \tag{4.9}$$

4.2 Solutions to AX=XB Problems

4.2.1 Closed Form Solution

Closed-form solutions are useful if we have no prior knowledge or initial guess of the parameters. The AX=XB problem admits a simple closed-form solution if we decouple the rotation and translation estimation by minimizing the algebraic errors, $\boldsymbol{g}_t$ and $\boldsymbol{g}_r$ defined in Equation (4.7) and Equation (4.8), separately.

First of all, the estimation problem of O with Equation (4.5) alone is in essence an axis-alignment problem (see Hartley et al. [2013]). Let us define

$$O^* \stackrel{\text{def}}{=} \underset{O \in \mathrm{SO}(3)}{\operatorname{argmin}} \sum_i^N \|\boldsymbol{r}_{ai} - O\boldsymbol{r}_{bi}\|^2, \tag{4.10}$$

then the closed form solution to O will be:

$$O^* = UD'V^\mathsf{T}, \tag{4.11}$$

with $UDV^\mathsf{T} \stackrel{\text{def}}{=} \mathrm{svd}\left(\sum_i^N \boldsymbol{r}_{ai}\boldsymbol{r}_{bi}^\mathsf{T}\right)$ and $D' \stackrel{\text{def}}{=} \mathrm{diag}\left(1, 1, \det(UV^\mathsf{T})\right)$, as discussed in Section 2.2.1.

Once O^* is estimated, we can fix the value of O in Equation (4.3) and estimate the translation parameter $\boldsymbol{\xi}$ directly as a linear least squares problem:

$$\boldsymbol{\xi}^* \stackrel{\text{def}}{=} \underset{\boldsymbol{\xi}}{\operatorname{argmin}} \sum_i^N \left\|[R(\boldsymbol{r}_{ai}) - I_3]\boldsymbol{\xi} + [\boldsymbol{t}_{ai} - O'\boldsymbol{t}_{bi}]\right\|^2. \tag{4.12}$$

Besides this approach, there are various closed form methods to address the AX=XB problem, including methods that can jointly estimate the rotation and translation by using, for example, dual-quaternion [Daniilidis, 1999].

Closed form methods are useful because they do not require iterations and approximate values of the parameters, their solutions are, however, often far from perfect due to their statistically suboptimal nature: they never address the statistical uncertainty of the measurements, but instead, they optimize some heuristically chosen algebraic expression (Förstner and Wrobel [2016] p.178). Such a strategy allows them to be solved as (simple) singular value problems but also renders the solutions vulnerable to measurement noises and outliers, thus less accurate.

4.2.2 Ordinary Least Squares Based Solution

To obtain a more accurate calibration/estimation result, it is necessary to employ an iterative refinement process using least squares estimation, which not only jointly estimates $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$, but also fully take into account the measurement uncertainties. To this end, a widely used least squares estimation approach for the AX=XB problem is to minimize the weighted sum of squared errors defined by $\boldsymbol{g}_t$ and $\boldsymbol{g}_r$:

$$\underset{\boldsymbol{x}}{\operatorname{argmin}} \sum_{i}^{N} \|\boldsymbol{g}_t(\boldsymbol{x}, \boldsymbol{l}_i)\|^2_{W_{ti}} + \|\boldsymbol{g}_r(\boldsymbol{x}, \boldsymbol{l}_i)\|^2_{W_{ri}}, \tag{4.13}$$

where $\|\boldsymbol{g}\|^2_W \overset{\text{def}}{=} \boldsymbol{g}^\mathsf{T} W \boldsymbol{g}$, and W is a positive-definite weight matrix.

Formulation (4.13) or its variants with different rotation parameterizations are prominent in the current robotics community, because optimizing such an ordinary-least-squares problem is easy to implement with the help of popular off-the-self optimizer such as g2o [Kümmerle et al., 2011] or Ceres [Agarwal and Mierle, 2010]. The work of Strobl and Hirzinger [2006] and Guo et al. [2012] are such examples.

However, the optimality of choosing such formulation is seldom discussed in most previous work, and how to set the values of the weight matrices W_{ri} and W_{ti} is also rarely mentioned. As we will see in the following discussion, there is a flaw in this formulation, which will limit the estimation accuracy when the measurement noise-level is high. Therefore, this formulation is only suitable for low noise situations.

We refer to the solution of Equation (4.13) as the Gauss–Markov model based solution (GM for short), because a plausible justification is the "Gauss–Markov" theorem, which says the least squares estimate gives *no bias and has minimal variance* if the functional relation $\boldsymbol{l} = \boldsymbol{f}(\boldsymbol{x})$ is linear and the weights are chosen to be $W = \boldsymbol{\Sigma}_{ll}^{-1}$, assuming the measurement errors are zero mean and statistically independent. The Gauss–Markov theorem also holds approximately for a nonlinear $\boldsymbol{f}$ if the variances of the measurement errors are small compared to the second derivatives of the functions (Förstner and Wrobel [2016] p.79).

Based on the Gauss–Markov theorem, we see that Equation (4.13) is actually using the residual vectors $\boldsymbol{g}_r$ and $\boldsymbol{g}_t$ as the measurement entities $\boldsymbol{l}$ rather than using the original measurements $\boldsymbol{r}$ and $\boldsymbol{t}$. The weight matrices W_r and W_t should therefore take the inverse of the covariance matrix of the residual vectors, i.e. $W_r = \boldsymbol{\Sigma}^{-1}_{g_r g_r}$ and $W_t = \boldsymbol{\Sigma}^{-1}_{g_t g_t}$. The covariance matrices of the residual vectors, $\boldsymbol{\Sigma}_{g_r}$ and $\boldsymbol{\Sigma}_{g_t}$, on the other hand have to be determined by applying variance propagation to Equation (4.7)–(4.8) because $\boldsymbol{g}_t$ and $\boldsymbol{g}_r$ are functions of

$\boldsymbol{r}$ and $\boldsymbol{t}$. The values of $\boldsymbol{\Sigma}_{g_r g_r}$ and $\boldsymbol{\Sigma}_{g_t g_t}$ should be

$$\boldsymbol{\Sigma}_{g_r g_r} \approx J_r \boldsymbol{\Sigma}_{rr} J_r^{\mathsf{T}} \tag{4.14a}$$

$$\boldsymbol{\Sigma}_{g_t g_t} \approx J_t \boldsymbol{\Sigma}_{tt} J_t^{\mathsf{T}} + J_r \boldsymbol{\Sigma}_{rr} J_r^{\mathsf{T}}, \tag{4.14b}$$

where $J_r \overset{\text{def}}{=} \frac{\partial \boldsymbol{g}_r}{\partial \boldsymbol{r}}$ and $J_t \overset{\text{def}}{=} \frac{\partial \boldsymbol{g}_t}{\partial \boldsymbol{t}}$ are the Jacobians, $\boldsymbol{\Sigma}_{tt}$ and $\boldsymbol{\Sigma}_{rr}$ are the covariance matrices of the measurement noise.

Once the weight matrices are determined, the minimization of Equation (4.13) can proceed and refine $\boldsymbol{x}$ iteratively. However, $\boldsymbol{r}$ *and* $\boldsymbol{t}$ *are held fixed during the whole estimation process and so are the Jacobians and the linearized model.* If there are significant errors in the measurement, they will persist and result in a degraded estimation accuracy. The degradation is confirmed by our experiment in Section 4.4 and thus motivated us look for a better approach.

4.2.3 Gauss–Helmert Model Based Solution

To tackle the aforementioned degradation problem in high noise situation, we propose to formulate the estimation problem using what is called the Gauss–Helmert model [Wolf, 1978], which makes corrections to not only the unknown parameters $\boldsymbol{x}$ but also the measurements $\boldsymbol{l}$ at each iteration.

To explain the difference between the Gauss–Helmert model and the Gauss–Markov model, let us denote the "raw" measurement as $\boldsymbol{l}^0$, its noise-free value as $\bar{\boldsymbol{l}}$ and the measurement error as $\boldsymbol{\epsilon}$, with the setting

$$\bar{\boldsymbol{l}} = \boldsymbol{l}^0 + \boldsymbol{\epsilon}. \tag{4.15}$$

Then, both the Gauss–Helmert model and the Gauss–Markov model try to obtain a *maximum likelihood estimation* of the parameter $\boldsymbol{x}$ by minimizing the weighted sum of the squared measurement errors, i.e.,

$$\underset{\boldsymbol{x},\boldsymbol{\epsilon}}{\operatorname{argmin}} \ \|\boldsymbol{\epsilon}\|_W^2 \tag{4.16}$$

assuming $\boldsymbol{\epsilon}$ follows a normal distribution and $W \overset{\text{def}}{=} \boldsymbol{\Sigma}_{ll}^{-1}$ is its inverse covariance matrix.

The key difference lies in the assumptions they make about the relation between the parameter $\boldsymbol{x}$ and the measurement $\boldsymbol{l}$. The Gauss–Markov model assumes $\boldsymbol{l}$ *is an explicit function of* $\boldsymbol{x}$, for example

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{l}, \tag{4.17}$$

while the Gauss–Helmert model, being more general, assumes an *implicit constraint* of the form:

$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{l}) = \mathbf{0}. \tag{4.18}$$

In the Gauss–Helmert model, the constraint $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{l}) = \mathbf{0}$ is strictly enforced and thus leads to a constrained optimization problem

$$\begin{aligned} &\underset{\boldsymbol{x},\boldsymbol{\epsilon}}{\operatorname{argmin}} \ \|\boldsymbol{\epsilon}\|_W^2 \\ &\quad \text{s.t. } \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{l}^0 + \boldsymbol{\epsilon}) = \mathbf{0}. \end{aligned} \tag{4.19}$$

We say the Gauss–Helmert model is more general because we can rewrite $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{l}$ as $\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{l} = \mathbf{0}$, thus Equation (4.17) is a special case of Equation (4.18). However, the explicit relation $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{l}$ makes the Gauss–Markov model more easy to solve because the optimization problem

$$\begin{aligned} &\underset{\boldsymbol{x},\boldsymbol{\epsilon}}{\operatorname{argmin}} \ \|\boldsymbol{\epsilon}\|_W^2 \\ &\quad \text{s.t. } \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{l}^0 + \boldsymbol{\epsilon}, \end{aligned} \tag{4.20}$$

can be simplified to a well-known ordinary least squares form

$$\underset{\boldsymbol{x}}{\operatorname{argmin}} \ \|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{l}^0\|_W^2. \tag{4.21}$$

Notice that in problem (4.21), we only need to make adjustments to the parameter $\boldsymbol{x}$ because the measurement $\boldsymbol{l}$ can be predicted from $\boldsymbol{f}(\boldsymbol{x})$. The Gauss–Helmert model, on the other hand, makes corrections to not only $\boldsymbol{x}$ but also $\boldsymbol{l}$ because the constraint $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{l}) = \mathbf{0}$ generally does not hold for the noisy measurements $\boldsymbol{l}^0$ even if we are provided with the true parameter $\bar{\boldsymbol{x}}$. To fulfill the implicit constraint $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{l}) = \mathbf{0}$, the measurement correction $\boldsymbol{\epsilon}$ must be estimated explicitly.

Back to our AX=XB calibration problem, it clearly does not belongs to the $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{l}$ type of problem because the parameters, $\boldsymbol{\xi}$ and O, cannot be separated from the measurements $\boldsymbol{r}$ and $\boldsymbol{t}$ in equation (4.7) and (4.8). Therefore, the optimality of the formulation (4.13), i.e. the Gauss–Markov model based solution, cannot be guaranteed. As discussed in previous section, significant errors in $\boldsymbol{r}$ and $\boldsymbol{t}$ will persist in the weight matrices as well as the linearized model, therefore leads to a degraded estimation accuracy.

A better approach to the AX=XB calibration problem should be using the Gauss–Helmert model, which use the error terms $\boldsymbol{g}_t$ and $\boldsymbol{g}_t$ not as the optimization

objective but as hard constraints instead, i.e.:

$$\begin{aligned} \underset{\boldsymbol{x},\{\boldsymbol{\epsilon}_i\}}{\operatorname{argmin}} & \sum_i^N \|\boldsymbol{\epsilon}_i\|_{W_i}^2 \\ \text{subject to} \quad & \boldsymbol{g}_t(\boldsymbol{x}, \boldsymbol{\epsilon}_i + \boldsymbol{l}_i^0) = \mathbf{0}, \\ & \boldsymbol{g}_r(\boldsymbol{x}, \boldsymbol{\epsilon}_i + \boldsymbol{l}_i^0) = \mathbf{0}, \ \forall i, \end{aligned} \tag{4.22}$$

which can be wrote more concisely as

$$\begin{aligned} \underset{\boldsymbol{x},\{\boldsymbol{\epsilon}_i\}}{\operatorname{argmin}} & \sum_i^N \|\boldsymbol{\epsilon}_i\|_{W_i}^2 \\ \text{subject to} \quad & \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\epsilon}_i + \boldsymbol{l}_i^0) = \mathbf{0}, \ \forall i, \end{aligned} \tag{4.23}$$

with $\boldsymbol{g}_r$ and $\boldsymbol{g}_t$ expressed as one constraint function $\boldsymbol{g}$ defined in Equation (4.9), and $W_i \overset{\text{def}}{=} \boldsymbol{\Sigma}_{l_i l_i}^{-1}$.

To solve a nonlinear optimization problem such as Equation (4.23), we follow a procedure similar to that of the nonlinear ordinary least squares, which iterates between linearizing the model and adjusting $(\boldsymbol{x}, \boldsymbol{l})$. To be more specific, let us assume that in the k-th iteration, we have the corrected measurements, $\boldsymbol{l}_i^k$, as well as the estimated parameters, $\boldsymbol{x}^k$, and would like to update them by

$$\boldsymbol{l}_i^{k+1} = \Delta\boldsymbol{l}_i + \boldsymbol{l}_i^k \quad \text{and} \quad \boldsymbol{x}^{k+1} = \Delta\boldsymbol{x} + \boldsymbol{x}^k.$$

We first linearize the non-linear constraint equation $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{l}) = \mathbf{0}$ around $(\boldsymbol{x}^k, \boldsymbol{l}_i^k)$ by:

$$\boldsymbol{g}(\boldsymbol{x}^k, \boldsymbol{l}_i^k) + J_i^k \Delta\boldsymbol{x} + L_i^k \Delta\boldsymbol{l}_i = \mathbf{0}, \tag{4.24}$$

with J_i^k being the Jacobians with respect to the parameters and L_i^k the Jacobians with respect to the measurements. With that, our optimization problem (4.23) becomes

$$\begin{aligned} \underset{\Delta\boldsymbol{x},\{\Delta\boldsymbol{l}_i\}}{\operatorname{argmin}} & \sum_i \|\Delta\boldsymbol{l}_i + \boldsymbol{\epsilon}_i^k\|_{W_i}^2 \\ \text{s.t.} \quad & \boldsymbol{g}(\boldsymbol{x}^k, \boldsymbol{l}_i^k) + J_i^k \Delta\boldsymbol{x} + L_i^k \Delta\boldsymbol{l}_i = \mathbf{0}, \quad \forall i, \end{aligned} \tag{4.25}$$

in which the terms $\boldsymbol{\epsilon}_i^k \overset{\text{def}}{=} \boldsymbol{l}_i^k - \boldsymbol{l}_i^0$ are the corresponding k-th iteration measurement errors.

Equation (4.25) is in essence an equality-constrained linear optimization problem, which can be converted into an unconstrained problem using the method of

Lagrange multipliers. Its solution is:

$$\Delta \boldsymbol{x} = \Big(\sum_i J_i^\mathsf{T} \Lambda_i J_i \Big)^{-1} \sum_i J_i^\mathsf{T} \Lambda_i \boldsymbol{c}_i, \tag{4.26a}$$

$$\Delta \boldsymbol{l}_i = \boldsymbol{\Sigma}_{l_i l_i} L_i^\mathsf{T} \Lambda_i (\boldsymbol{c}_i - J_i \Delta \boldsymbol{x}) - \boldsymbol{\epsilon}_i^k, \tag{4.26b}$$

$$\text{with } \Lambda_i \overset{\text{def}}{=} (L_i \boldsymbol{\Sigma}_{l_i l_i} L_i^\mathsf{T})^{-1}, \tag{4.26c}$$

$$\text{and } \quad \boldsymbol{c}_i \overset{\text{def}}{=} -\boldsymbol{g}(\boldsymbol{x}^k, \boldsymbol{l}_i^k) + L_i \boldsymbol{\epsilon}_i^k. \tag{4.26d}$$

Using Equation (4.26), we can update the estimate $(\boldsymbol{x}^{k+1}, \boldsymbol{l}_i^{k+1})$ and repeat the process until convergence.

The solution $\boldsymbol{x}$ obtained here is *best* and *unbiased*, given the measurement errors are normally distributed, see [Amiri-Simkooei, 2007]. The term best means that the solution has minimum variance compared to all other quadratic-based unbiased estimators. A detail discussion of the necessary and sufficient conditions for the existence of a unique solution for both the measurement correction as well as the estimated parameter can be found at [Neitzel and Schaffrin, 2016]. The theoretical precision of the final solution $\boldsymbol{x}$ is given by

$$\boldsymbol{\Sigma}_{xx} \overset{\text{def}}{=} \Big(\sum_i J_i^\mathsf{T} (L_i \boldsymbol{\Sigma}_{l_i l_i} L_i^\mathsf{T})^{-1} J_i \Big)^{-1}. \tag{4.27}$$

Also note that, compared to the Gauss–Markov model based approach, our Gauss–Helmert model based approach can obtain not only the extrinsic parameters X from $\boldsymbol{x}$, but also the corrected sensor trajectories $\{A\}$ and $\{B\}$ from $\boldsymbol{l}$.

4.3 Calibration with Multiple Sensors

We now extend the calibration problem to cases involving more than two sensors. As we will see in the following discussion, explicitly formulating the multi-sensor case base on our Gauss–Helmert estimation approach will give us the insight that there are actually interactions between the sensor pairs, which is a good reason for us to jointly calibrate the multiple sensors altogether instead of doing so one-by-one as separate 2-sensor cases.

