Automatic calibration of ship-mounted camera extrinsics

Daniel Bjerkehagen

Advisor: Edmund Brekke
Co-advisors: Esten Ingar Grøtli, Johannes Tjønnås

December 2022

Abstract

With increasing adoption of autonomous vehicles, so does the safety requirements of such systems increase. An essential factor in guaranteeing safe operations is good estimates of the extrinsic parameters of sensors, that being the position and orientation of the sensor. However, most of current calibration methods are costly both in time and money. Additionally, the orientation of a sensor may change slightly over time or with differing temperatures and levels of humidity. Common methods for data-driven extrinsic calibration often require infrastructure like calibration plates or use algorithms only applicable to that specific sensor. By developing an algorithm for automatically calibrating the extrinsic parameters of cameras this project tests combining Structure from Motion-algorithms with Hand-Eye Calibration solvers for the purpose of providing as-good-as-human accuracy for the orientation of ship-mounted cameras, while at the same time keeping the complexity of the problem formulation low. The novel pipeline and tests are implemented in Python. The problem of estimating the full extrinsics was simplified to only the orientation, due to the position of ship-mounted sensors generally being known with higher accuracy than the orientation. Through tests with both synthetic and real-world data, an ability to discern the orientation of cameras with high precision is demonstrated. The results are achieved requiring only ship poses and image streams as input to the developed method, and the algorithm demonstrates some robustness in simulations. The report describes the mathematical requirements of the input data for the Hand-Eye Calibration solvers to converge, and documents how the negative effects of failing to meet these requirements can be mitigated in the specific case of ship-mounted cameras. Discussion on the different parts of the algorithm pipeline concludes that iterative optimization with a cost-function inspired by the work of Park and Martin to yield best results, but more in-depth analyses should be performed to strengthen this finding. Further work on the topic of optimal data selection is motivated and discussed.

Sammendrag

Med økende bruk av autonome fartøy vil sikkerhetskravene til slike systemer også øke. En viktig faktor for å kunne garantere sikker drift er gode estimater av "extrinsic" kalibreringsparametre, hvilket beskriver posisjonen og orienteringen av sensorer på fartøy. De fleste metodene brukt i dag for å finne disse parametrene tar lang tid og er kostnadsrike. I tillegg kan orienteringen til sensorer endre seg over tid og som følge av endringer i temperatur eller luftfuktig. Dette motiverer bruken av metoder som utfører kalibreringen fortløpende, men eksisterende metoder for datadrevet kalibrering av "extrinsic" parametre krever spesiell infrastruktur eller bruker algoritmer som kun kan brukes på én spesifikk type sensor. Gjennom å utvikle en algoritme for å automatisk kalibrere "extrinsic" parametre for kamera montert på skip tester dette prosjektet kombinasjonen av "Structure from Motion"-algoritmer med metoder for å utføre "Hand-Eye Calibration", med mål om å oppnå estimater for kameraenes orientering med lik presisjon som manuell måling oppnår. Dette gjøres samtidig som metodikken holdes så enkel som mulig. Algoritmen og tester er implementert i Python. Valget om å forenkle problemstillingen til å kun estimere orientering ble begrunnet ved at posisjonen til sensorer på skip ofte er kjent med mye høyere nøyaktighet enn orienteringen. Gjennom tester som bruker både syntetisk og ekte data viser algoritmen en evne til å estimere kameras orientering med høy presisjon. Algoritmen krever kun skipsdata og bildestrømmer som inn-verdier for å oppnå resultatene, og den viser noe robusthet i simulasjoner. Rapporten gir matematiske krav som dataen må oppfylle for at "Hand-Eye Calibration"-løserene skal konvergere, og dokumenterer hvordan konsekvensene som følger når data ikke oppfyller disse kan minskes. Designvalg på de forskjellige delene av algoritmen diskuteres og det konkluderes med at iterativ optimering av en kostnadsfunksjon inspirert av arbeid av Park og Martin gir best resultat. Forslag til videre arbeid på problemstillingen om optimalt valg av data diskuteres og motiveres.

Preface

This project has been completed as a part of my 5-year masters program at the Department of Engineering Cybernetics, at the Norwegian University of Science and Technology (NTNU) and in cooperation with SINTEF.

I would like to thank my supervisors for valuable insight, especially on the topic of scientific writing, as well as believing in my ideas. A special thanks is also given to Torbjørn Barheim and his collegues at Kongsberg Maritime Seatex for their help with acquiring data. The project would also not be possible without SFI Autoship and the accesibility to data and cooperation it provided for this project.

Contents

Acronyms

HT Homogeneous Transform. 4–6, 9, 17

KM Kongsberg Maritime. 24, 25, 28, 32, 34, 35, 40, 44–46

NED North-East-Down. 16–18

NTNU Norwegian University of Science and Technology. vii, 24

SfM Structure from Motion. 19, 20, 48

VO Visual Odometry. 19

VSLAM Visual Localization And Mapping. 19, 20, 43, 48

Chapter 1

Introduction

Of the research being done on autonomous systems, the case of autonomous ships has shown to be both worthwhile academically and strongly motivated by the industry. Norway, with its rich history of seafaring, has seen several recent projects on semi- or fully autonomous ships in use in the industry. Yara Birkeland, Asko Maritime and the startup Zeabuz are examples of such ventures. While the radar historically has been one of the most important sensors in safe marine operations, recent advances in computer vision have enabled the use of the dense camera-data in autonomous systems.

The high amount of research on situational-awareness and robots which model their surroundings, as well as methods in autonomy applying sensor-data to make decisions, point to the importance of accurate calibration of these sensor-systems. All types of sensors have specific sets of parameters which must be calibrated with methods specific to that type of sensor. One important set of calibration-parameters relevant to all use-cases and calibrated more or less the same way for all sensors are the extrinsic parameters, meaning the relative position and orientation of the sensors. The extrinsic parameters are key to successfully interpreting sensor-data in a way that represents reality. To illustrate its importance, consider an autonomous car with side-mounted distance sensors. If the sensors are angled differently than assumed, the control system could end up interpreting a dangerously close vehicle as being at a safe distance.

One popular method for performing calibration of extrinsic parameters for a camera mounted on a robot arm involves solving the *Hand-Eye Calibration problem*. This is a mathematical equation for which the unknown parameter is the extrinsic calibration [1]. The formulation is simple and concise, being based only on a handful of mathematical equalities. Research has shown some Hand-Eye solvers able to get estimates of the extrinsic parameters as close as within 0.1° and 2 mm of the ground truth parameters in optimal controlled experiments [2]. The problem was first formulated in 1989, and research has since then mostly focused on methods to solve the problem with higher accuracy or lower runtime [2, 3]. The mathematical properties required of input data to yield the extrinsics observable has been known for quite some time [1, 4], but authors provide mostly general

guidelines rather than optimal strategies for selecting data in large datasets. Further, most research on the topic relies on the use of geometric calibration targets to reconstruct the camera motion to the correct scale.

The problem of estimating camera motion is a problem of interest in computer vision, and multiple algorithms exist for reconstructing the motion of a camera even when a calibration target is unavailable [5]. Such methods often rely on feature-detection and -tracking algorithms, and have become widely successful due to a number of impressive results like SLAM and VO.

Methods exist for combining structureless camera motion estimation with the Hand-Eye calibration formulation, notably the method developed by Andreff *et al.* [4], but most other Hand-Eye solvers assume scale of the camera motion is known. For the case of ship-mounted sensors, work has been done on automatically finding the extrinsics of the camera relative a sonar [6]. Roy *et al.* [7] use sensor egomotion reconstruction from a vessel with planar movement to estimate the extrinsic parameters through maximum a posteriori estimation method. To the authors knowledge, no research exists on the topic of using a camera egomotion reconstruction algorithm as data-baseline for extrinsic calibration of ship-mounted sensors using the Hand-Eye Calibration problem formulation.

By identifying the components of the Hand-Eye calibration problem with measurements available when cameras are rigidly mounted on ships, this project tests a novel algorithm pipeline for estimating the camera extrinsics using only images and measurements of the ship's position and attitude as input to the algorithm. The presented pipeline is thereby purely data-driven. The report also addresses some challenges when using ship-data for Hand-Eye Calibration, and a qualitative way of measuring the excitation in data for the purpose of Hand-Eye Calibration is presented.

In many applications, including marine operations, the position of a sensor is known with high precision due to extensive surveys done both before launch and during the lifetime of the ship. Finding good estimates for the orientation of the sensor, however, is not as easy. Due to this fact, and the fact that the developed pipeline works better on pure orientation-estimation, a simplification of the problem is made by not considering calibrating the position of the sensor.

Chapter 2

Theory

2.1 Basic mathematical concepts and objects

2.1.1 Frames and Conventions

Coordinate frames are used to represent orientations of rigid object as well as describing the rotation of vectors between multiple rigid objects. For 3-dimensional space defining the coordinate system A is done by defining three orthonormal vectors, or axes, $(\mathbf{x}_A, \mathbf{y}_A, \mathbf{z}_A)$ centered at an origin $\mathcal{O}_A$. With this, any point may be defined relative frame A as a unique linear combinations of the three axes.