First, we explicitly formulate the multi-sensor case. Let us assume there are multiple sensors rigidly attached to the robot, we index these sensors with letters $a, b, \ldots, m$ and denote their relative pose measurements as

$$\{(\boldsymbol{r}_{si}, \boldsymbol{t}_{si}) \mid s = a, \ldots, m; i = 1, \ldots, N\}. \tag{4.28}$$

Without loss of generality, we define sensor a as the base sensor and our objective is to estimate for each other sensor $s \in \{b, \ldots, m\}$ their relative rotation and

translation with respect to sensor a:

$$\{(\boldsymbol{\eta}_{as}, \boldsymbol{\xi}_{as}) \mid s = b, \ldots, m\}. \tag{4.29}$$

The collective parameter $\boldsymbol{x}$ and measurement $\boldsymbol{l}$ now become:

$$\boldsymbol{x} \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{\eta}_{ab} \\ \vdots \\ \boldsymbol{\eta}_{am} \\ \boldsymbol{\xi}_{ab} \\ \vdots \\ \boldsymbol{\xi}_{am} \end{bmatrix}, \quad \boldsymbol{l}_i \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{r}_{ai} \\ \vdots \\ \boldsymbol{r}_{mi} \\ \boldsymbol{t}_{ai} \\ \vdots \\ \boldsymbol{t}_{mi} \end{bmatrix}, \quad i = 1, \ldots, N, \tag{4.30}$$

and the constraints read

$$\boldsymbol{g}_t(\boldsymbol{x}, \boldsymbol{l}) \stackrel{\text{def}}{=} \begin{bmatrix} [R(\boldsymbol{r}_{ai}) - I_3]\boldsymbol{\xi}_{ab} + \boldsymbol{t}_{ai} - O(\boldsymbol{\eta}_{ab})\boldsymbol{t}_{bi} \\ \vdots \\ [R(\boldsymbol{r}_{ai}) - I_3]\boldsymbol{\xi}_{am} + \boldsymbol{t}_{ai} - O(\boldsymbol{\eta}_{am})\boldsymbol{t}_{mi} \end{bmatrix} = \boldsymbol{0}, \tag{4.31}$$

$$\boldsymbol{g}_r(\boldsymbol{x}, \boldsymbol{l}) \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{r}_{ai} - O(\boldsymbol{\eta}_{ab})\boldsymbol{r}_{bi} \\ \vdots \\ \boldsymbol{r}_{ai} - O(\boldsymbol{\eta}_{am})\boldsymbol{r}_{mi} \end{bmatrix} = \boldsymbol{0}. \tag{4.32}$$

For sensors that provide no rotation measurements such as GPS receiver, their constraints in $\boldsymbol{g}_r$ are simply omitted and we can still recover the rotation parameter $\boldsymbol{\eta}$ from the constraint $\boldsymbol{g}_t$. To see that, we assume the robot performs a pure translation (i.e. $R(\boldsymbol{r}) = I_3$) for example, then $\boldsymbol{g}_t$ will become $\boldsymbol{t}_{ai} - O(\boldsymbol{\eta}_{as})\boldsymbol{t}_{si}$ and has the exact form of $\boldsymbol{g}_r$, meaning the parameter O can also be estimated from the translation measurements $\boldsymbol{t}$.

Once the parameters, measurements and constraints are defined for the multi-sensor calibration case, we employ the Gauss–Helmert model based estimation approach described in Section 4.2.3 to obtain an estimate of all the extrinsic parameters as well as the corrected sensor trajectories.

4.3.1 Global Optimality

In the constraint formulation in Equation (4.31) and Equation (4.32), we connect all the sensors ($s = b, \ldots, m$) to a base sensor (a) in a star network fashion and thereby form one joint optimization problem. Despite the fact that the network is not fully connected, we claim that this does not impact optimality. Let us refer to sensor a as the *root* and sensors $s = b, \ldots, m$ as *leaves*. As shown below, we can proof that once the root-to-leaf constraints are fulfilled, all leaf-to-leaf

constraints are fulfilled automatically. Therefore, the star topology does not impact optimality.

First, assume $\mathsf{X} \in \mathrm{SE}(3)$ are the transformations correspond to the parameters $\boldsymbol{x}$ estimated by the Gauss–Helmert model, and $\mathsf{S} = \mathsf{B}, \ldots, \mathsf{M} \in \mathrm{SE}(3)$ are the relative sensor motions correspond to the corrected measurements $\boldsymbol{l}^0 + \boldsymbol{\epsilon}$. Then, the root-to-leaf constraints are fulfilled by definition after the optimization, i.e. for all sensors $s = b, \ldots, m$ holds:

$$\mathsf{A}\mathsf{X}_{as} = \mathsf{X}_{as}\mathsf{S}. \tag{4.33}$$

Without loss of generality, let us consider the leaf nodes m and b as example. If we set

$$\mathsf{X}_{bm} = \mathsf{X}_{ab}^{-1}\mathsf{X}_{am} \tag{4.34}$$

being the transformation between b and m given the transformations X_{ab} and X_{am}, then we can prove that following equations are fulfilled:

$$\mathsf{B}\mathsf{X}_{bm} = \mathsf{X}_{bm}\mathsf{M}. \tag{4.35}$$

We start from Equation (4.33) and set s to b and m respectively to obtain

$$\mathsf{X}_{ab}^{-1}\mathsf{A} = \mathsf{B}\mathsf{X}_{ab}^{-1}, \tag{4.36}$$

$$\mathsf{A}\mathsf{X}_{am} = \mathsf{X}_{am}\mathsf{M}. \tag{4.37}$$

Then, the left hand side of Equation (4.35) is

$$\begin{aligned} \mathsf{B}\mathsf{X}_{bm} &= \mathsf{B}[\mathsf{X}_{ab}^{-1}\mathsf{X}_{am}] && \text{(4.38a)} \\ &= [\mathsf{X}_{ab}^{-1}\mathsf{A}]\mathsf{X}_{am} && \text{(4.38b)} \\ &= \mathsf{X}_{ab}^{-1}[\mathsf{X}_{am}\mathsf{M}] && \text{(4.38c)} \\ &= \mathsf{X}_{bm}\mathsf{M}. && \text{(4.38d)} \end{aligned}$$

Thus, Equation (4.35) has been proven. As a result, the fulfillment of the root-to-leaf constraints automatically satisfies all leaf-to-leaf constraints and thus they do not need to be modeled explicitly, meaning the star topology is enough to ensure global optimality.

4.3.2 Advantages of Joint Calibration

Note that Equation (4.32) not only provides the information to estimate the rotation $\boldsymbol{\eta}$, but also an inter-sensor constraint on the rotation measurements between sensors, due to:

$$\theta_i \overset{\text{def}}{=} \|\boldsymbol{r}_{ai}\| = \|\mathsf{O}_{ab}\boldsymbol{r}_{bi}\| = \|\boldsymbol{r}_{bi}\| = \cdots = \|\boldsymbol{r}_{mi}\|. \tag{4.39}$$

It means that rotations within the same time step i will be corrected to have the same magnitude θ_i, using a weighted average of all the rotation measurements. Recall that in Equation (3.21) of Section 3.3.2, the rotation uncertainty/error σ_θ has a significant influence over the result uncertainty $\sigma_{\|\boldsymbol{\xi}\|}$. If we can drive σ_θ towards zero, than we can drastically improve the accuracy of $\boldsymbol{\xi}$. For instance, the same calculation of $\sigma_{\|\boldsymbol{\xi}\|}$ in Section 3.3.2 will be only 11 cm instead of 23 cm when σ_θ reduces from 1° to $(10^{-4})^\circ$.

This provide two valuable insights into how to improve the overall calibration accuracy. First, we should jointly estimate the extrinsic parameters altogether instead of estimating them one-by-one as separate 2-sensor cases. Because the more sensors are involved, the better sensor trajectories we can recover (e.g., lower σ_θ) and thus better calibration result (e.g., lower $\sigma_{\|\boldsymbol{\xi}\|}$). Second, base on the first point, we can even introduce temporary devices with high pose measurement accuracy to further reduce the trajectory measurement uncertainties during the calibration, hence improving the calibration accuracy.

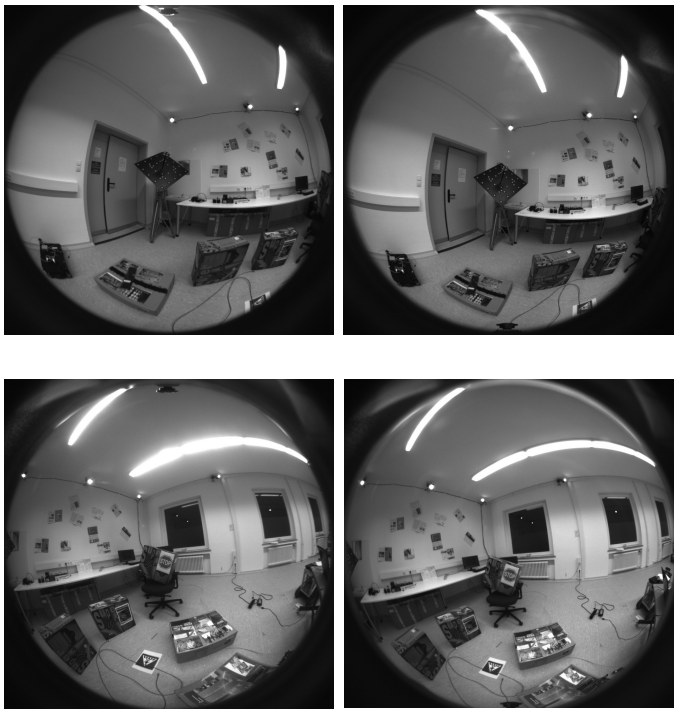
4.4 Experimental Results

The experiments are designed to show the capabilities of our Gauss–Helmert model based calibration method and to support our key claims. We claim that our approach (i) accurately determines the extrinsic calibration parameters, (ii) can cope with a noisy initial guess obtained through a direct approach, (iii) provides more accurate results compared to ordinary least squares using the Gauss–Markov model, and (iv) can be executed in a reasonable amount of time hence is useful for real world applications. We perform the evaluations on own real-world as well as simulated datasets to support these claims.

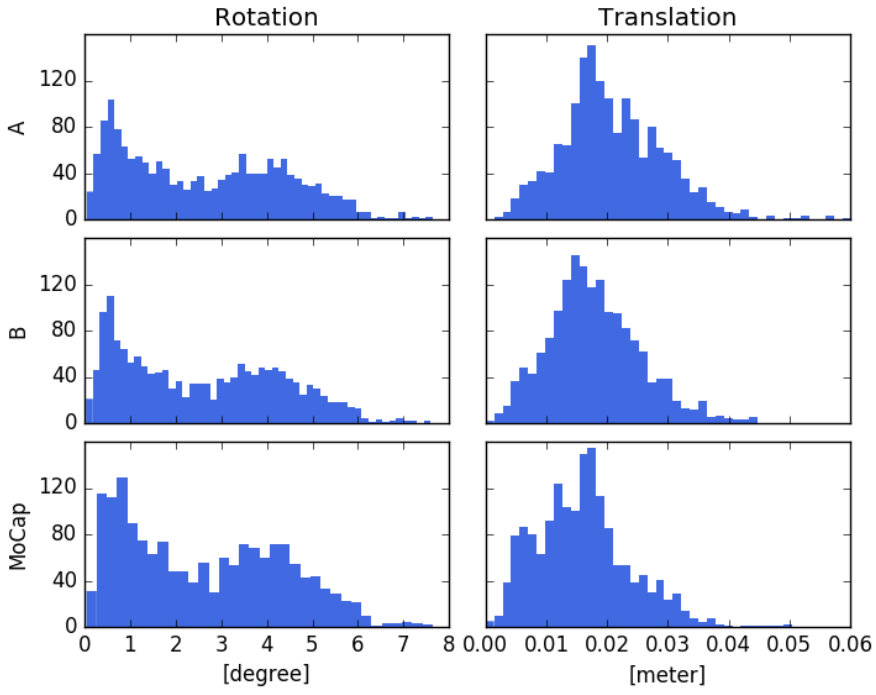
4.4.1 Real World Data

We first performed the calibration on the real world data. For the real world setup, we use a quadcopter with two stereo camera pairs (called A and B later on), one pointing forward and one pointing backwards. Figure 4.1a depicted example images of this calibration data. In addition to the stereo camera pairs, we place markers of a motion capture system on the quadcopter to simulate an additional sensor that has to be calibrated.

We recorded a total of $N = 1655$ relative motion-segments estimated using our own visual odometry pipeline [J. Schneider et al., 2016]. Around 25 of the segments are outliers as the visual odometry lost track. The maximum rotation angle



(a)



(b)

Figure 4.1: (a) Example stereo images used for calibration. (b) Histogram of the rotation and translation magnitude of our real world dataset. Left column shows the rotation ($\|\boldsymbol{r}\|$, in degree), and right column shows the translation ($\|\boldsymbol{t}\|$, in meters). From top to bottom, each row represents stereo pair A, pair B and the motion capture studio respectively.

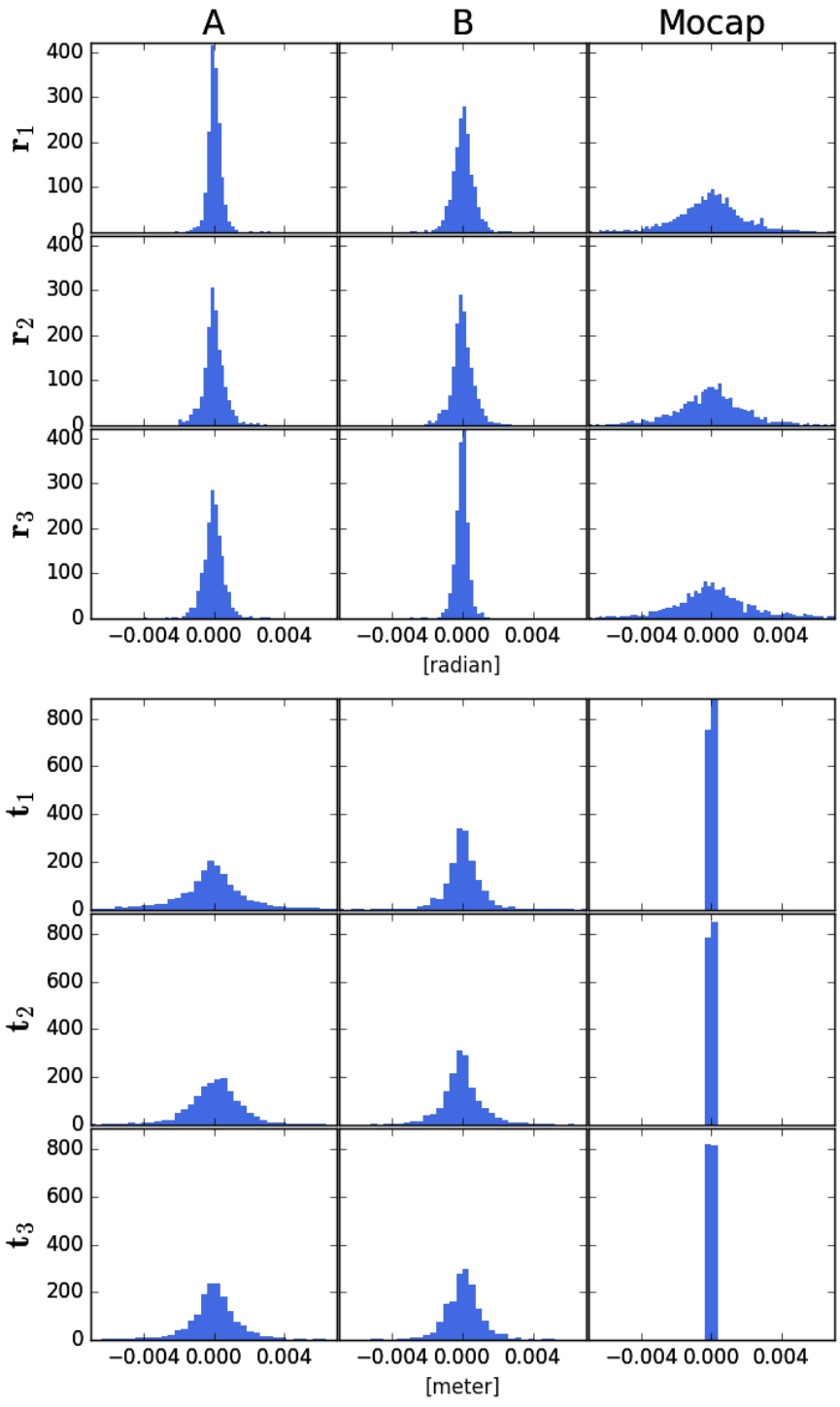


Figure 4.2: Distribution of measurement residuals $\boldsymbol{\epsilon}$ after the refinement by our approach. Columns from left to right are for stereo A, B and Mocap respectively. Rows from top to bottom are the 6 measurement components, namely $\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3$ for rotation and $\boldsymbol{t}_1, \boldsymbol{t}_2, \boldsymbol{t}_3$ for translation. It is clear that the Gaussian noise assumption holds for our real world dataset to a large extent, and our constraint model is correct since there is no bias.

Table 4.1: Standard deviation of the measurement noise

	σ_r [°]	σ_t [mm]
Stereo pair A	0.0286	2
Stereo pair B	0.0286	3
Mocap	0.573	0.2

Table 4.2: Precision of the estimated parameters $\boldsymbol{\xi}, \boldsymbol{\eta}$

	$\sigma_{\boldsymbol{\xi}_1}$	$\sigma_{\boldsymbol{\xi}_2}$	$\sigma_{\boldsymbol{\xi}_3}$	$\sigma_{\boldsymbol{\eta}_1}$	$\sigma_{\boldsymbol{\eta}_2}$	$\sigma_{\boldsymbol{\eta}_3}$
Stereo B	2.8 mm	1.90 mm	2.8 mm	0.041°	0.032°	0.033°
Mocap	1.43 mm	1.97 mm	1.13 mm	0.165°	0.191°	0.202°

of the inlier data is 7.6° and the data is recorded at 20 fps. Figure 4.1b depicts a histogram of the rotation angle and translation magnitude of the gathered data.