Further, if three new orthonormal vectors are defined as a linear combinations of the coordinate axes of frame A one may define a second coordinate frame. Naming this system B, defining its origin $\mathcal{O}_B^A$ and collecting its axes into the columns of a matrix $\mathbf{R}_{AB}$ as in Equation (2.1), orientation and position of B relative A has successfully defined numerically.

$$\mathbf{R}_{AB} = \begin{bmatrix} | & | & | \\ \mathbf{x}_B^A & \mathbf{y}_B^A & \mathbf{z}_B^A \\ | & | & | \end{bmatrix} \tag{2.1}$$

Notation

In this project, the following notation is adherred to when it comes to the notation of coordinate frames and similar mathematical objects.

- Any non-scalar object is given in bold. $\mathbf{v}, \mathbf{A}$.
- Vectors are written in lowercase, matrices in uppercase.
- The coordinate system for which a vector is defined in is superscripted. $\mathbf{v}^a$
- Coordinate transforms are given on the form $\mathbf{H}_{ab}$, being understood as either a matrix finding the coordinate expression in coordinate system "a" of a vector given in coordinate system "b", or as the pose of coordinate system "b" relative system "a".

- TThe angle-axis representation of orientations is denoted with θ as the angle, $\mathbf{a}$ as the unit-norm axis, and $\boldsymbol{\omega} = \theta\mathbf{a}$.

The following describes different ways to define commonly used coordinate frames which are relevant for this project.

The Body Frame

For vessels a common practice when defining a coordinate system rigidly attached to the ship is to define the X-axis pointing foreward along the bow, the Z-axis to be pointed downwards and the Y-axis to complete the right-handed coordinate system [8]. This coordinate frame is simply dubbed the *body frame*.

The Camera Frame

Some users [9–11] prefer to define the Z-axis of cameras to point along the optical axis, the Y-axis to point downwards along the camera body and the X-axis to complete the right-handed system. Others, however, prefer to have the X-axis be pointed along the optical axis, the Z-axis pointing upwards, and the Y-axis thereafter.

Naming the conventions "A" (Z along optical axis, Y down) and "B" (X along optical axis, Z up) respectively, Equation (2.2) relates the two through a rotation matrix.

$$\mathbf{R}_{\text{AB}} = \begin{bmatrix} \mathbf{x}_{\text{B}}^{\text{A}} & \mathbf{y}_{\text{B}}^{\text{A}} & \mathbf{z}_{\text{B}}^{\text{A}} \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix} \tag{2.2}$$

For marine operations where the body-coordinate system often is defined with the X-axis foreward and the Z-axis downwards, some might find it intuitive to define the camera coordinate frame equivalently. Therefore, a third convention is to have the X-axis points along the optical axis, the Z-axis pointed downwards and the Y-axis completing the coordinate system. The transformation relating this convention, "C", and convention "A" is given in Equation (2.3). An illustration of all three camera frame conventions is given in Figure 2.1.

$$\mathbf{R}_{\text{AC}} = \begin{bmatrix} \mathbf{x}_{\text{C}}^{\text{A}} & \mathbf{y}_{\text{C}}^{\text{A}} & \mathbf{z}_{\text{C}}^{\text{A}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \tag{2.3}$$

2.1.2 Homogeneous Transforms

Pairing a rotation matrix and a translation vector allows for representing the pose of an object. Such a pair may be collected into a 4×4 real matrix acting on homogeneous coordinate vectors, at which point the matrix is called a Homogeneous Transform (HT). This is commonly used as representation of pose in Computer

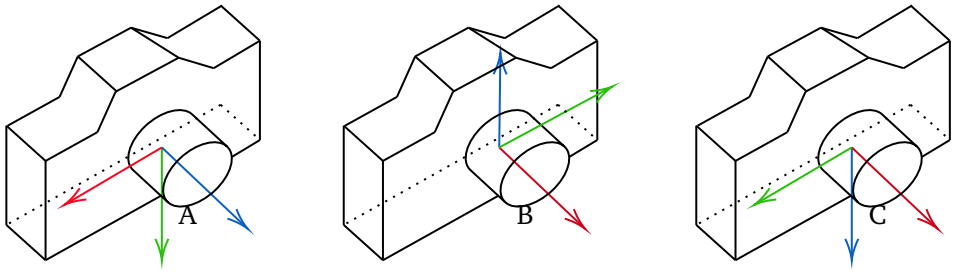


Figure 2.1: Three common conventions for defining a coordinate frame rigidly attached to a camera. In the figure, standard coloring of the axes as red=x, green=y and blue=z is used.

Vision [10, 11]. Given an orientation and position of some coordinate frame "b" relative the coordinate frame "w", $\mathbf{R}_{wb}$ and $\mathbf{t}_{wb}$, the Homogeneous Transform matrix is constructed as in Equation (2.4). The inverse of a Homogeneous Transform matrix is given as Equation (2.5).

$$\mathbf{H}_{wb} = \begin{bmatrix} \mathbf{R}_{wb} & \mathbf{t}_{wb} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{2.4}$$

$$\mathbf{H}_{wb}^{-1} = \mathbf{H}_{bw} = \begin{bmatrix} \mathbf{R}_{wb}^{T} & -\mathbf{R}_{wb}^{T}\mathbf{t}_{wb} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{2.5}$$

Poses are elements of the rigid motion group SE(3), and therefore HTs are a representation of SE(3)-elements. These elements, in addition to being interpreted as objects in themself, can also represent an action over vectors. The action of HTs over vectors transform the vectors from being defined relative one frame to being defined relative another, see Equation (2.6).

$$\mathbf{H}_{wb} \cdot \mathbf{p}^{b} = \mathbf{R}_{wb}\mathbf{p}^{b} + \mathbf{t}_{wb} = \mathbf{p}^{w} \tag{2.6}$$

2.1.3 Relative pose

The concept of a *relative pose* is used extensively throughout this project to describe data and its properties. Relative pose should be understood as the following. Let $\mathbf{H}_{na}$ and $\mathbf{H}_{nb}$ be HTs describing the pose of two different frames, "a" and "b", relative the same coordinate frame, "n". Frame "n" is described as a *reference frame* to the other frames. The relative movement of *b relative a* is then computed as Equation (2.7).

$$\begin{aligned} \mathbf{H}_{ab} &= \mathbf{H}_{na}^{-1}\mathbf{H}_{nb} \\ &= \mathbf{H}_{an}\mathbf{H}_{nb} \end{aligned} \tag{2.7}$$

Figure 2.2 illustrates the interpretation of relative pose, as defined in this project. The reader should note that though most authors prefer to mainly consider

HTs as the action on vectors, and therefore reverse the black arrows in Figure 2.2 to signify how the HTs transforms vectors between frames, the reverse interpretation of poses as transformations of the coordinate axes is employed in this project.

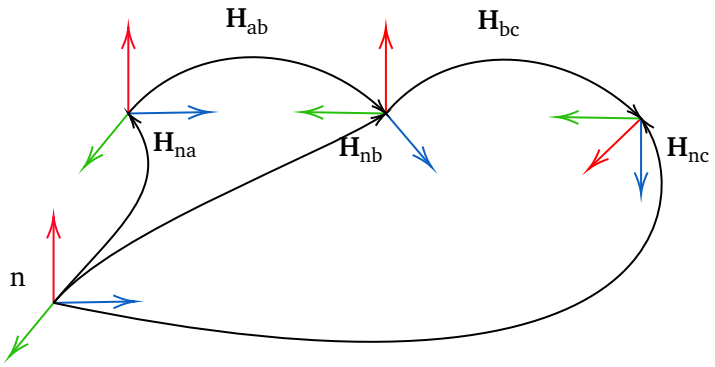


Figure 2.2: The relative pose between three coordinate frames, and their relationship with the reference frame

2.1.4 The SO(3) group: Properties and operations

Poses are made of a translation vector and an orientation, and the set of all orientations also form a group, name SO(3). As shown in Section 2.1.1, valid orientations may be construted by defining 3 orthonormal axes, but multiple simpler parametrizations of SO(3) ezist. Most notably is the angle-axis representation and Euler-angles [8]. For a matrix to be a orientation it is required that its columns are orthonormal, a constraint which may be formulated as Equation (2.8).

$$\mathbf{R}\mathbf{R}^{\mathrm{T}} = \mathbf{I}_{3\times 3} \tag{2.8}$$

Since orientations may be represented by elements of $\mathbb{R}^{3\times 3}$ that satisfy a constraint this means the rotation matrices are elements of a manifold on $\mathbb{R}^{3\times 3}$ defined by said constraint. The constraint is differentiable, which classifies SO(3) as a *Lie group* [12].