We first obtain an initial guess using the SVD-based direct method (i.e. Section 4.2.1) then run our approach to obtain an improved solution as well as its theoretical precision. The covariance matrices are heuristically set to $\boldsymbol{\Sigma}_{rr} = \sigma_r I_3$ and $\boldsymbol{\Sigma}_{tt} = \sigma_t I_3$ with σ_r, σ_t given in Table 4.1. We also assume the measurement noise are independent and identically distributed random variables. Our approach converged at the 4th iterations. The theoretical precision given by our approach is depicted in Table 4.2. They are the square roots of the main diagonal of the covariance matrix in Equation (4.27).

As this is a real world experiment, the ground truth is not available, so we cannot make an ground truth comparison. But judging from the measurement residual distribution (shown in Figure 4.2) given by our model after the estimation, it is clear that i) the Gaussian noise assumption holds for our dataset to a large extent, ii) our constraint model is correct and there is no bias in the estimation, otherwise the residual histograms will not be symmetrically centered around zero. Therefore we have a good reason to believe that the theoretical precision given by our approach is plausible.

4.4.2 Accuracy Comparison With Simulated Data

To provide a more quantitative experiments, we performed the analysis of the accuracy in simulation, which is the second experiment.

We generated in total 30,000 experiments (1000 per noise level), with noise levels starting with the values shown in Table 4.1 and scaled them with a factor varying from 1 to 30. With a factor of 30, the rotation error of the motion measurement

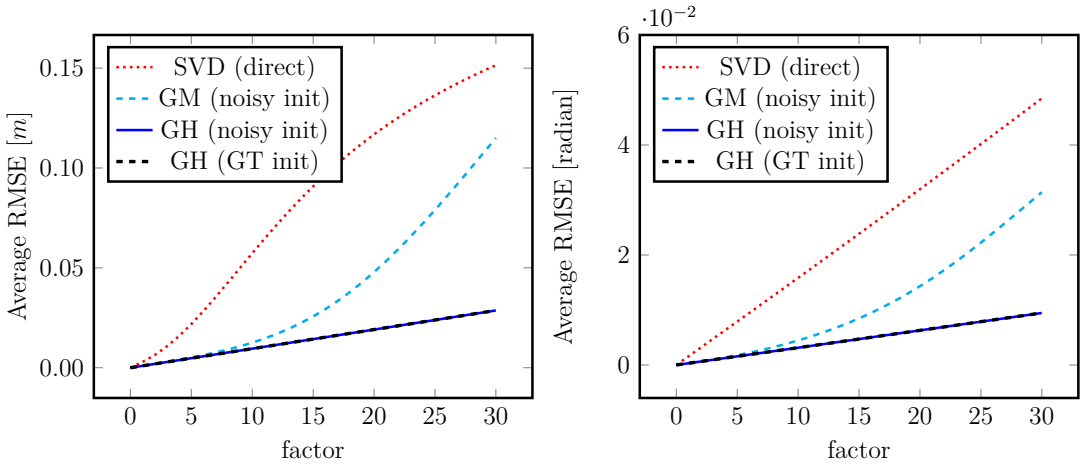


Figure 4.3: Accuracy comparison through Monte-Carlo simulation. The x-axis show the factor by which we scale the input noise for the translational (upper) and rotational (lower) component. The plots show the RMSE for the direct SVD approach, the GM as well as GH model using a noisy initialization, and finally the GH model initialized with the ground truth as the initial guess. The GH model outperforms the other approaches as expected and performs identical if initialized with noisy values or the ground truth ones.

can be as much as $\pm 25°$ according to the 3σ principle, which is quite high compared to real motion sensors.

This experiment is designed to compare the accuracy of: (i) the SVD-based direct solution (called SVD), (ii) least squares estimator based on the Gauss–Markov model (called GM), and (iii) our Gauss–Helmert based approach (called GH). Both GH and GM are using Equation (4.31) and Equation (4.32) as constraint function (or residual vectors). The initial guess to GH and GM are ground-truth values perturbed by uniform additive noise. For reasons of comparison, we also provided a ground truth initialization. The metric for comparison is the averaged Root-Mean-Square-Error (RMSE) of the estimated 6 rotation parameters and 6 translation parameters from 1,000 trials.

The result of the simulation is depicted in Figure 4.3. From this plot, we can draw several conclusions. First, the GH and GM perform always better than SVD. The only advantage of the SVD is that it is a direct solution and requires no initial guess. Second, the GH and GM approaches produce identical results as long as the noise-level is small. Third, as the noise level increases, the performance of the GM solution degrades while our GH solution does not. The error of the GH solution grows linearly with the increased input noise, which is the expected theoretical result. Thus, we can conclude that over the spectrum of all situations, our GH approach performs best. At the highest noise-levels, GH is better than GM by 75% in translation (0.0287 vs. 0.1150), and by 71% in rotation (0.0095 vs.

0.0314) in this simulation setup. Only for situation with low noise (e.g., a factor smaller than three), GH and GM show the same performance as can be expected given our explanation in Section 4.2.2. Finally, we can see that the GH approach produces the same results no matter we use the ground truth for initialization or a noisy variant of it.

4.4.3 Radius of Convergence

As our approach is in essence an iterative nonlinear optimization, we cannot guarantee that it will always find the global minimum given arbitrary initial guess. But in reality, we can obtain an initial guess either by manual or by utilizing a direct approach. So a more practical question is to ask whether the radius of convergence of our approach is large enough to converge to the global optimum given an initial values provided by a direct approach. Here we refer to the global optimum as the solution generated when using the ground truth as initial guess.

So we performed a similar experiment as before but with the initial guess coming from the SVD method, i.e. without any knowledge about the true configuration. The results are depicted in Figure 4.4 and show that the RMSE curves are identical for both initializations, despite the fact that the SVD solutions could deviate from the truth as much as 0.15 m. Hence we conclude that our approach is robust enough to be used in combination with SVD for providing the start value for the optimization.

4.4.4 Runtime

For our real world calibration, we considered 1,630 poses extracted from each of the three sensors. The timings of our Python code running on a i5 notebook computer are approx. 6 s for the overall approach. Around 600 ms is used for computing the initial solution using SVD and around 1.2 s-1.5 s is required per iteration. These timing involves all computations, except the motion estimation from the sensors itself, in this case the visual odometry. At least in our setup, computing the visual odometry takes longer than the calibration itself. So the computation requirements of our approach is acceptable.

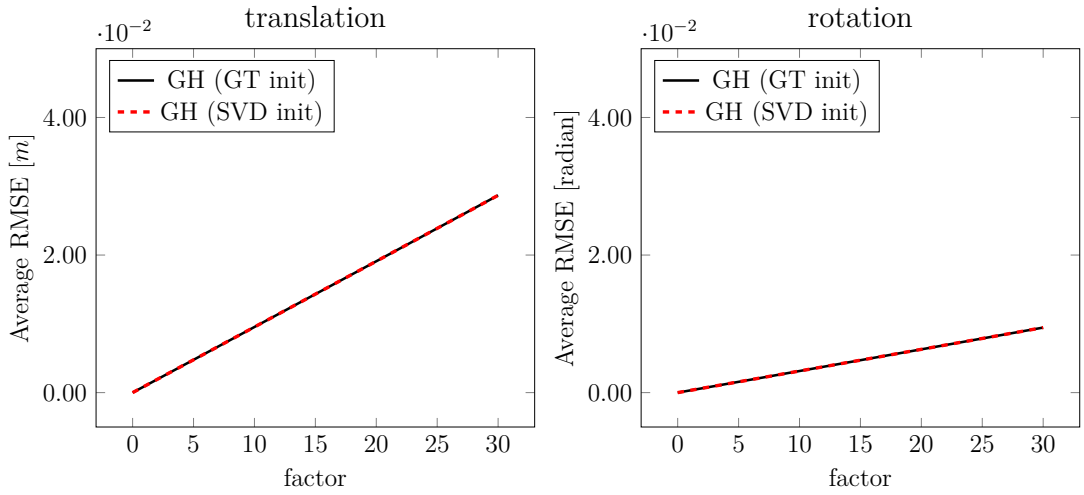


Figure 4.4: Accuracy in relation to the initial guess for different noise levels on the translational and rotational component. In all cases, the initialized via the SVD result or with the ground truth produces the same results. Thus, we conclude that we can safely use SVD to initialize our GH optimization.

4.5 Summary

In this chapter, we considered the estimation problem of the AX=XB calibration model. We pointed out an overlooked defect of traditional least squares estimation approach in this context and presented a novel approach to tackle to the problem. Our approach computes the extrinsic parameters in a statistically sound way using the Gauss–Helmert model. This allows us to successfully determine the relative transformations between the origins of the sensor coordinate systems and the robot's base. We implemented and evaluated our approach in simulations as well as on real data. The experiments suggest that our approach can accurately determine the extrinsic parameters of the individual sensors under realistic conditions. We provided comparisons to a direct SVD approach as well as to the ordinary Gauss–Markov least squares estimation and furthermore supported all claims made in this chapter through our evaluation.

Part II

Camera-Laser Data Fusion For Ego-motion Estimation

Chapter 5

Joint Ego-motion Estimation Through Relative Orientation Estimation and 1-DoF ICP

Using cameras or laser scanners to perform ego-motion estimation is a common practice in robotics research [Besl and McKay, 1992; Engel et al., 2018; Engel et al., 2014; Klein and Murray, 2007; Mur-Artal et al., 2015; Z. Zhang, 1994]. But when comparing systems using cameras to those using lasers, we often see that cameras are slightly better in estimating the angular components, i.e., the rotation of the movement, whereas laser scanners are superior for estimating the translation and for obtaining 3D points. Furthermore, cameras provide dense color information, which can simplify the data association using feature correspondences. Thus, coupling laser scanners and cameras can yield advantages. In this chapter, we begin the discussion on the second topic of the thesis, focusing on camera-laser fusion algorithms for ego-motion estimation problem. We assume a laser scanner and a monocular camera are presented as well as calibrated in the sensor system.

We will present a novel method to indirectly fuse the laser range data and camera images to jointly estimate the frame-to-frame motion of a mobile platform. It exploits image information to guide the ICP-based laser scan-matching process such that it is able to improve the motion estimation accuracy and yet does not require an initial guess of the motion parameters, nor a field-of-view overlap between the camera and the laser scanner. Our method first estimates the five degrees of freedom (DoF) relative orientation from image pairs through feature point correspondences (as described in Section 5.2), and then, utilizes such information to guide both the transformation-estimation and data-association

steps in the ICP-based laser scan-matching process: i) in the transformation-estimation step, we formulate the remaining scale estimation problem as a variant of the ICP problem with only one degree of freedom (as described in Section 5.3), ii) in the data-association step, we utilize the relative orientation information to constrain the data association direction between laser point clouds (as described in Section 5.4), thus significantly reduces the effort for finding the matches and allows the ICP to converge faster. We implemented our approach and evaluated it using KITTI data (in Section 5.5). The result shows that our approach provides accurate trajectory estimates, which are better than those of each sensing modality alone.

5.1 ICP Based Laser Scan-Matching

Estimating the ego-motion using a 2D or 3D laser scanner through point cloud alignment is often referred to as *scan-matching*. Scans are either matched pairwise or with respect to a local or global map in order to compute the relative transformation between the robot's poses at the different points in time. Popular approaches for that are the iterative closest point (ICP) algorithm [Besl and McKay, 1992; Z. Zhang, 1994] (as described in Section 2.3) and its variants, such as [Segal et al., 2009; Serafin and Grisetti, 2015], or correlative scan matching [Olson, 2009].

We follow the notation in Section 2.3 and refer to the point cloud $\{\boldsymbol{a}_i \in \mathbb{R}^3\}_{i=1}^{M}$ as the *previous* point cloud, and $\{\boldsymbol{b}_j \in \mathbb{R}^3\}_{j=1}^{N}$ as the *current* point cloud. Assume the two point clouds are generated from two consecutive laser scans, then our task is to determine the relative rotation $R \in \mathrm{SO3}$ and translation $\boldsymbol{t} \in \mathbb{R}^3$ between the two scanning locations, by registering the two point clouds.

As described in Section 2.3, since both the point correspondences and the transformation are unknown, the ICP algorithm iterates between a data-association step:

$$\boldsymbol{b}'_i \overset{\text{def}}{=} \underset{\boldsymbol{b}_j \in \{\boldsymbol{b}\}}{\operatorname{argmin}} \left\| R\boldsymbol{a}_i + \boldsymbol{t} - \boldsymbol{b}_j \right\|^2, \tag{5.1}$$

and a transformation-estimation step:

$$\underset{R,\boldsymbol{t}}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| R\boldsymbol{a}_i + \boldsymbol{t} - \boldsymbol{b}'_i \right\|^2. \tag{5.2}$$

The ICP algorithm is intuitive and powerful. However, it is in essence a greedy algorithm. The quality of the transformation parameters and the final correspondence highly depends on the initial correspondence.

Figure 5.1: 2D-to-2D corresponding image points.

5.2 Relative Orientation of the Image Pair

To mitigate the problem of standard ICP, we exploit the image information to obtain a partial estimate of the transformation, using the relative orientation derived from the images of consecutive time steps.

In a monocular camera setup, we can estimate five out of the six degrees of freedom of the transformation between camera viewpoints, purely based on image point correspondences (e.g. as shown in Figure 5.1). The five parameters consists of three parameters for rotation R and two parameters for translation direction, denoted as $\boldsymbol{t}_{\text{dir}} \in S^2$ (because the scale, which is the length of the translation $\boldsymbol{t}$, cannot be determined and thus one uses $\|\boldsymbol{t}_{\text{dir}}\| = 1$). This set of five parameters is often referred to as the *relative orientation* of the image pair, and it can be estimated by exploiting the coplanarity constraint:

$$\mathbf{x}_i^{\mathsf{T}} E \mathbf{x}'_i = 0, \tag{5.3}$$

where $\mathbf{x}_i, \mathbf{x}'_i$ are the 2D image coordinates of a corresponding point pairs, and E is the so-called essential matrix, from which the orientation parameters can be extracted.

Various direct solutions for computing the essential matrix E exist. We use Nistér's five-point algorithm [Nistér, 2004] and SIFT features together with a standard RANSAC procedure. The relative orientation parameters $(R^0, \boldsymbol{t}_{\text{dir}})$ are extracted from the essential matrix E and verified by standard checks such as the fact that triangulated image feature points must lie in front of the camera. Special cases such as zero translation are also handled.

5.3 1-DoF ICP for Scale Estimate

The key idea of our approach is to first estimate the relative orientation from the image pair, and then formulate the remaining scale estimation problem as a variant of the ICP problem with only one degree of freedom. The simplified ICP problem possesses several attractive properties.

Given the relative orientation $(R^0, \boldsymbol{t}_{\text{dir}})$ computed from the image pair, the metric scale of the translation $\|\boldsymbol{t}_{\text{true}}\|$ is unknown. We denote the unknown scale parameter as s and express the scale through the translation vector between the two poses as

$$\boldsymbol{t}_{\text{true}} = s\,\boldsymbol{t}_{\text{dir}}, \quad s \in [0, \infty). \tag{5.4}$$

To estimate s, we propose to solve a novel variant of the ICP problem with only one degree of freedom, which can be expressed through

$$s = \underset{s \geq 0}{\operatorname{argmin}} \sum_i \|R^0 \boldsymbol{a}_i + s\boldsymbol{t}_{\text{dir}} - \boldsymbol{b}'_i\|^2 \tag{5.5}$$

$$\text{or} \quad s = \underset{s \geq 0}{\operatorname{argmin}} \sum_i \left|\boldsymbol{n}_i^\mathsf{T}(R^0 \boldsymbol{a}_i + s\boldsymbol{t}_{\text{dir}} - \boldsymbol{b}'_i)\right|^2, \tag{5.6}$$

for the point-to-point and point-to-plane cost function respectively. Efficient closed form solution can be derived for both equations. To solve Equation (5.5), we define $\boldsymbol{e}_i \overset{\text{def}}{=} R^0 \boldsymbol{a}_i - \boldsymbol{b}'_i$ and obtain:

$$\Phi(s) \overset{\text{def}}{=} \sum_i \|s\boldsymbol{t}_{\text{dir}} + R^0 \boldsymbol{a}_i - \boldsymbol{b}'_i\|^2 \tag{5.7}$$

$$= \sum_i \|s\boldsymbol{t}_{\text{dir}} + \boldsymbol{e}_i\|^2 \tag{5.8}$$

$$= \sum_i s^2 + 2s\boldsymbol{e}_i^\mathsf{T}\boldsymbol{t}_{\text{dir}} + \boldsymbol{e}_i^\mathsf{T}\boldsymbol{e}_i. \tag{5.9}$$

By setting $\frac{\partial \Phi}{\partial s} = 0$, we obtain $\sum_i s + \boldsymbol{e}_i^\mathsf{T}\boldsymbol{t}_{\text{dir}} = 0$ and thus

$$s_{\text{new}} = -\frac{1}{N}\sum_i \boldsymbol{e}_i^\mathsf{T}\boldsymbol{t}_{\text{dir}}, \tag{5.10}$$

where N is total number of matched point pairs.