Lie groups are unique in that their nature allows them to be represented by elements of the tangent space of the group at the identity element, called the *Lie algebra*. The Lie algebra is a vector space, and transforming between the Lie group, the Lie algebra and its corresponding n-dimensional vector space is denoted by the symbols given in Equations (2.9) to (2.12) [12]. Here, $\mathfrak{g}$ is the Lie algebra of the group $\mathcal{G}$.

$$(\cdot)^{\wedge} : \mathbb{R}^n \to \mathfrak{g} \tag{2.9}$$
$$(\cdot)^{\vee} : \mathfrak{g} \to \mathbb{R}^n \tag{2.10}$$
$$\exp : \mathfrak{g} \to \mathcal{G} \tag{2.11}$$
$$\log : \mathcal{G} \to \mathfrak{g} \tag{2.12}$$

Transforming between the vector space and the group allows for certain computations to be performed more easily in the vector space which then are transformed into the group, instead of attempting to operate directly on group-elements. For this reason, two further transformations seen in Equations (2.13) and (2.14) are defined as short-hand transformations directly between the vector space and the group.

$$\mathrm{Exp} : \mathbb{R}^n \to \mathcal{G}, \;\; \mathrm{Exp}(\mathbf{a}) = \exp(\mathbf{a}^{\wedge}) \tag{2.13}$$
$$\mathrm{Log} : \mathcal{G} \to \mathbb{R}^n, \;\; \mathrm{Log}(\mathbf{R}) = \log(\mathbf{R})^{\vee} \tag{2.14}$$

Interestingly, elements of the Lie algebra of SO(3) are excactly the rotation axes of those orientations.

Metrics are functions defined over some set which allows for a notion of *closeness* between two elements of the set. Defining a metric over SO(3) is then useful when comparing e.g. an estimated orientation against the true value. Many metrics may be defined over this group, but one metric of particular geometric interpretation is the one used in [13], restated in Equation (2.15). Here, $||\cdot||_2$ is the length of a vector.

$$d(\mathbf{A}, \mathbf{B}) = ||\log(\mathbf{A}^T\mathbf{B})^{\vee}||_2, \;\; \mathbf{A}, \mathbf{B} \in \mathrm{SO}(3) \tag{2.15}$$

The metric may be understood as the angle of the shortest rotation connecting the orientations **A** and **B**.

2.2 Egomotion estimation algorithms

Multiple algorithms exist in computer vision which produce an estimate of the movement of a camera given an ordered set of pictures. Some of the approaches to estimate camera motion include Simultaneous Localization and Mapping (SLAM), Visual Odometry (VO), and Structure from Motion (SfM). These methods are unified under the term "camera egomotion estimation" [5]. Egomotion algorithms often produce, in addition to the camera motion, an estimate of the geometric structure in the scene captured by the images. What follows is a short summary of the techniques enabling these methods.

Estimating egomotion is often based on tracking points or parts of an image between subsequent frames for which the same point or part is visible. This creates a distinction between what is called "direct" and "indirect" methods. Direct

methods are categorized by tracking all or nearly all pixels of each image to estimate camera motions. Indirect methods first extract geometric "features" and then track the movement of these between pictures [14].

Given the tracks of points across multiple images, the next step is often to estimate the camera motion based on these tracks. Assuming the observed points are static relatibve the environment allows for geometric and numeric methods for estimating the camera motion. Examples include using the properties of projective cameras to restrict the set of possible camera motions given the observed tracks by applying "epipolar geometry", as well as finding the optimal linear movement given two closely related images [15].

Lastly, most methods refine the initial camera egomotion estimates through softly enforcing some sort of constraint, for instance minimizing the reprojection error in a bundle adjustment scheme or requiring points to have static positions in the reference frame [14].

2.3 Hand-Eye Calibration

2.3.1 History

The "Hand-Eye Calibration problem" originates in robotics, being the issue of finding how a sensor, often a camera, is mounted rigidly relative an end effector. The problem is often attributed to be studied first by Shiu *et al.* in 1989 [1]. Since then many papers have been written on different solution techniques to recover the extrinsic parameters [4, 13, 16], with research mostly focusing on improving the accuracy given better computational power. Today, numerical solutions to the problem are implemented in open-source software like OpenCV, as well as commercial products like Zivid's calibration library.

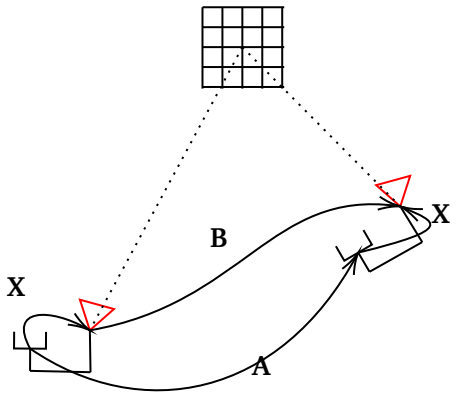


Figure 2.3: Conceptual illustration of the Hand-Eye Calibration problem. The setup consists of a camera (red), an end-effector and a calibration plate. In the illustration, the rigid system undergoes some controlled motion.

To understand the mathematical fomulation in Section 2.3.2, consider Figure 2.3. In the original formulation the system is attached to a robotic arm, allowing for precise movement of the end effector. The setup is moved between predetermined poses and pictures are taken of a stationary calibration target at each pose. Employing a geometric algorithm, like the 8-point algorithm [17], the camera movement between each picture is recovered. By combining knowledge of the "hand"-movement with the "eye"-movement, the mathematical equivalence presented in Section 2.3.2 may be made.

2.3.2 Mathematical formulation

Observing Figure 2.3, it can be shown that the geometry of the setup allows for the equality in Equation (2.16) to be made.

$$\mathbf{AX} = \mathbf{XB} \tag{2.16}$$

$$\begin{bmatrix} \mathbf{R}_A & \mathbf{t}_A \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix}\begin{bmatrix} \mathbf{R}_X & \mathbf{t}_X \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_X & \mathbf{t}_X \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix}\begin{bmatrix} \mathbf{R}_B & \mathbf{t}_B \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} \tag{2.17}$$

$$\sim$$

$$\mathbf{R}_A\mathbf{R}_X = \mathbf{R}_X\mathbf{R}_B \tag{2.18}$$

$$\mathbf{R}_A\mathbf{t}_X + \mathbf{t}_A = \mathbf{R}_X\mathbf{t}_B + \mathbf{t}_X \tag{2.19}$$

Here, **A** is the Homogeneous Transform (HT) describing the motion of the end effector between two poses, **B** is the HT describing the reconstructed camera motion and **X** is the unknown extrinsic parameters, also represented as a HT. The matrices **A** and **B** are therefore the relative pose between two absolute poses, as defined in Section 2.1.3. If these movements instead are chosen to be relative a fixed reference frame the formulation becomes the Hand-Eye-World Calibration problem [2], which is not considered in this project.

Multiple methods exist for finding the **X** which solves Equation (2.16), either by some closed-form equation or by iterative optimization of a loss function. A summary of select methods are given in Section 2.3.4.

2.3.3 Mathematical properties

Existence, uniqueness and degeneracy

It has been shown that to be able to uniquely determine **X**, the robot hand must undergo at least two motions with non-parallel axis of rotation [18]. Additionally, Andreff *et al.* show in [4] how failure to meet these conditions will results in different indeterminate cases, depending on the nature of the performed motions. Especially of note for this report is the case of purely planar motion, for which Andreff *et al.* prove that two nonzero movements will cause the entire extrinsics to be solvable *except* the height of the sensor relative the plane of motion.

It should be noted that in the case of planar motion, there is no restriction on the last degree of freedom. This fact can cause certain strategies for finding the unknown extrinsics to diverge.

Selecting data for Hand-Eye calibration

The previous section outlines how yielding the Hand-Eye Calibration observable requires only two Hand-Eye-pose pairs. For numerical estimation purposes it is beneficial to include more data than the absolute minimum necessary. However, it is not necessarily clear which poses the robot-arm should be commanded to assume, as to optimize the numerical results.

In [19], Schmidt *et al.* present criteria for what qualifies as *good data* to be used in a Hand-Eye estimation problem. If only the orientation-part of the extrinsics is to be estimated, their criteria may be condensed to:

1. Maximize the angle between rotation axes of relative movements (influence on error in rotation, no translation recovery possible for parallel axes).
2. Maximize the rotation angle of relative movements (influence on error in rotation and translation).