Similarly, for point-to-plane distances according to Equation (5.6), we define $w_i \overset{\text{def}}{=} \boldsymbol{n}_i^\mathsf{T}\boldsymbol{t}_{\text{dir}}$ and obtain

$$\Phi(s) \overset{\text{def}}{=} \sum_i \left|\boldsymbol{n}_i^\mathsf{T}(R^0 \boldsymbol{a}_i + s\boldsymbol{t}_{\text{dir}} - \boldsymbol{b}'_i)\right|^2 \tag{5.11}$$

$$= \sum_i \left|s\boldsymbol{n}_i^\mathsf{T}\boldsymbol{t}_{\text{dir}} + \boldsymbol{n}_i^\mathsf{T}(R^0 \boldsymbol{a}_i - \boldsymbol{b}'_i)\right|^2 \tag{5.12}$$

$$= \sum_i \left|sw_i + \boldsymbol{n}_i^\mathsf{T}\boldsymbol{e}_i\right|^2. \tag{5.13}$$

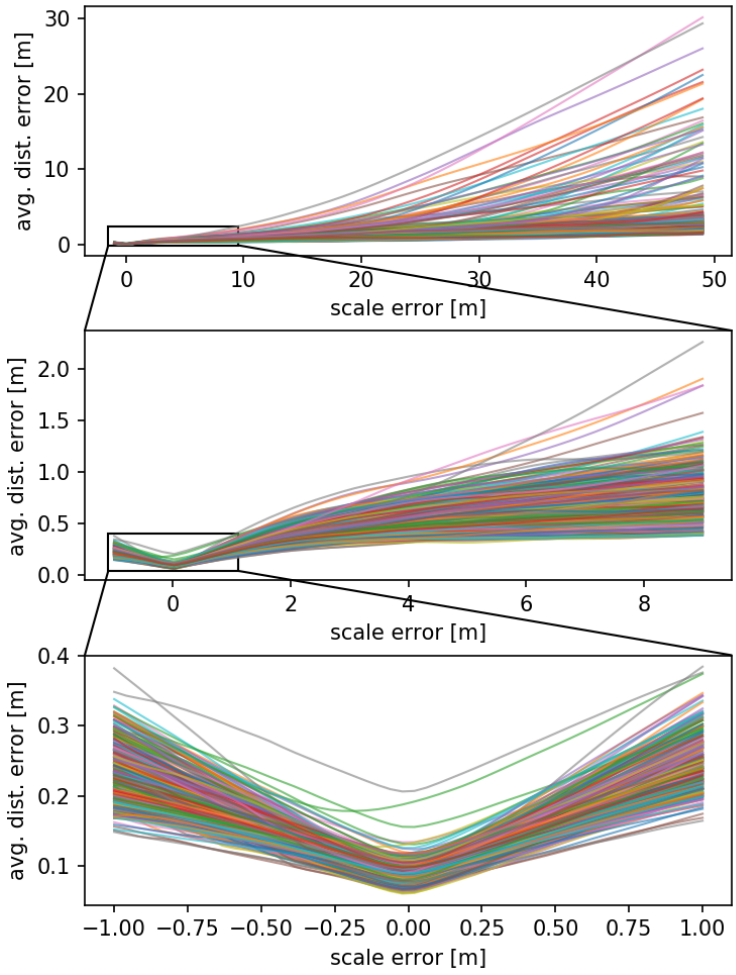


Figure 5.2: 1-DoF ICP point-to-point cost function evaluated on the KITTI dataset. The x axis shows the scale parameter deviation from the ground truth (0) and the y axis shows the averaged point matching error distance (Equation (5.7)). The function reveals a smooth surface and appears to be, at least in all our experiments, mostly convex.

Setting $\frac{\partial \Phi}{\partial s} = 0$ leads to $\sum_i s w_i^2 + w_i \boldsymbol{n}_i^\mathsf{T} \boldsymbol{e}_i = 0$ and thus

$$s_{\text{new}} = -\frac{\sum_i w_i \boldsymbol{n}_i^\mathsf{T} \boldsymbol{e}_i}{\sum_i w_i^2}. \tag{5.14}$$

This one degree of freedom ICP problem possesses several attractive properties. First and foremost, in all our analyzed cases, the cost function has a well distinguishable global minimum, especially in feature-rich environments. Consider the KITTI [Geiger et al., 2012a] dataset "odometry sequence 00" as an example. Figure 5.2 shows a plot of the point-to-point cost function in Equation (5.7) evaluated over keyframes covering the whole scene. Although the cost function depends on the scene structure, we found that the curve is smooth, appears to

be mostly convex, and that the global minimum represents the true scale well. Besides from that, the solution space of the scale parameter can also be easily bounded if additional information is available, as the scale parameter represents the physical straight line distance that the vehicle has traveled.

Based on the above observation, we propose a two-step approach to efficiently solve the 1-DoF ICP instead of a standard ICP implementation: 1D grid search followed by an iterative refinement. The first step is a grid search procedure that operates in a branch and bounce fashion in order to locate the basin of the global minimum. Given a search boundaries $[s_{\min}, s_{\max}]$, we generate a small number of linearly spaced hypotheses and evaluate the cost (e.g., Equation (5.7) or Equation (5.11)) for each hypothesis. We then select two of the hypotheses with minimum cost as the new search boundaries and repeat the process. In this way, the solution space can be drastically reduced in just a few iterations and it has a higher chance to avoid shallow local minima that may leads to registration failures. If a prior, s^0, exists, for example from wheel encoders, we can also incorporate it into the the grid search as an extra hypothesis in the first iteration. After the grid search has been performed, we then iteratively refine the grid search solution s using Equation (5.10) (or Equation (5.14)) in an ICP fashion. This time, the ICP algorithm is likely to reach a global optimum because of the grid search bootstrapping.

The algorithmic view of our method is given in Algorithm 2. It is worth pointing out that in the point-to-plane version, we can safely discard any points of the current scan that has a normal vector perpendicular to $\boldsymbol{t}_{\text{dir}}$, without compromising the solution. Because $\boldsymbol{n}_i^\mathsf{T}\boldsymbol{t}_{\text{dir}} = w_i = 0$ implies that these points have no influence on the solution. Such a filtering can greatly reduce the computational effort and is a big advantage for real-time processing. For example, in the KITTI dataset sequence 00, we can remove up to 40% of the scan points because they lie on the ground plane and do not contribute to the scale estimate.

After convergence, we execute one final ICP step that corrects all three translational parameters (but not the rotation parameters nor changes the data association). We observed that such operation leads to slightly better results for the estimated trajectory in the end.

Algorithm 2 1-DoF ICP for scale estimate

Input:
- Previous point cloud $\boldsymbol{a}$, current point cloud $\boldsymbol{b}$
- Relative orientation $R^0, \boldsymbol{t}_{\text{dir}}$
- Initial search boundary $s_{\min}, s_{\max}$
- Initial guess s^0

Parameter:
- Number of hypotheses per iteration $n \in [3, \infty)$
- Outlier distance threshold d_{out}

Output: Estimated scale s.

▷ **Step 1: Grid Search**

repeat
  Hypothesis $\{s_1, \cdots, s_n\} \leftarrow \text{linspace}(s_{\min}, s_{\max}, n)$;
  if first iteration **then** $s_{n+1} \leftarrow s^0$;
  for $s \in \{s_1, \cdots, s_{n+1}\}$ **do**
    Transform the previous cloud $\boldsymbol{a}' \leftarrow R^0\boldsymbol{a} + s\boldsymbol{t}_{\text{dir}}$;
    Match current cloud $\boldsymbol{b}' \leftarrow \text{argmin}_{\boldsymbol{b}} \|\boldsymbol{a}' - \boldsymbol{b}\|$;
    Calculate cost $C(s) \leftarrow \sum_{\boldsymbol{b}'} \|\boldsymbol{a}' - \boldsymbol{b}'\|$;
  Update $s_{\min}, s_{\max} \leftarrow$ the two s with lowest cost.
until converge **or** maximum iterations reached
$s \leftarrow \text{argmin}_{\{s_{\min}, s_{\max}\}} C(s)$

▷ **Step 2: Refinement**

repeat
  Transform previous cloud $\boldsymbol{a}' \leftarrow R^0\boldsymbol{a} + s\boldsymbol{t}_{\text{dir}}$;
  Match current cloud $\boldsymbol{b}' \leftarrow \text{argmin}_{\boldsymbol{b}} \|\boldsymbol{a}' - \boldsymbol{b}'\|$;
  Remove point pairs whose distance exceeded d_{out};
  if using point-to-point **then**
    Update $s \leftarrow -\frac{1}{N}\sum_i \boldsymbol{e}_i^\mathsf{T}\boldsymbol{t}_{\text{dir}}$ ▷ from Equation (5.10)
  else // using point-to-plane
    Update $s \leftarrow -\frac{\sum_i w_i \boldsymbol{n}_i^\mathsf{T}\boldsymbol{e}_i}{\sum_i w_i^2}$ ▷ from Equation (5.14)
until converge **or** maximum iterations reached
return s

5.4 Relative Orientation Constrained Data Association

Besides reducing the degree of freedom of the ICP problem from six to one, the relative orientation can also be used to guide the data association that has to take place in all ICP iterations. The key observation here is that the previous frame must be located in the direction $\boldsymbol{t}_{\text{dir}}$ with respect to the current one, the same must hold for the corresponding points, see Figure 5.3 for an illustration. Therefore, for a previous point $\boldsymbol{a}_i$, we can restrict its matching candidates $\boldsymbol{b}'_i$ to be located near to the ray $\boldsymbol{r}_i = R^0\boldsymbol{a}_i + \lambda\boldsymbol{t}_{\text{dir}}$, instead of arbitrary points in the whole current point cloud. Ideally, the point $\boldsymbol{b}'_i$ should lie exactly on the ray, but due to noise, we relax the constraint and allow the candidate point to slightly deviate from the ray.

To achieve this, we propose a modified closest point association procedure as listed in Algorithm 3. The main idea is to use a temporary coordinate system with $\boldsymbol{t}_{\text{dir}}$ being the X-axis and the current frame's origin being the origin of that frame. Any point correspondences that are inconsistent with the direction $\boldsymbol{t}_{\text{dir}}$ will have nonzero Y and Z components in its error vector in this temporary frame. Thus, we can define a weighted Euclidean distance metric, which heavily punishes the Y and Z components in this frame, i.e.,

$$d^2(\boldsymbol{a}'_i, \boldsymbol{b}_j) \overset{\text{def}}{=} (Q\boldsymbol{a}'_i - Q\boldsymbol{b}_j)^\mathsf{T} \begin{bmatrix} 1 & & \\ & \gamma & \\ & & \gamma \end{bmatrix} (Q\boldsymbol{a}'_i - Q\boldsymbol{b}_j) \tag{5.15}$$

$$= (\boldsymbol{a}'_i - \boldsymbol{b}_j)^\mathsf{T} Q^\mathsf{T} \begin{bmatrix} 1 & & \\ & \gamma & \\ & & \gamma \end{bmatrix} Q(\boldsymbol{a}'_i - \boldsymbol{b}_j) \tag{5.16}$$

$$\overset{\text{def}}{=} (\boldsymbol{a}'_i - \boldsymbol{b}_j)^\mathsf{T} W(\boldsymbol{a}'_i - \boldsymbol{b}_j), \tag{5.17}$$

where $\boldsymbol{a}'_i = R^0\boldsymbol{a}_i + s\boldsymbol{t}_{\text{dir}}$, $\gamma \gg 1$ is the penalty weight for the Y and Z components, and Q is a rotation matrix, which is used to transform the points $\boldsymbol{a}'_i$ and $\boldsymbol{b}_j$ from the current frame into the new temporary frame.

The rotation matrix Q depends on vector $\boldsymbol{t}_{\text{dir}}$ and can be generated by applying QR decomposition to $\boldsymbol{t}_{\text{dir}}$. The orthonormal matrix of the QR decomposition result is used as Q after transpose. The rows of Q consist of $\boldsymbol{t}_{\text{dir}}$ and two orthogonal complements of $\boldsymbol{t}_{\text{dir}}$ in $\mathbb{R}^3$, i.e., $Q = \begin{bmatrix} \boldsymbol{t}_{\text{dir}} & \boldsymbol{v}_1 & \boldsymbol{v}_2 \end{bmatrix}^\mathsf{T}$ and $\boldsymbol{v}_i \perp \boldsymbol{t}_{\text{dir}}, i = 1, 2$.

The proposed distance metric can be used in a standard k-d tree algorithm with

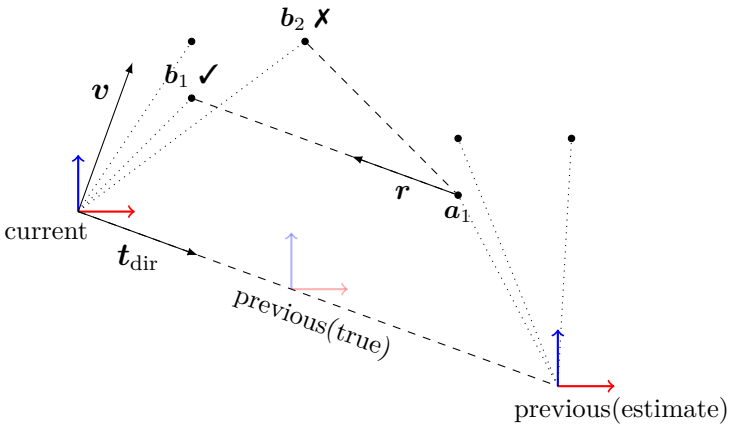


Figure 5.3: The relative orientation supports point cloud data association. Given $R^0, \boldsymbol{t}_{\text{dir}}$, point $\boldsymbol{a}_1$ is matched to $\boldsymbol{b}_1$ instead of $\boldsymbol{b}_2$ (which is although closer to $\boldsymbol{a}_1$), because point $\boldsymbol{b}_1$ is lying on the ray $\boldsymbol{r}(\lambda) = R^0\boldsymbol{a}_1 + \lambda\boldsymbol{t}_{\text{dir}}$. Thus, several wrong associations can be excluded.

Algorithm 3 Constrained Data Association

1: **Input:**
- Previous point cloud $\boldsymbol{a}$, current point cloud $\boldsymbol{b}$;
- Relative orientation and scale $R^0, \boldsymbol{t}_{\text{dir}}, s$;

2: **Parameter:**
- Penalty weight $\gamma \gg 1$;
- Outlier distance threshold d_{out};

3: **Output:** Matched point pairs $(\boldsymbol{m}, \boldsymbol{b})$.

4: Calculate QR decomposition: $QR = \boldsymbol{t}_{\text{dir}}$;

5: Weight matrix $W \leftarrow Q \begin{bmatrix} 1 & & \\ & \gamma & \\ & & \gamma \end{bmatrix} Q^\mathsf{T}$;

6: Transform previous points $\boldsymbol{a}' \leftarrow R^0\boldsymbol{a} + s\boldsymbol{t}_{\text{dir}}$;

7: Match previous points $\boldsymbol{b}' \leftarrow \text{argmin}_{\boldsymbol{a}}(\boldsymbol{a}' - \boldsymbol{b})^\mathsf{T} W (\boldsymbol{a}' - \boldsymbol{b})$;

8: Remove point pairs with $\|\boldsymbol{a}' - \boldsymbol{b}'\| > d_{\text{out}}$;

9: **return** point correspondences $(\boldsymbol{a}, \boldsymbol{b}')$

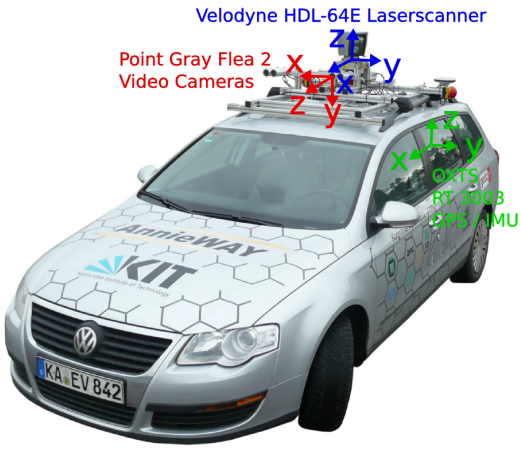


Figure 5.4: The car used to record the KITTI dataset.

minor modifications, while the other parts of ICP algorithm remain the same.

With this distance metric, we can efficiently transfer the knowledge gained from image feature correspondences into the process of laser point association without requiring an overlap in the field-of-views of both sensors.

5.5 Experimental Evaluation

In this chapter, we present a novel approach to joint laser-camera ego-motion estimation. We make the claims that our approach (i) allows for accurate frame-to-frame alignment from monocular vision and laser range data and that (ii) it is able to exploit the advantages of both modalities. Our experiments are designed to support these two claims.

We perform our evaluations on the KITTI dataset [Geiger et al., 2012a], which is a well known dataset recorded from a sensor vehicle driving in the city of Karlsruhe in Germany. As depicted in Figure 5.4, the vehicle is equipped with two sets of stereo cameras, a Velodyne HDL-64E and a GPS/INS system. We use the KITTI dataset because it is a standard dataset for these type of problems and we have ground truth available.

5.5.1 Error Evaluation

The first set of experiments is designed to support both claims, i.e., that our approach can accurate align frames pairwise and that it is able to exploit the advantages of both modalities.

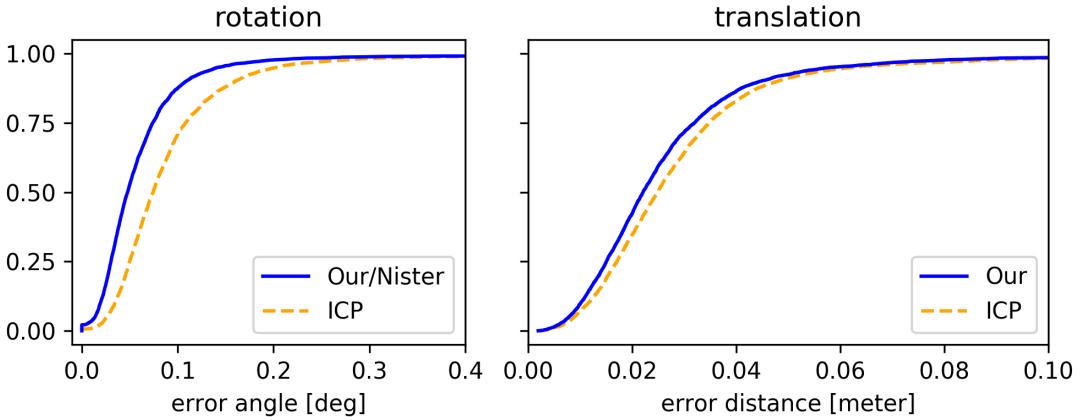


Figure 5.5: Cumulative error distribution: Percentage (y-axis) of cumulative errors (x-axis) in rotation and translation for our approach and laser-only ICP.

Figure 5.5 shows the cumulative error plots for the rotation (left plot) and the translation (right plot) of our approach in comparison to an optimized, laser-only ICP. These are the cumulative plots over all sequences of the KITTI dataset but the plots for the individual datasets looks similar and show the same characteristics. Within this evaluation, we use the geodesic distance on a unit sphere to parameterize the rotational error. Given a rotation matrix ΔR, we compute

$$e_{\text{rot}} = \arccos\left(\frac{\text{trace}(\Delta R) - 1}{2}\right) \tag{5.18}$$

as the error angle.

Based on the left plot of Figure 5.5, it is clearly visible that the relative orientation information from the camera provides a better estimate of rotational component of the ego-motion (blue line) than laser-based ICP (dashed orange line). The blue line shows the performance of Nistér's 5-point algorithm and our approach (as we use the 5-point algorithm for the rotation estimation). The fact that this approach is better than laser-based ICP can be seen because the blue curve is always above the dashed, orange one.

We can furthermore show that our approach outperforms laser-only ICP when estimating the translational part, see right plot of Figure 5.5. The blue line represents our approach and is always above the dashed, orange one, which corresponds to laser-only ICP. This is the case for two reasons: First, our point-to-point data association described in Section 5.4 is better than the regular ICP data association as we drastically reduce the number of potential matches since we only need to consider points that are in line with the rotation. This avoids several wrong data associations. Second, the orientation estimates of our approach

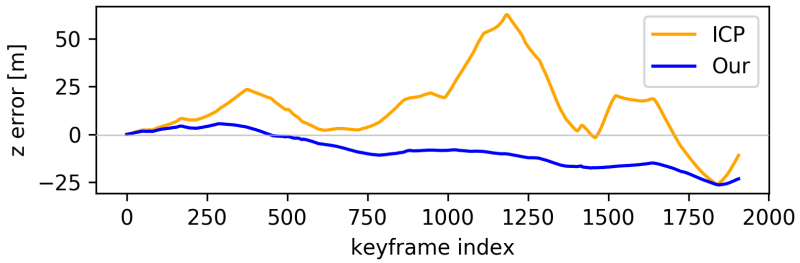


Figure 5.6: Error in Z direction of sequence 00 for the keyframes.

are better than those of laser-only ICP and they also impact the translation estimation. Note that no translational error can be provided for the camera-only case as the scale, i.e., the length of the motion vector, cannot determined using a monocular camera.

Thus, we can conclude that our approach outperforms visual odometry from the monocular camera (because we obtain an accurate scale estimate) as well as laser-based ICP (more accurate orientation and translation).

5.5.2 Trajectory Estimation

This second part of the evaluation also supports the first claim and furthermore provides a better visual impression about the quality of the estimated trajectories. We plot the ground truth trajectories, our estimates, and the ones of laser-only ICP for several KITTI sequences in Figure 5.7. Note that, compared to several other methods, our approach achieves its performance without any loop-closing.

In all sequences except the top left one, it is rather clear from the shown X/Y plots that our trajectory estimate is always closer to the ground truth than the ones obtained by laser-only ICP. For the top left trajectory (sequence 00), this is more difficult to see. When inspecting the error in the Z component, however, we can see in Figure 5.6 that our approach clearly outperforms laser-only ICP. For nearly all keyframes, the error in the height estimate (Z axis) is larger for the laser-only ICP estimate.

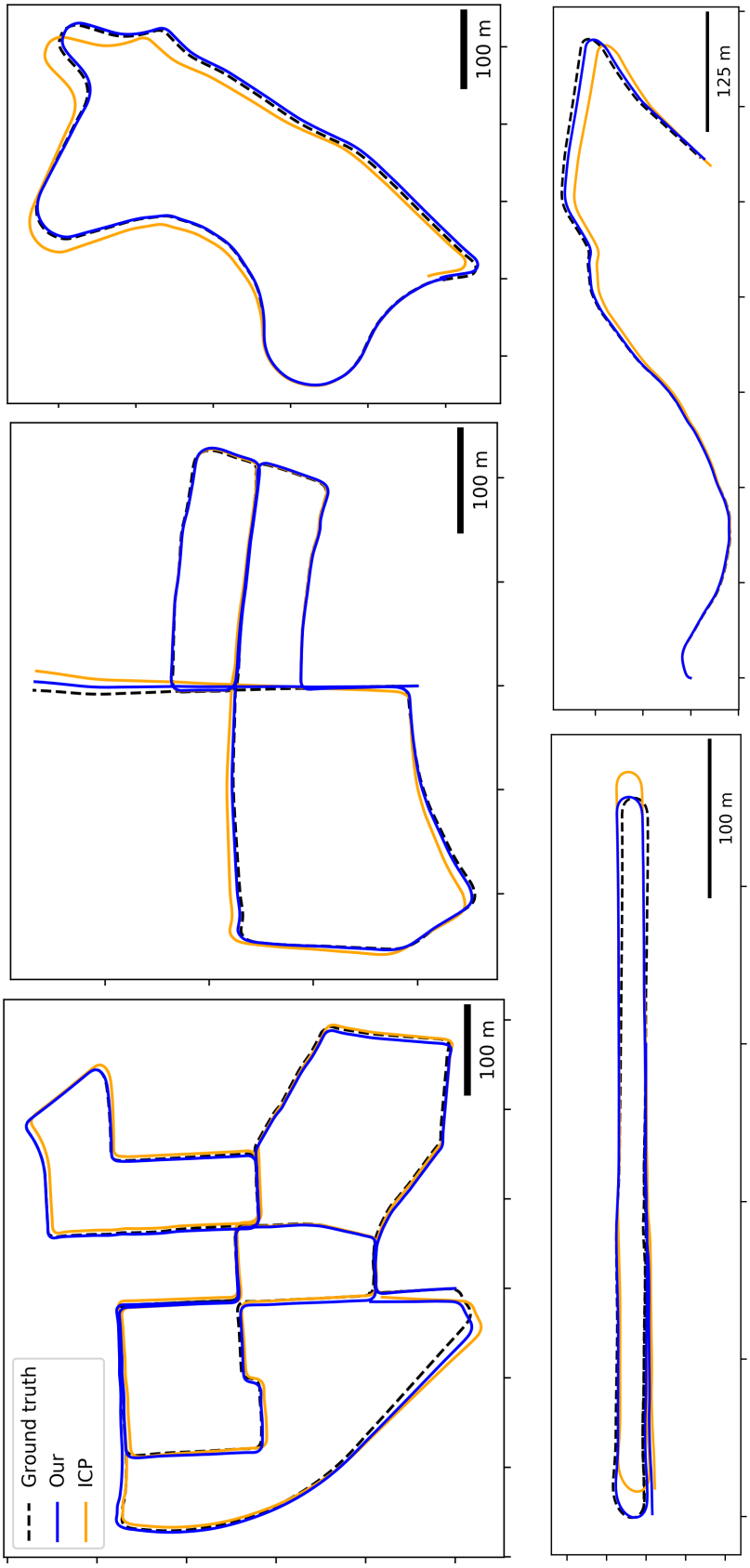


Figure 5.7: Resulting trajectories from a subset of the KITTI sequence through our frame-to-frame registration without any loop-closing.

5.6 Summary

In this chapter, we presented a novel approach to ego-motion estimation using a monocular camera and a laser range finder jointly. Our approach estimates the 5-DoF relative orientation from the camera images and uses a novel variant of ICP with 1-DoF to estimate the scale. We can furthermore constrain the possible data associations among the point clouds given constraints derived from the relative orientation. We implemented our approach and evaluated it using KITTI data. In sum, our approach provides accurate trajectory estimates, which are better than those of each sensing modality alone.

Chapter 6

Joint Ego-motion Estimation Through Direct Photometric Alignment

In this chapter, we propose the second camera-laser fusion algorithm addressing the ego-motion estimation problem. Unlike the previous method in Chapter 5, which indirectly fuses the information through image feature point correspondences, the method proposed here is a direct one based on pixel-wise photometric alignment, and it is designed to achieve superior estimation accuracy through maximizing the information usage of both the image and the laser scan, assuming the sensor field-of-view overlap is substantial.

In Section 6.1, we explicitly address the occlusion problem with a prediction algorithm tailored to deal with sparse laser point clouds. In Section 6.2, to address issues due to the sparsity of the range measurements, our approach identifies planar point sets from the laser data and extract the corresponding pixel patches from image data. The extracted dense planar image patches together with the sparse non-planar point cloud and pixels information are jointly used to estimate the frame-to-frame motion. For that, we rely on a homography formulation (described in Section 6.3) that is capable of incorporating both types of pixel alignments. In Section 6.4, to achieve high estimation accuracy, our approach employs a two-stage registration strategy. The first stage is aimed to ensure a proper initial pose estimate by jointly performing a coarse photometric pixel-alignments together with a geometric point cloud registration. The resulting estimate is then refined in the second stage by aligning only pixel intensities at the finest image level. The motivation behind this strategy is to combine the photometric and the geometric information while avoiding their respective pitfalls,

i.e., local minima in photometric alignment and estimation bias in point cloud registration due to sparse point correspondences. In Section 6.5, we evaluated the whole algorithm pipeline on different datasets and provided comparisons to other existing techniques. The evaluation result supported the claim that our approach can achieve competitive estimation accuracy.

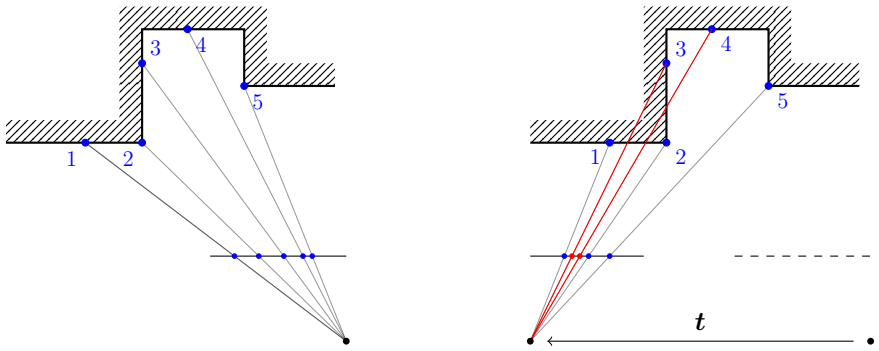
Similar to the discussion of last chapter, we assume the camera and laser scanner are time synchronized (e.g., by using hardware trigger) and that their relative transformation on the robot is known (i.e. calibrated). Thus, one can project a 3D laser point to the camera image and directly obtain the intensity value of the corresponding image pixel. We denote the *previous* visual-laser measurement, which consists of a point cloud $\{\boldsymbol{a}_i \in \mathbb{R}^3\}_{i=1}^{N}$ and an image $\mathcal{I}_a$, using the character a, while the *current* one uses b with point cloud $\{\boldsymbol{b}_j \in \mathbb{R}^3\}_{j=1}^{M}$ and image $\mathcal{I}_b$. Our task is to estimate the ego-motion of the robot between a and b, which consists of a relative rotation $R \in \mathrm{SO3}$ and translation $\boldsymbol{t} \in \mathbb{R}^3$.

6.1 Occlusion Detection for Sparse Point Clouds

Photometric alignment is based on the *constant image brightness* assumption, which assumes the intensities of corresponding pixels of a scene point in two (or more) images are equal. However, this assumption will be violated if the scene point is occluded during the viewpoint changes. The occluded points are outliers to the system and will deteriorate the estimation accuracy if they are not removed from the photometric alignment process.

To overcome the occlusion problem, we propose a novel method to predict which laser points of a sparse point cloud will be occluded under a certain camera motion. We then explicitly exclude these points from the motion estimation step. Compared to the standard Z-buffering approach, which is often used for dense depth images, our approach is more suitable for dealing with sparse laser point cloud data.

The key observation of our approach is that whenever parts of a point cloud are occluded in the current camera view, *the relative pixel order of the projected point cloud* in the current image will be different from the previous one. Consider Figure 6.1 as an example. Assume there are five scene points, which are labeled from left to right as $1, 2, 3, 4, 5$ in the original camera view (Figure 6.1a). After a camera translation, $\boldsymbol{t}$, we observe the scene again and obtain a new camera image by re-projecting the five points, as illustrated in Figure 6.1b. However, points 3 and 4 are occluded in the new view. Note that, at the same time, the pixel order in the new image becomes $1, 3, 4, 2, 5$ from left to right, which is different from



(a) before: 1 2 3 4 5 (b) after: 1 **3** **4** 2 5

Figure 6.1: The pixel order of projected 3D points will change if occlusions happened. (a) In the original view, we label the 3D points from left to right as $1, 2, 3, 4, 5$. (b) After the camera movement $\boldsymbol{t}$, points 3 and 4 are occluded and lead to a different pixel order in the new camera image, which is now $1, 3, 4, 2, 5$ from left to right. We exploit such pixel order changes to perform occlusion detection for sparse 3D point clouds.

Figure 6.2: Example result of the occlusion detection algorithm. The red points are the image projections of the (predicted) occluded points of a laser scan. The occlusions happen mostly at the borders of objects and switch sides as the viewpoint changes.

the original order $1, 2, 3, 4, 5$.

The phenomenon of pixel order changes is not limited to perspective projection but also holds true for the spherical projection, and we can exploit such pixel-order changes to identify occluded scene points. To be more specific, we can first compare the two sequences and find out which point-sets have been swapped, e.g. points {3,4} and point 2 in Figure 6.1. One of the point-set is occluded while the other is not. We identify the occluded one by comparing their depth values and the larger one is occluded, i.e. points 3 and 4 have larger depths than point 2, therefore they are occluded.

We generalize this idea and propose Algorithm 4 to perform occlusion detection for sparse 3D point clouds. Our algorithm takes a row (or a column) of points as input. A row (or column) means a subset of points having a close or the same pitch (or yaw) laser beam angle. Another input variable is a translation vector $\boldsymbol{t}$ representing the new camera position, which is expressed under the original point cloud coordinate frame. Because rotational camera movements do not induce scene occlusion, they are therefore not needed in the calculation.

Figure 6.2 shows an example detection result on the KITTI dataset. The red points are the predicted occluded scene points, projected on images taken at a different location. Notice how the projected pixels lie on different objects because of the occlusion. In this example, the occluded points take up to 11% of the total visible points. It is worth noticing that the occlusion happens not only because of the camera ego-motion, but also due to the displacement between the camera and the LiDAR sensors on the vehicle. Therefore, to account for both effects, we use the camera ego-motion plus the camera-LiDAR displacement as the translation input $\boldsymbol{t}$ to Algorithm 4.

Algorithm 4 Occlusion Prediction

1: **Input:**
- A row of points $\mathcal{P}$
- Translational movement $\boldsymbol{t}$

2: **Output:**
- Occlusion mask $\mathcal{O}$

▷ **Step 1. Point projection**

3: Index list $\mathcal{A} \leftarrow$ **Sort** indices of $\mathcal{P}$ **by** $\{\pi_x(\boldsymbol{p}) \mid \boldsymbol{p} \in \mathcal{P}\}$

4: Index list $\mathcal{B} \leftarrow$ **Sort** indices of $\mathcal{P}$ **by** $\{\pi_x(\boldsymbol{p} - \boldsymbol{t}) \mid \boldsymbol{p} \in \mathcal{P}\}$

▷ **Step 2. List comparison**

5: Occlusion mask $\mathcal{O}(\cdot) \leftarrow$ False ▷ default: nothing occluded

6: Index $a \leftarrow \mathcal{A}$.pop()

7: Index $b \leftarrow \mathcal{B}$.pop()

8: **loop** size($\mathcal{P}$) times

9: **if** $a == b$ **then**

10: update both a and b ▷ in line 18

11: **else**

12: **if** $\mathcal{P}_z(a) > \mathcal{P}_z(b)$ **then** ▷ a is behind b thus occluded

13: mark $\mathcal{O}(a) \leftarrow$ True

14: update only a

15: **else** ▷ b is occluded

16: mark $\mathcal{O}(b) \leftarrow$ True

17: update only b

18: **if** update a **then**

19: $a \leftarrow \mathcal{A}$.pop()

20: **if** $\mathcal{O}(a)$ is True **then repeat** line 19

21: **if** update b **then**

22: b $\leftarrow \mathcal{B}$.pop()

23: **if** $\mathcal{O}(b)$ is True **then repeat** line 22

24: **return** $\mathcal{O}$

6.2 Coplanar Point Detection

The depth measurements are often sparse and cover only a small portion of the image pixels when projected into the camera image (see Figure 6.4c for example). While most image pixels do not have depth information from the laser, such depth-less image pixels are either discarded (e.g., in Della Corte et al., 2018) or falsely assigned with a constant depth the same as their associated pixel (e.g., in Shin et al., 2018), which are both sub-optimal solutions.

We overcome this problem by exploiting planar regions in the scene, which are often abundant in structured (urban) environments. A scene plane usually corresponds to a large number of pixels, and such pixels can also be used to estimate the motion parameters even without knowing their depth values, because they can be projected to another image using plane-induced homography given the plane parameters. Therefore, to include as much as possible pixel information in the photometric term, our approach explicitly detects scene planes from the point cloud and use them for estimating the camera motion.

To identify which subset of the laser points are parts of a planar region, we propose a grid-based method inspired by the work by Weingarten et al. [2003] and Xiao et al. [2011]. The main idea is to first discretize the point cloud into a grid of cells and then, for each cell use principal component analysis (PCA) to fit a plane to the points that are inside the cell.

We also accelerate the detection process by incorporating prior knowledge about existing planes, e.g., knowledge about the ground plane or previously detected planes. Given prior plane parameters $(\boldsymbol{n}, d)$, where $\boldsymbol{n}$ is the normal vector of the plane and d is the plane-to-camera-origin distance, we compute the point-to-plane distance $|\boldsymbol{n}^\mathsf{T}\boldsymbol{p} - d|$ for each point $\boldsymbol{p}$ in the new point cloud. Points with a small distant are identified as inlier points for that plane. These inlier points are removed from the point cloud and the fitting process is performed again with the next prior plane parameters until all hypotheses are tested. This process happens as the first stage of the planar point detection and can identify a large portion of the planar points. After that, all the remaining (unmatched) points are then handled by the grid-based detection process. Algorithm 5 summarizes our proposed coplanar point detection method.