These criteria are based on the calculation performed by Tsai *et al.* in [18]. Restated briefly, Tsai *et al.* prove how uncertainty on the data propagates more strongly through to the estimate of the orientation when these criteria are not fulfilled. The precise relationship between these criteria and the uncertainty of the estimate is restated in Equation (2.20). Here, $\boldsymbol{\omega}_{ab}$ and $\boldsymbol{\omega}_{bc}$ are the rotation axes of two relative poses, while $\text{Var}(\boldsymbol{\omega}_A)$ and $\text{Var}(\boldsymbol{\omega}_B)$ are the uncertainties on Hand- and Eye movements respectively.

$$\text{Var}(\boldsymbol{\omega}_X) \propto \frac{\sqrt{\text{Var}(\boldsymbol{\omega}_A)^2 + \text{Var}(\boldsymbol{\omega}_B)^2}}{\sin[\angle(\boldsymbol{\omega}_{ab}, \boldsymbol{\omega}_{bc})]} \sqrt{\frac{1}{||\boldsymbol{\omega}_{ab}||^2} + \frac{1}{||\boldsymbol{\omega}_{bc}||^2}} \tag{2.20}$$

2.3.4 Hand-Eye solvers

With the Hand-Eye Calibration problem formulated, and requirements of the input data quantified, the last step is to solve the equation. Methods for finding the matrix $\mathbf{X}$ which solves the Hand-Eye Calibration problem are hereby dubbed *Hand-Eye solvers*, and may be assembled in two groups of two: They can be closed-form solutions or iterative solutions, and they can be either simultaneous or step-wise solvers [2]. For the first group, preliminary calculations are performed on the data before a single line of mathematical calculation computes the extrinsic calibration, for instance preparing data for- and performing a linear least squares solution. As for the iterative solvers, techniques such as optimization or contraction are employed to iteratively approach the solution [2]. Regarding the second group, simultaneous solvers find both the orientation and position of the sensor at the same time while step-wise solvers solve for the orientation first and then use that estimate to compute the position.

For this project, only the orientation of the camera is of interest to estimate. This means only part of the full Hand-Eye equation needs to be solved, and the relevant term is restated in Equation (2.21) for simplicity.

$$\mathbf{R}_{\mathrm{A}}\mathbf{R}_{\mathrm{X}} = \mathbf{R}_{\mathrm{X}}\mathbf{R}_{\mathrm{B}} \tag{2.21}$$

Below, a selection of Hand-Eye solvers are presented.

Park-Martin

In their 1994 paper, Park and Martin propose a technique for solving the Hand-Eye Calibration problem [13]. Their method differs from the methods presented in the first Hand-Eye papers [1, 18] in the sense that instead of being derived from geometry, Park and Martin's solution technique is derived using group theory. The technique provides a step-wise closed form solution, but the method can also be formulated as a step-wise iterative optimization.

Consider first the following mathematical properties, for which the authors provide proofs.

Property 1: $\log\left(\mathbf{X}\mathbf{B}\mathbf{X}^{T}\right) = \mathbf{X}\log\left(\mathbf{B}\right)\mathbf{X}^{T}$

Property 2: $\mathbf{X}\log\left(\mathbf{B}\right)\mathbf{X}^{T} = \left(\mathbf{X}\mathrm{Log}\left(\mathbf{B}\right)\right)^{\wedge}$

By these two properties, it is clear the rotational Hand-Eye Equation, Equation (2.21), may be reformulated into Equation (2.22).

$$\mathbf{R}_{\mathrm{X}}\mathrm{Log}\left(\mathbf{R}_{\mathrm{B}}\right) = \mathrm{Log}\left(\mathbf{R}_{\mathrm{A}}\right) \tag{2.22}$$

Further, Park and Martin derive the closed-form solution of Equation (2.22) to be $\mathbf{R}_{\mathrm{X}} = (\mathbf{M}^{\mathrm{T}}\mathbf{M})^{-1/2}\mathbf{M}^{\mathrm{T}}$, where $\mathbf{M}$ is as shown in Equation (2.23).

$$\mathbf{M} = \sum_{i} \mathrm{Log}\left(\mathbf{R}_{\mathrm{B}_{\mathrm{i}}}\right)\mathrm{Log}\left(\mathbf{R}_{\mathrm{A}_{\mathrm{i}}}\right)^{\mathrm{T}} \tag{2.23}$$

The equality in Equation (2.22) can be understood as the following: Changes in the orientation of the camera is directly linked to how the ship changes orientation. Meaning, all rotation performed by the camera must stem from some rotation of the ship, and the two are linked numerically through the extrinsic parameters of the camera.

Andreff-Horaud-Espiau

The authors Andreff, Horaud and Espiau present in their 2001 paper an alternative formulation of the Hand-Eye Calibration problem, which recasts the problem as a linear equation from which the scale of camera movements also may be estimated.

The formulation uses the Kronecker product, defined in Equation (2.24), and matrix vectorization, defined in Equation (2.25), as well as properties of these.

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix} \tag{2.24}$$

$$\text{vec}(\mathbf{A}) = [a_{11}, \ldots, a_{m1}, a_{12}, \ldots, a_{m2}, \ldots, a_{mn}]^{\text{T}} \tag{2.25}$$

Property 3: $\text{vec}(\mathbf{AXB}) = (\mathbf{B}^{\text{T}} \otimes \mathbf{A})\text{vec}(\mathbf{X})$

Shuffling Equation (2.21) and applying vectorization and *property 3* allows for rephrasing the rotational Hand-Eye Calibration problem as Equation (2.26).

$$\begin{aligned} \mathbf{R}_{\text{A}}\mathbf{R}_{\text{X}} &= \mathbf{R}_{\text{X}}\mathbf{R}_{\text{B}} \\ \mathbf{R}_{\text{A}}\mathbf{R}_{\text{X}}\mathbf{R}_{\text{B}}^{\text{T}} &= \mathbf{R}_{\text{X}} \\ \text{vec}(\mathbf{R}_{\text{A}}\mathbf{R}_{\text{X}}\mathbf{R}_{\text{B}}^{\text{T}}) &= \text{vec}(\mathbf{R}_{\text{X}}) \\ (\mathbf{R}_{\text{B}} \otimes \mathbf{R}_{\text{A}})\text{vec}(\mathbf{R}_{\text{X}}) &= \text{vec}(\mathbf{R}_{\text{X}}) \\ (\mathbf{I}_{9\times 9} - \mathbf{R}_{\text{B}} \otimes \mathbf{R}_{\text{A}})\text{vec}(\mathbf{R}_{\text{X}}) &= \mathbf{0}_{9\times 1} \end{aligned} \tag{2.26}$$

The authors show how the same method can be applied on the translational part of Equation (2.16), while including a factor for unknown scale of the camera movement to form Equation (2.27). This does, however, require flipping the Hand-Eye problem so that the **A** matrices represent camera movements, and **B** matrices are the arm movements, meaning the estimated extrinsic is inverted. Also, Andreff *et al.*'s definition of the Kronecker product is the transposed of the definition used in this report. Their derivations lead to the linear formulation in Equation (2.27), which is dubbed *AHE simultaneous* in this report. *AHE* from the three authors and *simultaneous* as it solves for both orientation and position at the same time.

$$\begin{bmatrix} \mathbf{I}_{9\times 9} - \mathbf{R}_{\text{B}} \otimes \mathbf{R}_{\text{A}} & \mathbf{0}_{9\times 3} & \mathbf{0}_{9\times 1} \\ \mathbf{t}_{\text{B}}^{\text{T}} \otimes \mathbf{I}_{3\times 3} & \mathbf{I}_{3\times 3} - \mathbf{R}_{\text{A}} & -\mathbf{t}_{\text{A}} \end{bmatrix} \begin{bmatrix} \text{vec}(\mathbf{R}_{\text{X}}) \\ \mathbf{t}_{\text{X}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{9\times 1} \\ \mathbf{0}_{3\times 1} \end{bmatrix} \tag{2.27}$$

Iterative optimization

The general problem of iterative optimization is often formulated in the framework of a nonlinear least squares optimization problem. One popular all-round solver is found in the open-source Python library SciPy [20], where the optimization problem is defined as seen in Equation (2.28). Here, f_i is named the *ith residual*, x are the *optimization variables*, $F(x)$ is the cost function and $\rho(\cdot)$ is some weighting function.

$$\min_x F(x) = \frac{1}{2}\sum_i \rho(f_i(x)^2) \tag{2.28}$$

When using nonlinear least squares solvers to minimize error related to the Hand-Eye Calibration problem, the sum over i in Equation (2.28) is the sum over

all Hand-Eye movement pairs $(\mathbf{A}_i, \mathbf{B}_i)$. A collection of different pertinant choices for the residual function when attempting to solve the Hand-Eye calibration problem is presented below.

Park and Martin's approach [13] to solving the Hand-Eye calibration problem is itself a least squares solution, so one can expect a cost-function made of the terms in their method to do well. The residual can be seen in Equation (2.29), and is directly reflecting Equation (2.22).

$$f_i(x) = \mathbf{R}_x \text{Log}(\mathbf{B}_i) - \text{Log}(\mathbf{A}_i) \tag{2.29}$$

Similarily, the AHE closed-form solution involves finding the null-space of a matrix, see Equation (2.26). This can be compared to minimizing a strictly positive function, and may then be formulated as the residual in a cost-function seen in Equation (2.30).

$$f_i(x) = (\mathbf{I}_{9\times 9} - \mathbf{R}_\text{B} \otimes \mathbf{R}_\text{A})\text{vec}(\mathbf{R}_\text{X}) \tag{2.30}$$

Having defined a metric over SO(3) in Section 2.1.4, an alternative residual for solving the Hand-Eye Calibration problem can simply be the difference between the terms in the Hand-Eye equation, Equation (2.18). This is seen in Equation (2.31).

$$f_i(x) = \text{Log}\left(\left(\mathbf{R}_{\text{A,i}}\mathbf{R}_\text{X}\right)^\text{T}\left(\mathbf{R}_\text{X}\mathbf{R}_{\text{B,i}}\right)\right) \tag{2.31}$$

Note how all the presented residuals return vectors while the framework in SciPy requires each residual to be a scalar function. One can solve this by returning the norm of f_i, or simply returning the vector residual as three seperate residuals.

Chapter 3

Method

3.1 Hand-Eye formulation for ships

The goal in this project is to estimate the orientation of ship-mounted cameras. Modern ships have advanced sensor-suites fusing GNSS measurements with inertial- and attitude measurements, meaning the ship's pose is available frequently and with high accuracy. Further, as explained in Section 2.2, there exists several algorithms which estimate camera egomotion in surroundings without a calibration plate, with the downside of the reconstructed motion having unknown scale.

With these two facts in place, it is possible to recognize that modern ships with rigidly mounted cameras have all the data necessary to formulate a fitting Hand-Eye Calibration problem, with the solution being the unknown extrinsics. The ship's attitude and position sensor gives data analogous of the "hand" movements in Section 2.3 and the aformentioned egomotion algorithms give the scaleless "eye" movements. This formulation is presented in Equation (3.1). Here, the λ represents the unknown scale factor of the reconstructed motion required to map translations into the same scale as the ship-movements.

$$\mathbf{AX} = \mathbf{XB}(\lambda) \tag{3.1}$$

Further, Equation (3.1) may be expanded into Equation (3.2), seperating estimation of the orientation and position of the camera, $\mathbf{R}_X$ and $\mathbf{t}_X$.

$$\begin{gathered}
\mathbf{AX} = \mathbf{XB}(\lambda) \\
\begin{bmatrix} \mathbf{R}_A & \mathbf{t}_A \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix}\begin{bmatrix} \mathbf{R}_X & \mathbf{t}_X \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_X & \mathbf{t}_X \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix}\begin{bmatrix} \mathbf{R}_B & \mathbf{t}_B\lambda \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} \\
\sim \\
\mathbf{R}_A\mathbf{R}_X = \mathbf{R}_X\mathbf{R}_B \\
\mathbf{R}_A\mathbf{t}_X + \mathbf{t}_A = \mathbf{R}_X\mathbf{t}_B\lambda + \mathbf{t}_X
\end{gathered} \tag{3.2}$$

A conceptual illustration of the setup is shown in Figure 3.1.

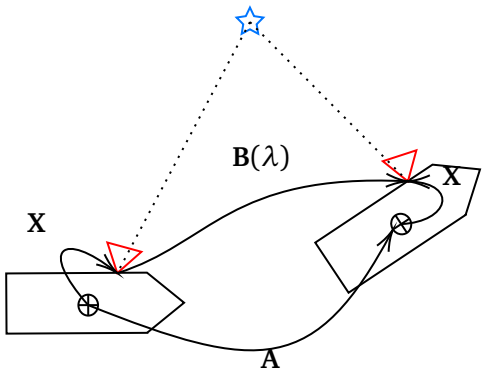


Figure 3.1: Conceptual illustration of the Hand-Eye Calibration problem for the case of a ship-mounted camera. Note the variable λ representing the unknown scale factor, and how a feature detection algorithm must be employed instead of a calibration plate, with a detected landmark represented as a star.

3.1.1 Mathematical derivation

As explained in Section 2.3.2, the matrices **A** and **B** in Equation (3.1) are *relative poses*. This subsection shows more concretely how these are computed for the case of a ship-mounted camera, as well as the mathematical derivation of the Hand-Eye calibration problem for the case of ship-mounted cameras.

For a ship with with high-accuracy navigational units, $\mathbf{H}_{\text{nb}}(t)$ is denoted as the pose of the body-frame relative North-East-Down (NED) at time t and $\mathbf{H}_{\text{mi}}(t)$ as the pose of sensor i relative the unknown *mediary* frame output by an ego-motion algorithm for the picture taken at time t. Then given two timestamps, (t_p, t_q) such that $t_p < t_q$, these relative movements are calculated as follows in Equation (3.3). It is required that t_p be a strictly earlier point in time than t_q, since the case of $t_p = t_q$ causes the relative motion to be the identity, which contains no useful information.

$$\begin{aligned} \mathbf{H}_{\text{b,pq}} &= \left(\mathbf{H}_{\text{nb}}(t_p)\right)^{-1} \mathbf{H}_{\text{nb}}(t_q) := \mathbf{A}_{pq} \\ \mathbf{H}_{\text{i,pq}} &= \left(\mathbf{H}_{\text{mi}}(t_p)\right)^{-1} \mathbf{H}_{\text{mi}}(t_q) := \mathbf{B}_{pq} \end{aligned} \tag{3.3}$$

Denoting $\mathbf{H}_{\text{bi}}$ as the extrinsic calibration of camera i, Equation (3.4) shows how the Hand-Eye calibration problem in fact may be derived given the available measurements.

$$
\begin{aligned}
\mathbf{I} &= \mathbf{I} \\
\mathbf{G}\mathbf{G}^{-1} &= \mathbf{H}\mathbf{H}^{-1}, && ,\forall\ \mathbf{G},\mathbf{H} \in \mathrm{SE}(3) \\
\mathbf{H}_{\mathrm{nb}}(t_q)\mathbf{H}_{\mathrm{bn}}(t_q) &= \mathbf{H}_{\mathrm{nb}}(t_p)\mathbf{H}_{\mathrm{bn}}(t_p) && ,\forall\ t_p \neq t_q \\
\mathbf{H}_{\mathrm{nb}}(t_q)\mathbf{H}_{\mathrm{bi}}\mathbf{H}_{\mathrm{in}}(t_q) &= \mathbf{H}_{\mathrm{nb}}(t_p)\mathbf{H}_{\mathrm{bi}}\mathbf{H}_{\mathrm{in}}(t_p) && ,\mathbf{H}_{\mathrm{bn}}(t_p)\cdot \\
\mathbf{H}_{\mathrm{bn}}(t_p)\mathbf{H}_{\mathrm{nb}}(t_q)\mathbf{H}_{\mathrm{bi}}\mathbf{H}_{\mathrm{in}}(t_q) &= \mathbf{H}_{\mathrm{bi}}\mathbf{H}_{\mathrm{in}}(t_p) && ,\cdot\ \mathbf{H}_{\mathrm{ni}}(t_q) \\
\mathbf{H}_{\mathrm{bn}}(t_p)\mathbf{H}_{\mathrm{nb}}(t_q)\mathbf{H}_{\mathrm{bi}} &= \mathbf{H}_{\mathrm{bi}}\mathbf{H}_{\mathrm{in}}(t_p)\mathbf{H}_{\mathrm{ni}}(t_q) \\
\mathbf{H}_{\mathrm{nb}}^{-1}(t_p)\mathbf{H}_{\mathrm{nb}}(t_q)\mathbf{H}_{\mathrm{bi}} &= \mathbf{H}_{\mathrm{bi}}\mathbf{H}_{\mathrm{ni}}^{-1}(t_p)\mathbf{H}_{\mathrm{ni}}(t_q) \\
&= \\
\mathbf{A}_{pq}\mathbf{H}_{\mathrm{bi}} &= \mathbf{H}_{\mathrm{bi}}\mathbf{B}_{pq}
\end{aligned}
\tag{3.4}
$$

It may be observed how the Homogeneous Transform $\mathbf{H}_{\mathrm{ni}}(t)$ is not a part of the available measurements, since it is unknown how the arbitrarily constructed egomotion reconstruction-frame alignes with NED. This can luckily be ignored when only the orientation is to be estimated, as shown in Equation (3.5).

$$
\mathbf{R}_{\mathrm{in}}(t_p)\mathbf{R}_{\mathrm{ni}}(t_q) = \big(\mathbf{R}_{\mathrm{im}}(t_p)\mathbf{R}_{\mathrm{mn}}\big)\big(\mathbf{R}_{\mathrm{nm}}\mathbf{R}_{\mathrm{mi}}(t_q)\big) = \mathbf{R}_{\mathrm{im}}(t_p)\mathbf{R}_{\mathrm{mi}}(t_q) \tag{3.5}
$$

This shows how only the relative motion irrespective of coordinate frame between two points in time is needed. Note also from Equation (3.4) how choosing $t_p = t_q$ leads to the equation $\mathbf{H}_{\mathrm{bi}} = \mathbf{H}_{\mathrm{bi}}$, which obviously contains no value.

With the Hand-Eye Calibration problem formulated for the case of ship-mounted camera, the next step is to choose a Hand-Eye solver and generate estimates.