Algorithm 5 Coplanar Point Detection

Input:
- Point cloud $\mathcal{IP}$
- Prior plane parameters $\{(\boldsymbol{n}, d)\}$

Parameter:
- Grid size s
- Point-to-plane distance threshold ϵ

Output:
- Planar points list $\mathcal{P}$
- Plane normal list $\mathcal{N}$

▷ **Step 1: Prior Plane Fitting**

for each prior plane parameters $(\boldsymbol{n}, d)$ **do**
  Inliers $\mathcal{I} \leftarrow \{\boldsymbol{p} \in \mathcal{IP} \setminus \mathcal{P} \mid |\boldsymbol{n}^\mathsf{T}\boldsymbol{p} - d| < \epsilon\}$
  Planar points list $\mathcal{P} \xleftarrow{\text{insert}}$ Inliers $\mathcal{I}$
  Plane normal list $\mathcal{N} \xleftarrow{\text{insert}} \boldsymbol{n}$

▷ **Step 2: Discretization**

Point list $\mathcal{L} \leftarrow \{\emptyset\}$
for each point $\boldsymbol{p}$ in the remaining point cloud $\mathcal{IP} \setminus \mathcal{P}$ **do**
  Cell coordinates $(u, v, w) \leftarrow$ **discretize**$(\boldsymbol{p}, s)$
  Point list $\mathcal{L} \xleftarrow{\text{insert}}$ item $\{(u, v, w) : \boldsymbol{p}\}$
Sort point list $\mathcal{L}$ **by** (u, v, w)
Cell list $\mathcal{C} \leftarrow \{\emptyset\}$
Current cell $c \leftarrow \{\emptyset\}$
for each point $\boldsymbol{p}_i$ in the sorted point list $\mathcal{L}$ **do**
  Current cell $c \xleftarrow{\text{insert}} \boldsymbol{p}_i$
  if current $(u, v, w)_i \neq$ next coordinates $(u, v, w)_{i+1}$ **then**
    Cell list $\mathcal{C} \xleftarrow{\text{insert}}$ current cell c
    Current cell $c \leftarrow$ new cell $\{\emptyset\}$

▷ **Step 3: Plane Detection**

for point set $\{\boldsymbol{p}\}$ of each cell in the cell list $\mathcal{C}$ **do**
  Eigenvalues $\lambda_1 \leq \lambda_2 \leq \lambda_3 \leftarrow$ **PCA**$(\{\boldsymbol{p}\})$
  if size$(\{\boldsymbol{p}\}) < 7$ **then** ▷ $\{\boldsymbol{p}\}$ is too sparse.
    skip this cell
  if λ_1/**size**$(\{\boldsymbol{p}\}) > \epsilon$ **then** ▷ $\{\boldsymbol{p}\}$ not planar.
    skip this cell
  Normal vector $\boldsymbol{n} \leftarrow$ eigenvector $\boldsymbol{v}_1$ (for eigenvalue λ_1)
  Plane normal list $\mathcal{N} \xleftarrow{\text{insert}} \boldsymbol{n}$
  Planar points list $\mathcal{P} \xleftarrow{\text{insert}} \{\boldsymbol{p}\}$
return $\mathcal{P}, \mathcal{N}$

6.3 Homography-Based Photometric Alignment

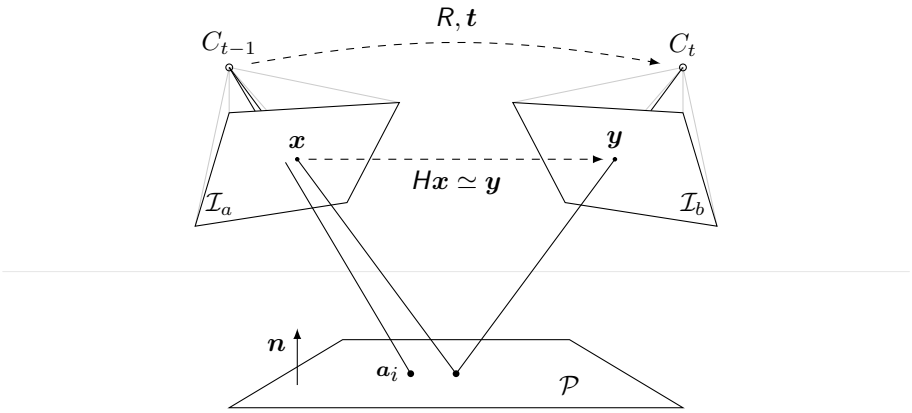


Figure 6.3: Plane-induced homography relation of two images.

Once we extracted the (dense) planar image patches, they are used together with the (sparse) non-planar point cloud-projected pixels to estimate the motion parameters with our homography formulation.

Figure 6.3 shows a plane-induced homography relation of two images. Assume a laser point $\boldsymbol{a}_i$ is located on a 3D plane $\mathcal{P}$ with a normal vector $\boldsymbol{n}$ and a plane-to-camera-origin distance $d \stackrel{\text{def}}{=} \boldsymbol{n}^\mathsf{T}\boldsymbol{a}_i$. Any points $\boldsymbol{p}$ belongs to this plane will satisfy the equation

$$\boldsymbol{n}^\mathsf{T}\boldsymbol{p} = d. \tag{6.1}$$

Now assume there is an previous image point $\boldsymbol{x} \stackrel{\text{def}}{=} [u, v, 1]^\mathsf{T}$ on $\mathcal{I}_a$, its back projected ray $\boldsymbol{r}(\lambda) \stackrel{\text{def}}{=} \lambda\boldsymbol{x}$ intersects the plane $\mathcal{P}$. According to Equation (6.1), the intersection happens at

$$\boldsymbol{n}^\mathsf{T}(\lambda\boldsymbol{x}) = d \quad \mapsto \quad \lambda = \frac{d}{\boldsymbol{n}^\mathsf{T}\boldsymbol{x}}. \tag{6.2}$$

Therefore, the intersection point is $\frac{d}{\boldsymbol{n}^\mathsf{T}\boldsymbol{x}}\boldsymbol{x}$ and will have a 3D homogeneous coordinates

$$\begin{bmatrix} \frac{d}{\boldsymbol{n}^\mathsf{T}\boldsymbol{x}}\boldsymbol{x} \\ 1 \end{bmatrix} \simeq \begin{bmatrix} d\boldsymbol{x} \\ \boldsymbol{n}^\mathsf{T}\boldsymbol{x} \end{bmatrix} \in \mathbb{R}^4. \tag{6.3}$$

Given the relative motion parameters R and $\boldsymbol{t}$, we can project this intersection point to the current image $\mathcal{I}_b$ and obtain

$$\boldsymbol{y} \simeq \begin{bmatrix} R & \boldsymbol{t} \end{bmatrix} \begin{bmatrix} d\boldsymbol{x} \\ \boldsymbol{n}^\mathsf{T}\boldsymbol{x} \end{bmatrix} = (dR + \boldsymbol{t}\boldsymbol{n}^\mathsf{T})\boldsymbol{x}. \tag{6.4}$$

With the camera intrinsic matrix K and $\boldsymbol{x}$, $\boldsymbol{y}$ being in pixel coordinates, Equation (6.4) becomes

$$\boldsymbol{y} \simeq K(dR + \boldsymbol{t}\boldsymbol{n}^\mathsf{T})K^{-1}\boldsymbol{x}. \tag{6.5}$$

Therefore, the pixels of a 3D plane in the two images, i.e. $\boldsymbol{x}$ and $\boldsymbol{y}$, are related through

$$\boldsymbol{y} \simeq H\boldsymbol{x}, \tag{6.6}$$

where $H \overset{\text{def}}{=} K(dR + \boldsymbol{t}\boldsymbol{n}^\mathsf{T})K^{-1}$ is a plane-induced homography.

For an arbitrary laser point that does not lie on a planar region, the homography formulation in Equation (6.6) is still applicable because it can be seen as a special case where the pixel by itself defines a fronto-parallel patch, i.e.,

$$\mathcal{P} = \{\boldsymbol{a}_i\} \qquad \text{and} \qquad \boldsymbol{n} = [0, 0, 1]^\mathsf{T}. \tag{6.7}$$

In this case, $d = [0, 0, 1]\boldsymbol{a}_i$ is the depth of $\boldsymbol{a}_i$, thus $\boldsymbol{a}_i = dK^{-1}\boldsymbol{x}$ and the entity $H\boldsymbol{x}$ amounts to the standard 3D point projection as

$$H\boldsymbol{x} = K(R + \frac{\boldsymbol{t}\boldsymbol{n}^\mathsf{T}}{d})dK^{-1}\boldsymbol{x} \tag{6.8}$$

$$= K(R\boldsymbol{a}_i + \boldsymbol{t}\frac{\boldsymbol{n}^\mathsf{T}\boldsymbol{a}_i}{d}) \tag{6.9}$$

$$= K(R\boldsymbol{a}_i + \boldsymbol{t}). \tag{6.10}$$

Base on this homography formulation, we define our photometric cost function for estimating the motion parameters R and $\boldsymbol{t}$ as

$$E_{\text{pho},i} \overset{\text{def}}{=} \sum_{\mathbf{x} \in \mathcal{P}_{\text{img}}(\boldsymbol{a}_i)} \varphi\Big(\underbrace{\mathcal{I}'_a(\mathbf{x}) - \mathcal{I}_b\big(\pi(H_i(R, \boldsymbol{t})\boldsymbol{x})\big)}_{\{e_{\text{pho}}\}} \Big) \tag{6.11}$$

where

- $\varphi(\cdot)$ is a robustification function based on the t-distribution (of five degree of freedom, as in Kerl et al. [2013]):

$$\varphi(e) \overset{\text{def}}{=} \frac{6}{5 + \frac{e^2}{\sigma^2}}\, e^2 \tag{6.12}$$

  with σ being the standard deviation of all residuals $\{e\}$;

- $\mathbf{x} \overset{\text{def}}{=} [u, v]^\mathsf{T}$ is a pixel coordinates, and $\boldsymbol{x} \overset{\text{def}}{=} [u, v, 1]^\mathsf{T}$ is its homogeneous form;

- $\pi(\cdot)$ is the Euclidean normalization that transforms homogeneous coordinates into (inhomogeneous) pixel coordinates, i.e. $\pi(\boldsymbol{x}) = \mathbf{x}$, or more generally

$$\pi([u, v, w]^\mathsf{T}) \stackrel{\text{def}}{=} [u/w, v/w]^\mathsf{T}; \tag{6.13}$$

- $\mathcal{P}_{\text{img}}(\boldsymbol{a}_i)$ denotes a set of neighboring pixels around the image point of $\boldsymbol{a}_i$. If the image point of $\boldsymbol{a}_i$ is denoted as $\mathbf{a}_i \stackrel{\text{def}}{=} \pi(K\boldsymbol{a}_i)$, then

$$\mathcal{P}_{\text{img}}(\boldsymbol{a}_i) \stackrel{\text{def}}{=} \begin{cases} \{\mathbf{x} \in \mathbb{Z}^2 \,|\, \|\mathbf{a}_i - \mathbf{x}\| \leq r\}, & \text{if } \boldsymbol{a}_i \text{ is planar,} \\ \{\mathbf{a}_i\}, & \text{if non-planar,} \end{cases} \tag{6.14}$$

  where r is a predefined radius;

- $H_i(R, \boldsymbol{t}) \stackrel{\text{def}}{=} K(\boldsymbol{a}_i^\mathsf{T}\boldsymbol{n}_i R + \boldsymbol{t}\boldsymbol{n}_i^\mathsf{T})K^{-1}$ is the homography associated to the laser point $\boldsymbol{a}_i$. For non-planar points we set $\boldsymbol{n}_i = [0, 0, 1]^\mathsf{T}$. Otherwise $\boldsymbol{n}_i$ is calculated from the laser point cloud using a method described in Section 6.2;

- $\mathcal{I}'_a(\cdot) \stackrel{\text{def}}{=} \alpha\mathcal{I}_a(\cdot) + \beta$ is used to model the gain, α, and the bias, β, between the two intensity images, to account for possible different camera exposure settings and ambient light changes. Both α and β are unknown parameters to be estimated during the optimization.

6.4 Two-Stage Registration

Photometric alignment is in essence a highly nonlinear optimization problem with lots of local minima. To ensure a proper initial estimate and avoid false minima, we first optimize a joint objective that rewards both consistent photometric alignment (with smoothed images) as well as tight point cloud registration. For that, besides the photometric term in Equation (6.11), we also incorporate a geometric term to account for the point-to-plane point cloud registration errors as in the ICP:

$$E_{\text{geo},i} \stackrel{\text{def}}{=} \varphi\big(\underbrace{\boldsymbol{n}_i^\mathsf{T}(R\boldsymbol{a}_i + \boldsymbol{t} - \boldsymbol{b}'_i)}_{\{e_{\text{geo}}\}}\big), \tag{6.15}$$

where $\boldsymbol{b}'_i$ is the nearest-neighbor to the transformed $\boldsymbol{a}_i$ in the point cloud $\boldsymbol{b}$, determined by using a k-d tree search. For non-planar points, $\boldsymbol{n}_i$ refers to the surface normal of the points.

Combining Equation (6.11) and Equation (6.15), we have in the first registration stage a minimization problem of the form:

$$\underset{\boldsymbol{R},\boldsymbol{t},\alpha,\beta}{\text{argmin}} \frac{1}{\sigma_{\text{geo}}^2} \sum_i E_{\text{geo},i} + \frac{1}{\sigma_{\text{pho}}^2} \sum_{i \in \text{Vis}} E_{\text{pho},i}, \tag{6.16}$$

where

- σ_{geo} and σ_{pho} are the standard deviation of the residuals $\{e_{\text{geo}}\}$ and $\{e_{\text{pho}}\}$;
- $i \in \text{Vis}$ stands for laser points that are visible and not occluded in both camera images $\mathcal{I}_a$ and $\mathcal{I}_b$, which are smoothed by a Gaussian function and then down-sampled.

In the second stage of the alignment procedure, the estimation of $\boldsymbol{R}$ and $\boldsymbol{t}$ is refined by performing photometric alignment at the finest resolution. Therefore, a cost function similar to Equation (6.16) is used in the second stage, but with only the photometric term E_{pho} and using raw images.

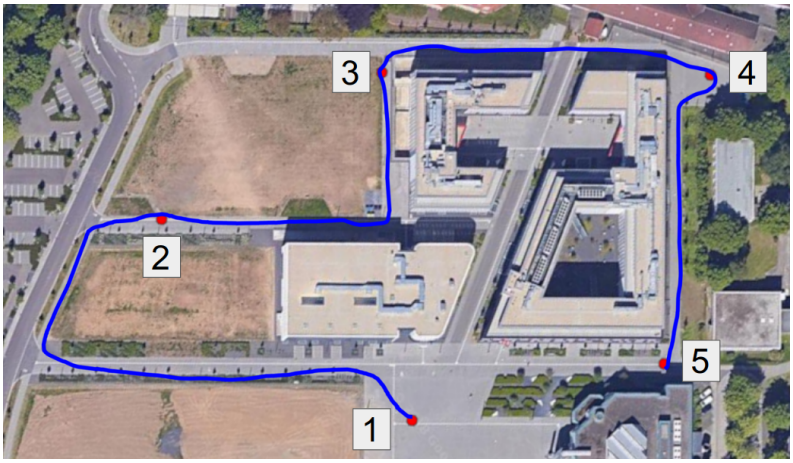
In both stages, we minimize the objective with a standard iterative Gauss-Newton optimization algorithm. Our experimental result in Section 6.5 suggests that our two-stage registration strategy can significantly improve the estimation accuracy.

6.5 Experimental Evaluation

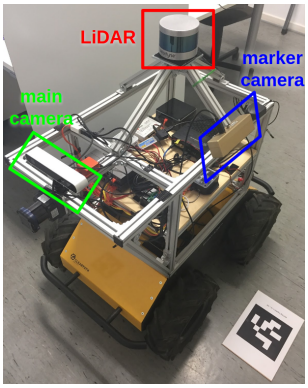
The main focus of this work is a novel direct approach to joint laser-camera odometry. The experiments are designed to show the capabilities of our method and to support our key claim that our approach is able to accurately estimate frame-to-frame motion using monocular vision and laser range data. We perform the evaluations on own robotic datasets as well as on publicly available ones.

6.5.1 Outdoor LiDAR-Camera Dataset with Ground Truth Control Points

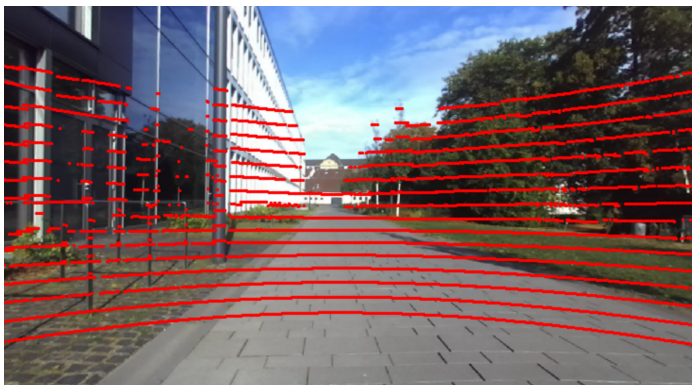
The first experiment is to verify the proposed method with a mobile robot in an outdoor environment. The robot is a Clearpath Husky mobile platform equipped with a 16-beams Velodyne VLP-16 LiDAR and a stereo-camera (we use images from only the left camera here), as shown in Figure 6.4b. Along the performed experiment route, there are five geodetic control points on the ground with precisely measured coordinates around our campus, as illustrated in Figure 6.4a. We place Apriltag markers [Olson, 2011] on top of the control points and utilize an auxiliary camera on the robot to detect these markers on the ground when the robot drives by them. In this way, we obtain the positions of the robot relative to the control points. We use these positions as ground-truth locations in the environment to evaluate the trajectory estimated with our approach. Due to the orientation of the markers are somewhat uncertain, we compare the point-to-point distances and the result is shown in Table 6.1.