3.1.2 Some considerations when using ship-data in Hand-Eye

Formulating the Hand-Eye Calibration problem for the case of ship-mounted cameras brings some specific challenges which need to be adressed. Firstly, as noted in Section 2.3.3, the full estimation problem is degenerate when "hand" motions are mostly planar. This is the case for ship-data and especially when the ship is near land, where waves are less dominant on the ship movement. This poses a challenge, as it is the times at which the ship is near land that most egomotion algorithms have the easiest time tracking features. In this project, this is taken care of by only estimating the orientation of the camera, which is fully observable even in the case of planar data. Near-planar data will also cause numerical instability but work such as [21] suggest the possibility of detecting and avoiding the degenerate directions in the parameter space.

The second main concern when employing this pipeline is the missing scale in the egomotion estimation. Much research is being done on the topic of scale estimation, often employing deep learning to achieve this. The Hand-Eye solver of Andreff *et al.* presented in Section 2.3.4 also estimates the scale, but is more sensitive to the planarity of the data. Again, this issue may also be solved by simply ignoring this fact and solely estimating the orientation, as done in this project.

3.2 Algorithm pipeline

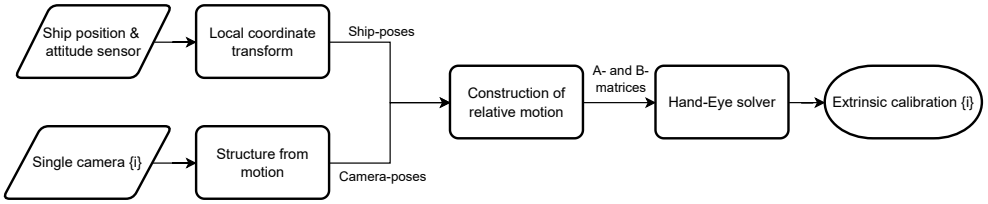


Figure 3.2: Flowchart of the Automatic Extrinsics Calibration-algorithm pipeline

With the previous sections as motivation, an algorithm is presented for finding the extrinsic calibration parameters based on nothing but acquired ship-data. The algorithm is implemented in Python. What follows is an explanation of the parts of the algorithm pipeline, as well as a summary of some questions which challenge the validity of the given output from the algorithm. Some of these questions and design choices are addressed, while some questions are left unanswered as potential further work.

One notable simplification is made: While the pipeline in theory supports estimation of the full camera extrinsics, both orientation and position, only the estimation of orientation is evaluated in this project. The reason for this is the fact that the pipeline developed does not yet have a method to compensate for the planarity and missing scale of the data, leading to unsatisfactory results concerning the estimate of camera position.

3.2.1 Local coordinates

The positional data from the ship's navigational units are often given in a coordinate system preferred by the GNSS-system, like WGS84 or some geodetic coordinate. The Hand-Eye formulation, however, requires translations to be given in inertial Euclidian frames. A first step in the algorithm is to process the positional data to construct a local tangent plane from which to define the NED-coordinate system.

For the purposes of this project, it was decided that for the short timespans analyzed (1-2 minutes of data at the time, see Section 4.1) the local flat-earth-approximation is sufficient. The local tangent plane is defined with its origin at the first datapoint. This is done for simplicity, even though an objectively less erroneous approach is to use the middlemost datapoint as origin, thereby halving the error. Still, considering the short timespan of the data, it was decided that this difference is negligible. One can, however, imagine that if longer timespans of data were used for estimation, then it would be advantageous to make a batchwise approach where the local tangent plane is redefined for each batch.

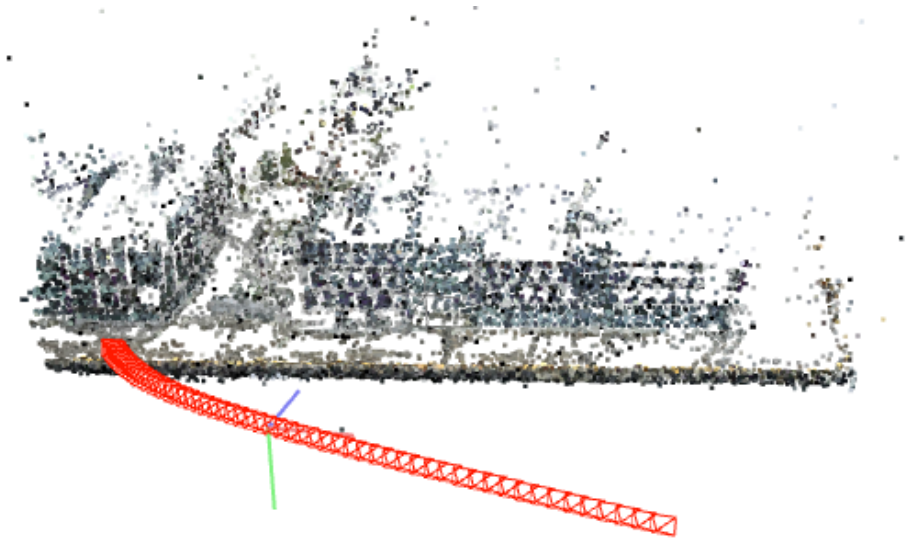


Figure 3.3: Example output reconstruction from the COLMAP SfM software. Shown are camera poses in red and successfully tracked image points.

3.2.2 Structure-from-Motion

To reconstruct the motion of the camera based solely on images, an open-source Structure from Motion (SfM) library can be used. Some examples include COLMAP [22] and OpenSfM [9]. The implementation in this project uses the former, employing OpenSfM to verify the reconstructions. Both libraries mentioned perform standard SfM using feature-extraction and tracking, refining estimates through bundle-adjustment.

An example of the output reconstruction from COLMAP is seen in Figure 3.3. The points plotted are those which the algorithm successfully tracks across more than 3 images.

As mentioned in Section 2.2, there are multiple different ways to perform ego-motion reconstruction. The choice to do this with SfM in this project was mostly motivated by the simplicity, and one could just as easily implement the pipeline with either VSLAM or VO algorithms to generate the camera poses. Still, Structure from Motion methods generally provide higher accuracy reconstruction due to the lack of runtime-constraints as in VSLAM or VO methods, which are often required to run in real-time. Nevertheless, if the pipeline presented in this report is to be extended to work in real-time or near real-time operations, it might be a good choice to switch to a VSLAM or an online SfM algorithm. It is also possible to skip the reconstruction frameworks entirely, and simply pick image-pairs to do pair-wise relative pose estimation based on purely epipolar geometry. This is however expected to give poorer but faster pose-estimates, since algorithms like SfM and VSLAM track the scene and use the assumption of world-points being constant to

build a consistent pose-estimate.

Another weakness of SfM as opposed to VSLAM is the lack of loop-closure correction. One expects the SfM reconstruction to drift more over time than a VSLAM reconstruction detecting loop-closures. Again, the short timespan of the data used to perform analyses in this project is expected to minimize the effect of drift.

3.2.3 Construction of relative motion

The Hand-Eye problem formulation requires the hand- and eye motions to be pairwise relative poses. That is, the **A** and **B**-matrices fed into the Hand-Eye solver must be "paired" in the sense that a pair $(\mathbf{A}, \mathbf{B})$ must correspond to the relative pose of the hand and eye between the same two points in time.

When performing Hand-Eye Calibration on the setup with a camera mounted on a robot arm, the movements of the arm may be chosen arbitrarily, allowing for construction of a finite set of optimal movements. For real-time operations, choosing a finite set of optimal movements is not always possible, for three reasons: Firstly, the motion of the system may already be predefined by the user or the control-system, so the calibration algorithm must work with the data it is supplied. Secondly, if a calibration method is to be employed for datasets with a large time horizon, the amount of data mat, over time, be too large to handle efficiently in software. Thirdly, if the data is corrupted by noise and outliers, it may be beneficial to pick the set of datapoints which balances most excitation and least noise.

A point of interest therefore lies in developing a strategy for choosing which pairs of datapoints are to be combined into a single relative pose, which the Hand-Eye Calibration is then based on. If one considers the movements $\mathbf{H}_{pq}$ and $\mathbf{H}_{qp}$ to be equal in the eyes of the Hand-Eye Calibration, as the two movements have the same rotation axis up to a sign difference, then for n datapoints there will be $\binom{n}{2} = \frac{n!}{2\cdot(n-2)!} = \frac{1}{2}n(n-1) \propto n^2$ unique data pairs to choose from. This is problematic for the runtime of the algorithm.

For these reasons, one is motivated to have an intelligent strategy for choosing the datapoints which make up a single relative pose, or alternatively employing a receding horizon where old datapoints are in turn removed from the dataset. At the same time, as long as old datapairs are consistent with new observations, one does not wish to disregard the old datapairs if these are more strongly excited than the new.

The simplest such strategy is to make all poses be relative the first point in time. This is the only method employed in this project. Another choice is to pick the datapairs which fulfill the criteria in Section 2.3.3 best.