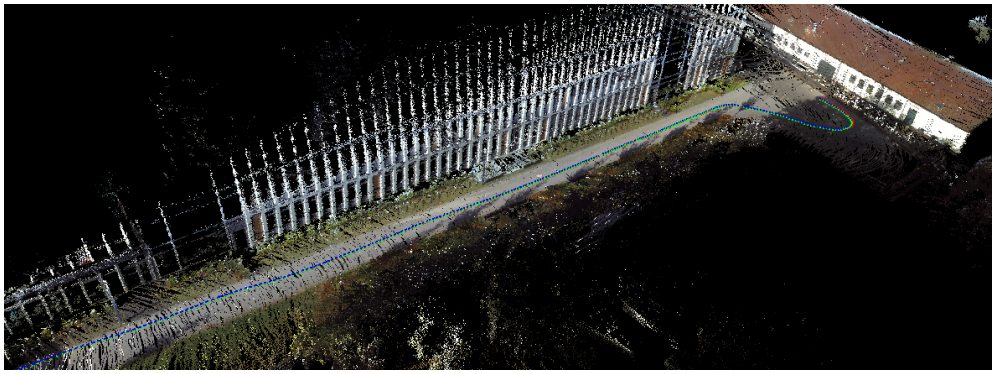
(a) Experimental route and control points.



(b) Robot platform.



(c) Laser points projected into the camera image.



(d) 3D mapping result of the path from control point 4 to 5.

Figure 6.4: Outdoor experiment. We drive a Clearpath Husky robot around the campus. The experimental path passes by five precisely known geodetic control points, which are used for the ground truth evaluation.

Table 6.1: Relative distance error measured at five control points.

Segments		1-2	2-3	3-4	4-5
Ground Truth Point Dist. (m)		114.12	93.86	116.87	103.62
Estimated Trajectory length (m)		213.47	133.16	125.17	110.05
Geometric term only	Dist. Error (m)	3.20	1.43	2.60	2.92
	Rel. Error (%)	1.49	1.07	2.07	2.65
Photometric term only	Dist. Error (m)	0.92	2.25	0.26	0.57
	Rel. Error (%)	0.43	1.69	0.21	0.52
Combined	Dist. Error (m)	0.26	0.12	0.02	0.35
	Rel. Error (%)	**0.12**	**0.09**	**0.02**	**0.32**

Table 6.2: Comparison on relative translational error using the KITTI odometry dataset.

	Sequences without loops			
Approach	01	03	04	10
J. Zhang and Singh [2017]	1.4%	0.9%	0.7%	0.8%
Shin et al. [2018]	1.5%	**0.9%**	0.7%	0.7%
Our	**1.0%**	1.1%	**0.6%**	**0.7%**

As shown in the last row of Table 6.1, our approach achieves a competitive accuracy with relative distance errors as low as 0.1%, without using loop-closing. Figure 6.4d depicts a colored point cloud generated by our approach.

To see the benefit of using the two-stage registration strategy, we also include in Table 6.1 the results of using only the geometric term or the photometric term. The result suggests that the accuracy improvements of using two-stage registration are significant.

6.5.2 Comparison to State-of-the-Art Methods Using KITTI

The second experiment performs evaluations about the motion estimation quality of our approch using the odometry datasets of the KITTI odometry dataset [Geiger et al., 2012a]. We performed the motion estimation using the point clouds from the 64-beams Velodyne LiDAR and the monochromic images from the camera 0. The results of sequences without loop-closing are reported in Table 6.2 for comparison, including the reported results of Shin et al. [2018] (a photometric-alignment based visual-laser odometry approach), as well as the state-of-the-art laser-based approach, LOAM [J. Zhang and Singh, 2017]. The result shown in

Table 6.2 suggests that our approach perform better or on par with the state-of-the-art in terms of translational error.

6.6 Summary

In this chapter, we presented a novel direct approach to joint laser-camera odometry. Our method exploits planar information, performs occlusion prediction, and employs a two-stage registration. This allows us to estimate frame-to-frame motions with high accuracy. We implemented and evaluated our approach on different datasets and provided comparisons to other existing techniques. The evaluation result supported the claim that our approach can achieve competitive estimation accuracy.

Chapter 7

Related Work

In this chapter, we review previous work on sensor extrinsic calibration problem, and report existing laser-camera fusion approaches. We discuss the strengths and weakness of previous research, and explain their relations to our work presented in this thesis.

7.1 Extrinsic Calibration

In this thesis, we categorized common calibration methods into three types: AX=B, AX=YB, and AX=XB. We now briefly survey works that are related to each method.

7.1.1 Marker-Based Methods

Calibration methods based on model AX=B formed the vast majority of calibration studies, covering popular exteroceptive sensors such as cameras and laser scanners. We refer to approaches based on model AX=B as marker-based approaches, because the setup of a reference frame common to all sensors often requires using markers, in the form of control points, landmarks, or reference objects with known geometry.

Marker-based approaches try to estimate the extrinsic parameters directly from the sensed features by maximizing a quality measure or the agreement of the sensor data with specific constraints, e.g., [Faugueras and Toscani, 1989; Pandey et al., 2015; Scaramuzza et al., 2007; Taylor et al., 2015]. Typical sensors that fit in this paradigm are cameras and laser scanners.

For calibration involving cameras, usually a set of point correspondences matched from different views of the same scene are used. The sum of squared point re-projection errors is then served as a cost function for an error minimization in the parameter space, e.g., in the work of [Carrera et al., 2011; Faugueras and Toscani, 1989; Heng et al., 2014; Heng et al., 2013; Zisserman et al., 1995]. There are also methods operate directly on dense images, using a metric known as Mutual Information [Shannon, 1948], but they are mainly used for aligning hyperspectral cameras, or medical imaging devices such as Medical Resonance Imaging (MRI) and Computed Tomography (CT). A survey of mutual-information-based techniques has been presented by Pluim et al. [2003].

For calibration involving laser scanners, objects with distinguishable shape are often used as markers, such as flat surface [Fernndez-Moral et al., 2015; Rwekmper et al., 2015], checkerboard [Geiger et al., 2012b], scene corners [Gomez-Ojeda et al., 2015], or even trajectories of tracked objects [Schenk et al., 2012].

For camera-laser calibration problems, gradient information can also be used, e.g., in the work [Taylor et al., 2013; Taylor et al., 2015] by a metric called gradient orientation measure. The work of Corsini et al. [2009] provides another solution if we consider camera-laser calibration as an image-to-geometry registration problem. In their approach, illumination-related geometric properties such as surface normals, ambient occlusion and reflection directions, are used to generate a 2D image from a 3D model, so that mutual-information-based techniques can be used to register the synthetic image with the photometric one from cameras.

7.1.2 Relative-Motion-Based Methods

Calibration methods based on the model AX=XB forms another popular group, which we refer as relative-motion-based methods. Unlike the previous marker-based approaches, this type of methods exploit constraints between the motions of individual sensors instead of external markers, hence the name. They are sensor agnostic and can be used to calibrate almost any kind of sensor that can produce a (relative) trajectory estimate of itself.

The iconic equation AX=XB was first proposed in the work of Shiu and Ahmad [1989]. They try to calibrate a camera that is mounted next to an end-effector of a robotic arm, as illustrated in Figure 7.1. Such a calibration problem is often referred as hand-eye calibration, which is a typical case of motion-based extrinsic calibration.

Many previous works are focused on providing a solution to the equation AX=XB (hence the hand-eye calibration problem). Shiu and Ahmad [1989] provide a closed-form solution by decoupling the rotation and translation estimation, and

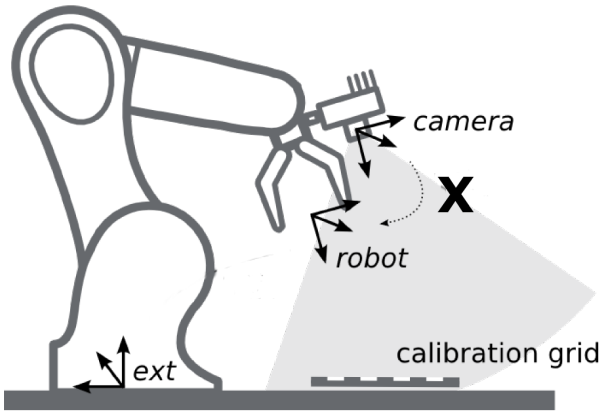


Figure 7.1: AX=XB hand-eye calibration problem is aimed to calibrate a camera that is mounted next to an end-effector of a robotic arm.

discussed the solution uniqueness condition. Following works propose various alternative closed form solutions by using, for example, angle-axis representation [Park and Martin, 1994], dual-quaternions formulation [Daniilidis, 1999], screw motion and screw axis [Fassi and Legnani, 2005], or the idea of orthogonal dual tensors [Condurache and Burlacu, 2016]. Despite the simplicity and being fast to compute, these direct approaches do not take measurement uncertainties into full consideration, thus rendering these methods vulnerable to noise. To improve robustness and accuracy, Dornaika and Horaud [1998] propose to jointly optimize rotation and translation with nonlinear optimization. The work of Strobl and Hirzinger [2006] proposes a metric on the special Euclidean group SE(3) and considered the relative weighting between rotation/translation components in the error metrics. Besides common least squares formulation, Zhao [2011] propose to formulate the problem with a L_∞ cost function and utilize convex optimization approach to solve it.

Besides the hand-eye calibration problem, there are also research focused on other types of sensor. For examples, the study of laser-camera calibration using motion-based method has been reported by Taylor and Nieto [2015]. Camera-odometry calibration is another example, which is a popular topic driven by the needs of information fusion for cars and wheeled mobile robots [Chang et al., 1993; Guo et al., 2012; Heng et al., 2013; S. Schneider et al., 2013]. To this end, Guo et al. [2012] proposed a two-step analytical least squares solution to estimate the rotation and translation separately, assuming only 2D in-plane motions are available. In contrast to the commonly used offline batch optimization, S. Schneider et al. [2013] reported an online recursive estimation approach for camera-odometry calibration, which is based on the Unscented Kalman filter.

7.1.3 Absolute-Motion-Based Methods

The third type of calibration methods, based on the model AX=YB, also often appears in the hand-eye calibration studies, e.g., [Dornaika and Horaud, 1998; Li et al., 2016; Tabb and Yousef, 2015; Wang, 1992; Zhuang et al., 1994], but they try to simultaneously estimate the hand-eye transformation (i.e. X) and the pose of the robot in the world (i.e. Y), using absolute positions of the sensors. Wang [1992] first submitted this formulation explicitly for hand-eye calibration. Li et al. [2016] addressed the same problem but assumes the pose measurements are asynchronous. Tabb and Yousef [2015] provided a solution based on parameterizing the rotation components using Euler angles, while Zhuang et al. [1994] used quaternion algebra to derive explicit linear solutions for X and Y. Dornaika and Horaud [1998] used quaternion algebra as well but also employed a nonlinearly optimization with two penalty functions.

7.1.4 Observability of Parameters

For calibration methods based on model AX=XB and AX=YB, care has to be taken with respect to the observability. Parameters become unobservable when the motions experienced by the sensors do not contain enough rotations in 3D, which leads to incomplete confinement to all the six (or twelve) transformation parameters during the estimation. Brookshire and Teller [2011] discuss the parameter observability in an algebraic way by inspecting the rank of the Fisher information matrix, while a more recent work by Maye et al. [2016] brings the observability analysis into practical use. They aim to separate the calibration parameters into observable and unobservable parts in real-time and update only those parameters that are observable during the optimization. They carry out the analysis in a numerical way by using rank-revealing QR and singular value decompositions of the Fisher information matrix.

7.1.5 Noise sensitivity analysis

Observability analysis can, however, merely state whether the parameter is observable or not. Our work on the noise sensitivity analysis in Chapter 3 can be seen as a step forward. The noise sensitivity analysis can not only provide the information about observability, but more importantly, quantify how the sensor and trajectory configuration relates to the calibration accuracy, given the existence of measurement noise. There are attempts to provide such a noise sensitivity analysis. Brookshire and Teller [2012] provided a formal discussion

on measurement uncertainties and their effects on the calibration result. They formulate a projected Gaussian noise model for unit dual quaternion SE(3) parameterization. By using unit dual quaternion with eight parameters, the rotation and translation are jointly represented, including their uncertainty. Maximum likelihood optimization is then carried out under this noise model, which provides a Cramer-Rao bound for the calibration's uncertainty. Their analysis is rigorous, however, covers only the model AX=XB. Dornaika and Horaud [1998] also performed a sensitivity analysis, but only for their methods and the linear method developed by Zhuang et al. [1994].

7.1.6 Summary

To the best of our knowledge, no other work exists to provide a systematic noise sensitivity analysis for all three types of method, nor work exists that bases the optimization for motion-based calibration on the Gauss–Helmert model, which jointly optimize the model parameter and measurements together in order to take the measurement noise into full account.

7.2 Camera-Laser Data Fusion

In the second part of the thesis, we focused on camera-laser fusion and contribute with two new visual-laser odometry algorithms. Previous work in visual-laser odometry can be categorized into two groups: *visual-odometry-based* approaches and *point-cloud-registration-based* approaches.

7.2.1 Visual-Odometry-Based Methods

Visual-odometry-based approaches try to apply a visual odometry pipeline with known pixel depth information coming from the laser scan. For example, the work of Shin et al. [2018] tries to solve the visual-laser SLAM problem within the direct sparse odometry (DSO) [Engel et al., 2018] framework. They use the projected laser points as feature points instead of using the salient gradient points extracted from the images. With the depth values of the feature points known and fixed, they perform a multi-frame photometric optimization the same as the DSO to estimate the poses of the keyframes. The work of J. Zhang et al. [2017] named depth enhanced monocular odometry has a similar framework.

RGB-D image registration [Della Corte et al., 2018; Kerl et al., 2013; Newcombe et al., 2011] tries to solve a similar problem, but they are using devices such

Figure 7.2: Microsoft Kinect, a RGB-D camera which can provide color (RGB) as well as depth (D) images.

as the Microsoft Kinect camera or stereo cameras, which can provide dense 3D information in the form of depth images along with regular RGB color information, as depicted in Figure 7.2. Numerous methods are tailored to such dense 3D measurements with large overlapping fields of view. Recent examples are KinectFusion by Newcombe et al. [2011], DVO by Kerl et al. [2013] and MPR by Della Corte et al. [2018]. Of which, the multi-cue photometric point cloud registration approach (MPR) [Della Corte et al., 2018] tries to jointly register color, depth, and normal information within a unified framework by considering the depth and normal information as channels of a multi-channel image. All three RGB-D registration methods utilize projective data association to speed up the registration process and to jointly exploit the depth and color cues.

However, a common problem for visual-laser odometry methods based on visual-odometry or RGB-D image registration is they are only applicable to the laser points that are visible in the camera image. Since these approaches do not consider the laser points that are outside the field of view of the camera, much of the range measurements will be discarded with sensors like Velodyne LiDARs, which can provide a 360 degrees scan. Such setting renders the system less accurate and vulnerable to texture-less scenes. In extreme cases, where there is no overlap between the fields-of-view of the scanner and the camera, these approaches will no longer work.

7.2.2 Point-Cloud-Registration-Based Methods

In contrast to that, the point cloud registration (ICP) based approaches try to align the whole point cloud with the help of image information in various aspects. For example, the methods by Pandey et al. [2011] and J. Zhang and Singh [2015] simply use the visual odometry result as an initial guess to the ICP process, making the ICP less likely to be trapped in local minima. A more advanced way to fuse the information is to use image/color information to guide and accelerate the data association process [Joung et al., 2009; Men et al., 2011; Naikal et al., 2009]. The works of Joung et al. [2009] and Men et al. [2011] treat the color information as the fourth channel input to the ICP, allowing a faster convergence rate than normal ICP as reported by Men et al. [2011]. The color information is not used in the error minimization process in Men's approach, which is in contrast to the work of Joung et al. [2009], whose error function incorporates the color consistency of matched points. They both use color data to filter out unlikely point candidates before ICP. The work of Naikal et al. [2009] achieves the same goal but employs a different strategy. The data association is established through image patch matching instead of using a k-d tree-based closest point assignment. They project scan points onto the respective images so that the 3D points can be associated to image patches around the projected location. A patch matching process is then carried out across images by minimizing a bidirectional sum of absolute differences. The resulting patch correspondences eventually determine the scan point correspondences. The visual odometry result is furthermore used to provide a search window for the patch matching process.

The common problem for aforementioned ICP based approaches is, however, due to the inevitable outlier point correspondences, the true solution may not necessarily locate at the exact minimum of the ICP cost function. This is especially the case when the point cloud is sparse. Furthermore, only the laser points with the improved correspondences are used to estimate the relative transformation. Therefore, to achieve better accuracy, it is necessary to optimize a joint objective that rewards both tight point cloud alignment (via a geometric term) as well as consistent image appearance (via a photometric term), to obtain better estimation accuracy.

7.2.3 Summary

To address the problems mentioned before, we proposed in Chapter 5 a novel visual-laser odometry approach that is able to work with general sensor configurations without requiring a field-of-view overlap, while still be able to exploits

the camera information to effectively constrain the ICP data association. And in Chapter 6, assuming the field-of-view overlap between the sensors is substantial, we proposed another visual-laser odometry approach that tries maximize the information usage of both sensor to achieve excellent estimation accuracy.

Chapter 8

Conclusion

Autonomous robots and vehicles often utilize a multi-sensor system to perform vital tasks such as localization or mapping. The joint system of various sensors with different sensing modalities is capable of providing better localization or mapping results than individual sensor alone in terms of accuracy, completeness or robustness. However, before any multi-sensor system can be put into use, two import matters have to addressed. Firstly, how to accurately determine the relative transformations (i.e. the spatial relationship) between individual sensors on the robot? This is a vital task known as extrinsic calibration. Without this calibration information, measurements from different sensors cannot be fused. Secondly, how to combine data from multiple sensors to exploit their respective sensing advantages and thus better solve the perception task? This is another important task known as data fusion. Both subjects have significant impacts on the performance of a multi-sensor system.

8.1 Summary

In this thesis, we focus on extrinsic calibration and camera-laser data fusion problems for multi-sensor systems. We addressed aspects related to improving the extrinsic calibration accuracy and presented novel data fusion algorithms for the ego-motion estimation problem using data from a laser scanner and a monocular camera.