3.2.4 Hand-Eye solvers

As briefly discussed in Section 2.3.4, there exist multiple methods of solving the Hand-Eye Calibration problem, both analytically and through iterative optimization. These solvers work directly on the input $(\mathbf{A}, \mathbf{B})$ motion pairs, without fur-

ther knowledge of overarching system or structure therein. In theory, any of these methods should work about equally well for the purpose of estimating the extrinsic parameters using ship-data. In practice, however, due to the degeneracy of the data it is expected that closed-form solvers will result in divergent or nonsensical estimates. Loosely explained, even though the data generated by ships aren't exactly planar, just as dividing by a number which is very small but not *exactly* zero, the closeness to degeneracy should make sensitive and bad results expected.

3.3 A qualitative measure of Hand-Eye excitation

In Section 2.3.3, the requirements for what is considered good data for performing Hand-Eye Calibration is listed, based on calculations performed by Tsai *et al.* in [18]. These guidelines are quantitative, but the strategy of movement-selection suggested by the authors is not applicable in this project since the movements cannot be decided ahead of time by the calibration system. A data-selection criteria which works on any given batch of data, and which can be employed in real-time to select the best datapairs given a stream of high amounts of noisy data would be beneficial. Below, a qualitative method for evaluating excitation in data for Hand-Eye Calibration is presented, with the motivation of building a rigorous optimal data-picking strategy in the future.

How selection of data affects the propagation of uncertainty to the estimate of camera orientation was presented in Equation (2.20). A first step towards understanding data-selection for Hand-Eye calibration is to simply see what characteristics data must possess to minimize these terms present in Equation (2.20). To this end, the data is evaluated two ways. The *type 1* excitation is defined as the size $||\boldsymbol{\omega}_{\mathrm{b}i}||$, with $\boldsymbol{\omega}_{\mathrm{b}i}$ being the orientation axis of ship-pose i. *Type 2* excitation is defined as $|\sin\left[\angle(\boldsymbol{\omega}_{\mathrm{b},i}, \boldsymbol{\omega}_{\mathrm{b},j})\right]|$, for any two body-orientations i and j. Due to the fact that $\mathbf{R}(\theta, -\mathbf{a}) = \mathbf{R}(-\theta, \mathbf{a})$, the choice of angles and axes which yield the smallest value for each of these metrics are chosen. Type 1 excitation is then maximized if each relative pose has orientation-angle as close to 180° as possible, and type 2 excitation is maximized if the angle between the orientation-axes of two relative poses are as close to 90° as possible. The two metrics are evaluated graphically in the next section.

If the system which is to be calibrated allows for arbitrarily choosing movements, the suggested strategy by Tsai *et al.* is to evenly space N poses around a regular N-sided polygon [18]. In the case of ship-mounted cameras, the biggest concern to the integrity of the presented algorithm pipeline is the planarity of the data. If possible, it is then advised to have the ship perform rapid turns and braking maneuvers as to maximize roll and pitch-motion.

Chapter 4

Results

4.1 Experimental setup

It has been shown that the choice of parametrization of SO(3) can affect both the accuracy and runtime of iterative optimization problems where the optimization variable is an orientation [2]. For simplicity in this project, only Euler-angles were as optimization variables in the iterative Hand-Eye solvers.

4.1.1 The datasets

For this paper, three types of datasets are used to generate the results presented. These are: *synthetic uniform*, *synthetic planar* and *real-world* data. A short presentation of each of them follows.

Synthetic uniform

As explained in Section 2.3.3, the extrinsic parameters are only completely recoverable from the Hand-Eye problem when at least two non-zero movements with non-parallel rotation axis are present. Generally, more diverse movements will lead to better numerical properties for the closed-form solutions. For this reason, it is logical to compare the methods against each other using a dataset containing uniformly random data, which therefore reflects the amount of excitation when there is no underlying structure in the data.

The *synthetic uniform* dataset consists of uniformly random poses sampled from a predefined sample space. Randomly drawing positions of the ship is as simple as drawing 3 elements from a uniform distribution limited to lie between an upper and lower treshold. Drawing uniformly over SO(3), however, is not as trivial. Drawing an axis and an angle uniformly does not uniformly cover the space of all rotations, an neither does drawing Euler-angles uniformly [23]. In this project, the method used in [24] is used for uniformly drawing orientations from SO(3). This is the same method used as MATLAB. The poses generated are used as ship-poses, a set of arbitrary ground truth extrinsic parameters are chosen,

and the camera poses are calculated therefrom. The synthetic uniform dataset is therefore also generated to be free of noise.

Synthetic planar

The end-goal of this project is to analyze the effect of using ship data for performing Hand-Eye Calibration. As noted earlier, data from ships are of special interest to the problem formulation, as planar movement theoretically is a degenerate case of the Hand-Eye Calibration problem. To eliminate factors like noise on measurements of the ships pose and noise on the reconstructed camera-poses, the *synthetic planar* dataset was constructed to resemble an imagined ship-dataset.

The dataset is generated as a simple motion model of a ship with constant velocity and a random walk on the roll, pitch and yaw-angles. The random walk on yaw additionally has a stabilizing term, meaning the ship over time will tend to return to the origin.

An example output of both synthetic datasets may be seen in Figure 4.1

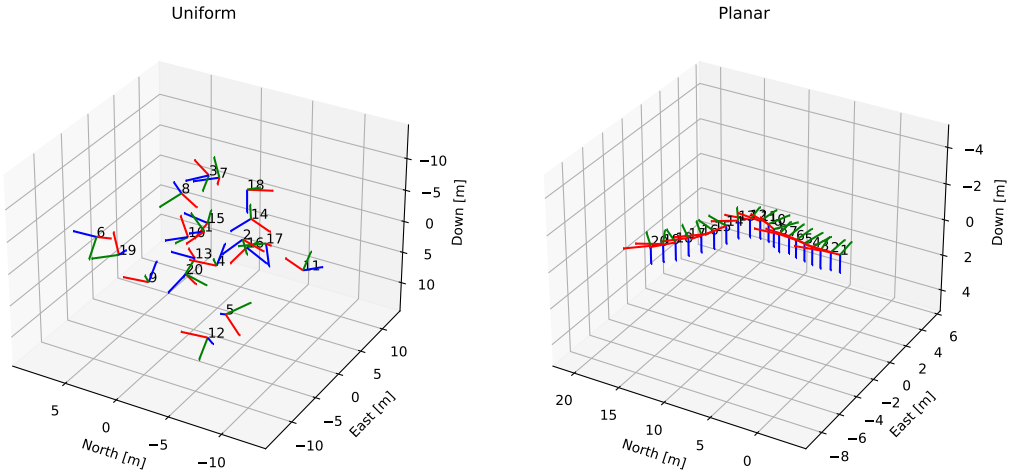


Figure 4.1: The 15 first generated poses in the synthetic uniform and synthetic planar datasets

Real-world data

The real-world data used for testing in this project was made available through the research center SFI Autoship, a collaboration between Norwegian University of Science and Technology (NTNU) and various commercial actors, including Kongsberg Maritime (KM) and SINTEF among others. The data was supplied by KM. Of

the data available, the datasets selected for testing in this project are from two different Kongsberg Maritime research projects, hereby dubbed *weakly excited* and *strongly excited*. The weakly excited KM dataset is from a passenger cruise ship mounted with stereo rigs for research purposes. The large size and nature of it being a passenger ship causes slow turning and little influence from waves on the ship orientation, which means the dataset contains minimal of the optimal Hand-Eye excitation outlined in Section 2.3.3. This is why the dataset is dubbed weakly excited. The strongly excited dataset is collected from a smaller research-vessel, and therefore contains sharper turning of the ship and more movement caused by waves and wakes.

In both of the experimental setups, subsets of the datasets for which the ship is in motion near land was chosen, to enable detection and tracking of features. It is expected for feature tracking to be considerably more difficult in the case where the ship is on open sea. Both ships from which the data was collected had data from their respective stereo-rigs, but for simplicity in this project only data from a single camera is considered. Further, the previously established extrinsic parameters of each camera were available to compare against the estimates produced by the algorithm. One important point to note is that the accuracy of these estimates are not known, making it difficult to discern whether the new estimate or old extrinsic is closer to the actual extrinsics.

Another important point to note is that the ship pose for both these datasets were estimated by Kongsberg Maritime *Seapath* units. These units are known for their high accuracy and good time-synchronization [25]. So although the algorithm pipeline may be implemented on any physical setup, the high accuracy of the Seapath used to generate results in this project means results may vary.

Figures 4.2 and 4.3 show examples of images in each of the datasets. Note by looking at the horizon how Figure 4.3 show more motion than that in Figure 4.2. Further, the reconstructed camera motion for each dataset is shown in Figure 4.4. The reconstructed poses are plotted from above, and for clarity only every fourth pose is plotted. Note the large jumps in the reconstructed camera-motion based on the KM strongly excited dataset's images.

4.2 Metrics

The following explains the metrics used to generate the results presented in Section 4.3.