We started with examining the relative calibration accuracies of three common types of calibration methods in Chapter 3. We performed a rigorous study on the noise sensitivity of each method from a novel geometric perspective. By quantifying and comparing the relative calibration accuracies of the three

methods, we are able to answer the question of "which method is better and why?". We are the first to offer such a systematic comparison and the result could give us an insight into choosing the best calibration method when multiple options are available. We then addressed the estimation problem of the common AX=XB type of calibration problem in Chapter 4. The main challenge in this context is how to obtain a statistically-optimal solution for the calibration parameters when working with an implicit constraint model such as the AX=XB transformation equation. We first pointed out an overlooked defect of common least squares approach in the context, and presented a better approach which can fully take into account the measurement uncertainties. Our approach utilizes the Gauss-Helmert paradigm to estimate not only the extrinsic parameters but also the pose observation errors, thus recovering the underlying sensor movements that exactly fulfill the motion constraints. Compared to traditional least squares approaches, our approach can provide statistically-optimal estimates that are more accurate and robust to noise. Besides, we considered not only the calibration of a sensor pair but also the case involving multiple sensors, allowing our approach to calibrate multiple sensors simultaneously. Our propgram is open sourced and can be accessed in https://github.com/PRBonn/extrinsic_calibration.

We also contributed, in the second part of the thesis, with two novel data fusion algorithms to address the ego-motion estimation problem, using complementary data from a laser scanner and a monocular camera. Our first algorithm in Chapter 5 exploits the advantages of both sensors in a such way that it is able to work in general conditions without requiring a field-of-view overlap between the camera and the laser scanner, nor an initial guess of the motion parameters. This is achieved by utilizing the 5-DoF relative orientation information estimated from image pairs through feature point correspondences. We demonstrated how to use such information to guide both the transformation-estimation and data-association steps within the ICP-based laser scan-matching process. The resulting approach can provide accurate trajectory estimates, which are better than those of each sensing modality alone. We presented our second algorithm in Chapter 6, which combines the camera and the laser scanner information in a direct way, assuming the field-of-view overlap between the sensors is substantial. By maximizing the information usage of both the sparse laser point cloud and the dense image, our algorithm is able to achieve excellent estimation accuracy. Several novel techniques are utilized in this algorithm. For examples, we explicitly address the occlusion problem using a prediction algorithm tailored to deal with sparse laser point clouds; we presented a method to address issues due to the sparsity of the laser measurements, by exploiting planar point sets from the laser data; We employed a homography formulation capable of incorporating the alignment of both dense planar image patches and sparse pixels for non-

planar point cloud; We presented a two-stage registration strategy which can combine the photometric and the geometric information while avoiding their respective pitfalls, i.e., local minima in photometric alignment and estimation bias in point cloud registration due to sparse point correspondences. Combining all these techniques, our approach is able to achieve state-of-the-art ego-motion estimation accuracy.

In summary, the approaches presented in this thesis allow us to answer the following questions:

- Which extrinsic calibration method is more accurate?
- How to obtain a statistically-optimal calibration result using relative-motion-based calibration method?
- How to accurately estimate the ego-motion of a robot using a laser scanner and a monocular camera?

8.2 Future Work

Despite the encouraging results presented in this thesis, there are promising areas via which this research could be continued.

One interesting research direction for extrinsic calibration is to develop hybrid calibration methods that combine model AX=B and model AX=XB, in other words, methods that utilize information from both calibration objects and sensor motions. This would be beneficial for sensors such as cameras and laser scanners. It could provide a way to solve both the applicability problem of model AX=B as well the observability problem of model AX=XB.

Regarding our camera-laser ego-motion approaches, one possible way of improving is to complete the pipeline by adding functions such as loop-closing, re-localization, pose-graph optimization backend, point cloud integration etc. These components were beyond the scope of this thesis but they are useful for real world applications.

Finally, the coding aspect of this work is currently not in a form that are production-ready. Work could be done to improve the quality of the code to allow non-expert users to make use of the system.

List of Figures

List of Tables

List of Algorithms

Bibliography

Agarwal, S., K. Mierle, et al. (2010). *Ceres Solver*. http://ceres-solver.org.

Amiri-Simkooei, A. (2007). "Least-squares variance component estimation: theory and GPS applications". PhD thesis. TU Delft, Delft University of Technology.

Atlas, the World's Most Dynamic Humanoid (2018). URL: https://www.bostondynamics.com/atlas (visited on 10/10/2018).

Bentley, J.L. (1975). "Multidimensional Binary Search Trees Used for Associative Searching". In: *Communications of the ACM* 18.9, pp. 509–517.

Besl, P.J. and N.D. McKay (1992). "A Method for Registration of 3-d Shapes". In: *IEEE Transactions on Pattern Analalysis and Machine Intelligence (TPAMI)* 14.2, pp. 239–256.

BigDog, the First Advanced Rough-Terrain Robot (2018). URL: https://www.bostondynamics.com/bigdog (visited on 10/10/2018).

Brookshire, J. and S. Teller (2011). "Automatic calibration of multiple coplanar sensors". In: *Proceedings of Robotics: Science and Systems (RSS)*. Vol. 33.

Brookshire, J. and S. Teller (2012). "Extrinsic Calibration from Per-Sensor Egomotion". In: *Proceedings of Robotics: Science and Systems (RSS)*.

Carrera, G., A. Angeli, and A.J. Davison (2011). "SLAM-Based Automatic Extrinsic Calibration of a Multi-Camera Rig". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Chang, Y., X. Lebegue, and J.K. Aggarwal (1993). "Calibrating a mobile camera's parameters". In: *Pattern Recognition* 26.1, pp. 75–88.

Condurache, D. and A. Burlacu (2016). "Orthogonal dual tensor method for solving the AX=XB sensor calibration problem". In: *Mechanism and Machine Theory* 104, pp. 382–404.

Corsini, M., M. Dellepiane, F. Ponchio, and R. Scopigno (2009). "Image-to-Geometry Registration: a Mutual Information Method exploiting Illumination-related Geometric Properties". In: *Computer Graphics Forum* 28.7, pp. 1755–1764.

Curiosity Rover (2018). URL: https://www.nasa.gov/mission_pages/msl/overview/index.html (visited on 10/10/2018).

Daniilidis, K. (1999). "Hand-eye calibration using dual quaternions". In: *International Journal of Robotics Research (IJRR)* 18, pp. 286–298.

Della Corte, B., I. Bogoslavskyi, C. Stachniss, and G. Grisetti (2018). "A General Framework for Flexible Multi-Cue Photometric Point Cloud Registration". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Dornaika, F. and R. Horaud (1998). "Simultaneous robot-world and hand-eye calibration". In: *IEEE Transactions on Robotics and Automation (TRA)* 14.4, pp. 617–622.

Engel, J., V. Koltun, and D. Cremers (2018). "Direct Sparse Odometry". In: *IEEE Transactions on Pattern Analalysis and Machine Intelligence (TPAMI)* 40.3, pp. 611–625.

Engel, J., T. Schöps, and D. Cremers (2014). "LSD-SLAM: Large-scale direct monocular SLAM". In: *Proceedings of the Europ. Conference on Computer Vision (ECCV)*, pp. 834–849.

Fassi, I. and G. Legnani (2005). "Hand to sensor calibration: A geometrical interpretation of the matrix equation AX=XB". In: *Journal of Robotic Systems* 22, pp. 497–506.

Faugueras, O.D. and G. Toscani (1989). "The Calibration Problem for Stereoscopic Vision". In: *Sensor Devices and Systems for Robotics*, pp. 195–213.

Fernndez-Moral, E., V. Arevalo, and J. Gonzlez-Jimnez (2015). "Extrinsic Calibration of a Set of 2D Laser Rangefinders". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Förstner, W. and B. Wrobel (2016). *Photogrammetric Computer Vision – Statistics, Geometry, Orientation and Reconstruction*. Springer Verlag.

Geiger, A., P. Lenz, and R. Urtasun (2012a). "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3354–3361.

Geiger, A., F. Moosmann, M. Car, and B. Schuster (2012b). "Automatic Camera and Range Sensor Calibration using a single Shot". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Gomez-Ojeda, R., J. Briales, E. Fernndez-Moral, and J. Gonzlez-Jimnez (2015). "Extrinsic Calibration of a 2D Laser-Rangefinder and a Camera Based on Scene Corners". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Gower, J.C., G.B. Dijksterhuis, et al. (2004). *Procrustes problems*. Vol. 30. Oxford University Press on Demand.

Guo, C.X., F.M. Mirzaei, and S.I. Roumeliotis (2012). "An analytical least-squares solution to the odometer-camera extrinsic calibration problem". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Hartley, R., J. Trumpf, Y. Dai, and H. Li (2013). "Rotation averaging". In: *International Journal of Computer Vision (IJCV)* 103, pp. 267–305.

Heng, L., M. Brki, G.H. Lee, P.T. Furgale, R. Siegwart, and M. Pollefeys (2014). "Infrastructure-Based Calibration of a Multi-Camera Rig". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Heng, L., B. Li, and M. Pollefeys (2013). "Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Huang, K.H. and C. Stachniss (2017). "Extrinsic Multi-Sensor Calibration For Mobile Robots Using the Gauss-Helmert Model". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Huang, K.H. and C. Stachniss (2018a). "Joint Ego-motion Estimation Using a Laser Scanner and a Monocular Camera Through Relative Orientation Estimation and 1-DoF ICP". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Huang, K.H. and C. Stachniss (2018b). "On Geometric Models and Their Accuracy for Extrinsic Sensor Calibration". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Huang, K.H. and C. Stachniss (2019). "Accurate Direct Visual-Laser Odometry with Explicit Occlusion Handling and Plane Detection". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Joung, J.H., K.H. An, J.W. Kang, M.J. Chung, and W. Yu (2009). "3D environment reconstruction using modified color ICP algorithm by fusion of a camera and a 3D laser range finder". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Kerl, C., J. Sturm, and D. Cremers (2013). "Robust Odometry Estimation for RGB-D Cameras". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3748–3754.

Khosravian, A., T. Chin, and I. Reid (2017). "A Branch-And-Bound Algorithm for Checkerboard Extraction in Camera-Laser Calibration". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Klein, G. and D. Murray (2007). "Parallel Tracking and Mapping for Small AR Workspaces". In: *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*.

Kümmerle, R., G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard (2011). "g2o: A general framework for graph optimization". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3607–3613.

Li, H., Q. Ma, T. Wang, and G. Chirikjian (2016). "Simultaneous Hand-Eye and Robot-World Calibration by Solving the AX=YB Problem without Correspondence". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Maye, J., H. Sommer, G. Agamennoni, R. Siegwart, and P. Furgale (2016). "Online self-calibration for robotic systems". In: *International Journal of Robotics Research (IJRR)* 35, pp. 357–380.

Men, H., B. Gebre, and K. Pochiraju (2011). "Color Point Cloud Registration with 4D ICP Algorithm". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Mur-Artal, R., J.M.M. Montiel, and J. D Tardos (2015). "ORB-SLAM: a versatile and accurate monocular SLAM system". In: *IEEE Transactions on Robotics (TRO)* 31.5, pp. 1147–1163.

Naikal, N., J. Kua, G. Chen, and A. Zakhor (2009). "Image augmented laser scan matching for indoor dead reckoning". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Neitzel, F. and B. Schaffrin (2016). "On the Gauss–Helmert model with a singular dispersion matrix where BQ is of smaller rank than B". In: *Journal of computational and applied mathematics* 291, pp. 458–467.

Newcombe, R.A., S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon (2011). "KinectFusion: Real-Time Dense Surface Mapping and Tracking". In: *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 127–136.

Nistér, D. (2004). "An efficient solution to the five-point relative pose problem". In: *IEEE Transactions on Pattern Analalysis and Machine Intelligence (TPAMI)* 26.6, pp. 756–770.

Olson, E.B. (2009). "Real-Time Correlative Scan Matching". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4387–4393.

Olson, E.B. (2011). "AprilTag: A Robust and Flexible Visual Fiducial System". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Pandey, G., J.R. McBride, S. Savarese, and R.M. Eustice (2011). "Visually Bootstrapped Generalized ICP". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Pandey, G., J.R. McBride, S. Savarese, and R.M. Eustice (2015). "Automatic Extrinsic Calibration of Vision and Lidar by Maximizing Mutual Information". In: *Journal of Field Robotics (JFR)* 32, pp. 696–722.

Park, F.C. and B.J. Martin (1994). "Robot sensor calibration: solving AX=XB on the Euclidean group". In: *IEEE Transactions on Robotics and Automation (TRA)* 10, pp. 717–721.

Pluim, J.P.W., J.B.A. Maintz, and M.A. Viergever (2003). "Mutual-information-based registration of medical images: a survey". In: *IEEE Trans. on Medical Imaging* 22.8, pp. 986–1004.

Pomerleau, F., F. Colas, and R. Siegwart (2015). "A Review of Point Cloud Registration Algorithms for Mobile Robotics". In: *Foundations and Trends in Robotics* 4, pp. 1–104.

Rodriguez, O (1840). "Des lois géometriques qui regissent les deplacements d'un système solide independament des causes qui peuvent les produire". In: *Journal de mathématiques pures et appliquées* 5, pp. 380–440.

Rusinkiewicz, S. and M. Levoy (2001). "Efficient variants of the ICP algorithm". In: *Proc. of Int. Conf. on 3-D Digital Imaging and Modeling*, pp. 145–152.

Rwekmper, J., M. Ruhnke, B. Steder, W. Burgard, and G.D. Tipaldi (2015). "Automatic Extrinsic Calibration of Multiple Laser Range Sensors with Little

Overlap". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Scaramuzza, D., A. Harati, and R. Siegwart (2007). "Extrinsic Self Calibration of a Camera and a 3D Laser Range Finder from Natural Scenes". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Schenk, K., A. Kolarow, M. Eisenbach, K. Debes, and H. Gross (2012). "Automatic Calibration of a Stationary Network of Laser Range Finders by Matching Movement Trajectories". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Schneider, J., C. Eling, L. Klingbeil, H. Kuhlmann, W. Förstner, and C. Stachniss (2016). "Fast and Effective Online Pose Estimation and Mapping for UAVs". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4784–4791.

Schneider, S., T. Luettel, and H. Wuensche (2013). "Odometry-based online extrinsic sensor calibration". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1287–1292.

Segal, A., D. Haehnel, and S. Thrun (2009). "Generalized-ICP". In: *Proceedings of Robotics: Science and Systems (RSS)*.

Self-driving technology (2018). URL: https://waymo.com/ (visited on 10/10/2018).

Serafin, J. and G. Grisetti (2015). "NICP: Dense Normal Based Point Cloud Registration". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 742–749.

Shannon, C.E. (1948). "A mathematical theory of communication". In: *Bell System Technical Journal* 27.3, pp. 379–423.

Shin, Y., Y.S. Park, and A. Kim (2018). "Direct Visual SLAM Using Sparse Depth for Camera-LiDAR System". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Shiu, Y.C. and S. Ahmad (1989). "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX=XB". In: *IEEE Transactions on Robotics and Automation (TRA)* 5, pp. 16–29.

Strobl, K.H. and G. Hirzinger (2006). "Optimal Hand-Eye Calibration". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4647–4653.

Tabb, A. and K.A. Yousef (2015). "Parameterizations for Reducing Camera Reprojection Error for Robot-World Hand-Eye Calibration". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Taylor, Z. and J. Nieto (2015). "Motion-based calibration of multimodal sensor arrays". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4843–4850.

Taylor, Z., J. Nieto, and D. Johnson (2013). "Automatic Calibration of Multi-Modal Sensor Systems Using a Gradient Orientation Measure". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Taylor, Z., J. Nieto, and D. Johnson (2015). "Multi-Modal Sensor Calibration Using a Gradient Orientation Measure". In: *Journal of Field Robotics (JFR)* 32, pp. 675–695.

Umeyama, S. (1991). "Least-squares estimation of transformation parameters between two point patterns". In: *IEEE Transactions on Pattern Analalysis and Machine Intelligence (TPAMI)* 13.4, pp. 376–380.

Wang, C.C. (1992). "Extrinsic calibration of a vision sensor mounted on a robot". In: *IEEE Transactions on Robotics and Automation (TRA)* 8.2, pp. 161–175.

Weingarten, J., G. Gruener, and R. Siegwart (2003). "A fast and robust 3d feature extraction algorithm for structured environment reconstruction". In: *Proceedings of the International Conference on Advanced Robotics (ICAR)*, pp. 390–397.

Wolf, H. (1978). "Das geodätische Gauß-Helmert-Modell und seine Eigenschaften". In: *Zeitschrift for Vermessungswesen (103)*, 103:41–43.

Xiao, J.H., J.H. Zhang, J.W. Zhang, H.X. Zhang, and H.P. Hildre (2011). "Fast plane detection for SLAM from noisy range images in both structured and unstructured environments". In: *the International Conference on Mechatronics and Automation (ICMA)*, pp. 1768–1773.

Zhang, J., M. Kaess, and S. Singh (2017). "A real-time method for depth enhanced visual odometry". In: *Autonomous Robots* 41, pp. 31–43.

Zhang, J. and S. Singh (2015). "Visual-Lidar Odometry and Mapping: Low-Drift, Robust, and Fast". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Zhang, J. and S. Singh (2017). "Low-drift and real-time lidar odometry and mapping". In: *Autonomous Robots* 41, pp. 401–416.

Zhang, Z. (1994). "Iterative point matching for registration of free-form curves and surfaces". In: *International Journal of Computer Vision (IJCV)* 13.2, pp. 119–152.

Zhao, Z. (2011). "Hand-Eye Calibration Using Convex Optimization". In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2947–2952.

Zhuang, H.Q., Z.S. Roth, and R. Sudhakar (1994). "Simultaneous robot/world and tool/flange calibration by solving homogeneous transformation equations of the form AX=YB". In: *IEEE Transactions on Robotics and Automation (TRA)* 10.4, pp. 549–554.

Zisserman, A., P.A. Beardsley, and I.D. Reid (1995). "Metric calibration of a stereo rig". In: *Proc. of the IEEE Workshop on Representation of Visual Scenes*, pp. 93–100.