4.2.1 Comparing reconstructions

For evaluating the validity of egomotion reconstructions, a metric like the one presented in section 3.3 of [26] can be used. A challenge is that this method requires the knowledge of the ground-truth camera motion, which for this project is unknown since the extrinsics are to be estimated. Further, the kind of method presented in [26] does not take into account that the reconstruction frame does

(a) Taken at 07:30:00 **(b)** Taken at 07:30:17

Figure 4.2: Example images from the KM weakly excited dataset

(a) Taken at 08:14:20 **(b)** Taken at 08:14:37

Figure 4.3: Example images from the KM strongly excited dataset

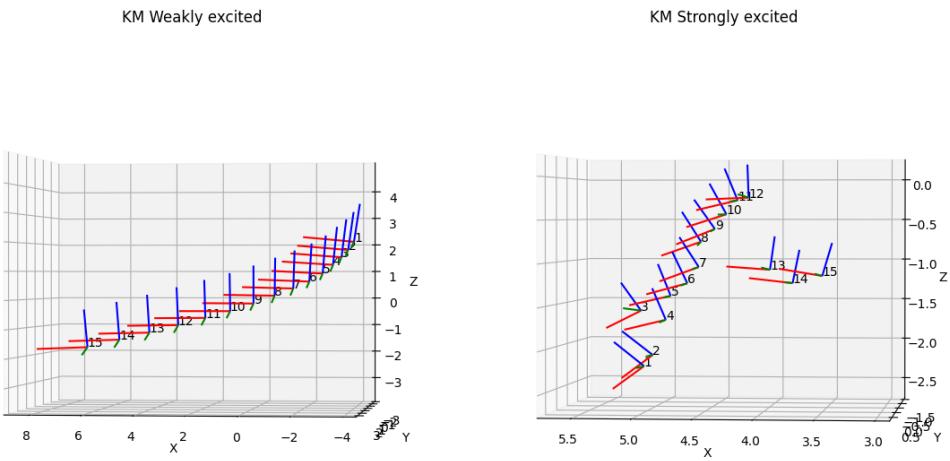


Figure 4.4: Reconstructed camera motions based on images in the KM weakly- and strongly excited datasets. The camera-poses are viewed from above, and for clarity only every fourth pose is plotted. The viewed reconstruction is performed by COLMAP.

not necessarily need to align with the reference frame of the ground truth trajectory, and therefore punishes a trajectory which in theory could be completely consistent up to a coordinate transform.

An alternative metric for evaluating correctness of reconstructions is used in this project. Firstly, the reconstructions are not compared to the unknown ground truth, but rather to each other by considering the relative pose between the reconstructions at each point in time. This is calculated as in Equation (4.1).

$$\mathbf{H}_{\text{err}}(t_i) = (\mathbf{H}_{\text{COLMAP}}(t_i))^{-1}\mathbf{H}_{\text{OpenSfM}}(t_i) \tag{4.1}$$

Equation (4.1) will generally be some non-identity transformation. To measure to which degree the reconstructions drift away from each other, this project compares the value of Equation (4.1) for different timepoints against each other. Remembering that only the orientation is of interest in this project, comparing the reconstruction metrics against each other is done by analyzing the SO(3)-metric between the ith timestamp and the first. To put it simply, this metric is to be understood as the measure of drift between the reconstructions.

For comparison, a similar analysis is done for the transformation connecting each reconstruction to the ship-orientation. The ship-poses and camera poses are related through the assumed constant extrinsic calibration, and therefore the reconstruction metric between these should also ideally be non-drifting.

4.2.2 Error between orientations

For comparing the estimated orientation of the camera versus the old assumed ground truth, the metric over SO(3) as presented in Section 2.1.4 is used. Recall that this is the length of the shortest rotation connecting the orientations, and that this metric generally does not have anything to do with Euler-angles even though the ground truth extrinsics are given in Euler-angles. For simpler interpretation, the angular difference is scaled to degrees.

For comparing estimated camera-orientation versus the old extrinsics, the metric over SO(3) as presented in Section 2.1.4 are used. For simpler reading of results, the angular difference is scaled to degrees. To be explicit, the metric is shown in Equation (4.2).

$$\text{err} = \frac{180}{\pi}||\text{Log}(\mathbf{R}_{\text{GT}}^{\text{T}}\mathbf{R}_{\text{est}})||_2 \tag{4.2}$$

4.2.3 Error in the Hand-Eye equation

Since the estimation of the extrinsic parameters are based on finding a solution to the equation $\mathbf{AX} = \mathbf{XB}$, a reasonable metric for comparing estimates is to insert them into the equation and calculate the error between each side of the equation. Since this project focuses on the orientation of cameras, only the rotational components need to be analyzed with the SO(3)-metric presented earlier. The error is averaged over all $(\mathbf{A}, \mathbf{B})$-pairs in the dataset. For consistency, the error is also

scaled to degrees. The physical interpretation is lost when averaging angles in this way, but the metric is still a reflection of the error between the terms. The precise definition of the metric is seen in Equation (4.3).

$$\text{err}_{\text{HE}} = \frac{180}{\pi}\frac{1}{N}\sum_{i=1}^{N}||\text{Log}((\mathbf{A}_i\mathbf{X})^{\text{T}}(\mathbf{X}\mathbf{B}_i))||_2 \tag{4.3}$$

4.3 Figures

4.4 Analysis of the datasets

To enable a discussion of the performance of the algorithm pipeline, the validity and properties of the input-data is analyzed first. In Figure 4.5, the camera reconstructions are compared as outlined in Section 4.2.1 for each point in time of the KM weakly excited dataset. Drift, measured in the way defined in this project, is highest for the last couple datapoints, at which point it is around 25° for the OpenSfM-reconstruction and about 15° for the COLMAP-reconstruction. Note in Figure 4.5 how the two sets of metrics for which the COLMAP-reconstruction is used for comparison contains the least amount of drift. For this reason, the COLMAP-reconstruction was chosen to be used in further analyses.

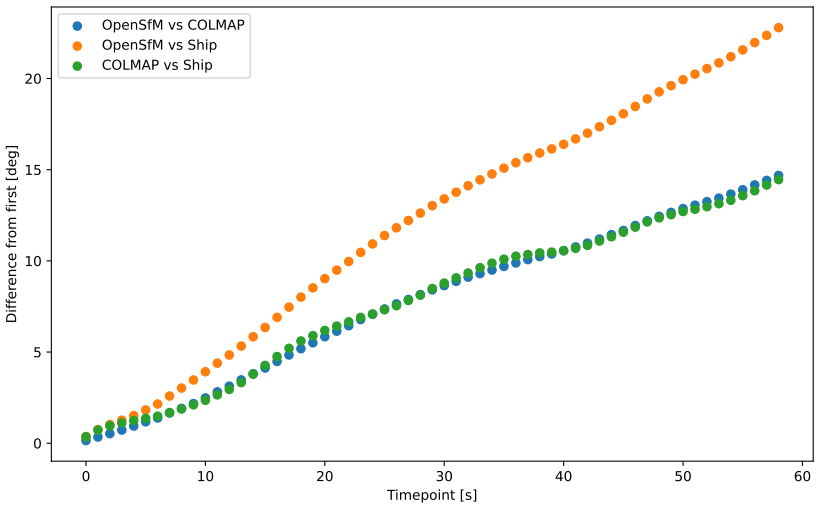


Figure 4.5: Change over time in the transformation connecting the listed sets of poses in the KM weakly excited dataset

The qualitative method of evaluating the excitation of data presented in Section 3.3 was performed on the four datasets, with results seen in Figures 4.6 to 4.9. The leftmost figures show histograms of the lengths of the rotation axes for all

poses present in the dataset. The axes are taken from the relative ship-pose at each timestamp, and choosing to use the camera-poses leads to near identical results. The rightmost figures are cross-plots of the sine between subsequent rotation axes for each datapoint, meaning the value at row i and column j depicts $\left|\sin\left[\angle(\boldsymbol{\omega}_{\mathrm{b}i}, \boldsymbol{\omega}_{\mathrm{b}j})\right]\right|$. Again, the choice to use ship-poses for this analysis is arbitrary. It should noted that the rightmost figures are symmetric about the diagonal, which itself contains only zeros.

Figure 4.6 shows that the synthetic uniform dataset contains a high amount of both types of excitation defined in this project. The type 1 excitation is mostly concentrated around 180°. The type 2 excitation of the synthetic uniform dataset is in contrast to that of the synthetic planar seen in Figure 4.7, which contains nearly none due to the planar motion yielding all orientation axes nearly parallel. The obtained type 2 excitation of the synthetic uniform dataset should be expected given how much more likely it is to randomly generate two non-parallel vectors than it is to generate two parallel vectors.

Seen in Figures 4.8 and 4.9, both of the real-world datasets' qualitative excitation measures contain similar amounts of type 2 excitation. The structure is also similar, with high amounts of excitation for the early timepoints, and decreasing excitation over time. The weakly excited dataset has seemingly smoother transitions in the type 2 excitation between each timepoint than the strongly excited dataset. The figures also suggest that in both real-world datasets, the angle between orientation axes are 90° at one or several points in time. The type 1 excitation of the weakly excited dataset is mostly concentrated around the minimum of 0°, while the strongly excited dataset has a large spike at 80°.

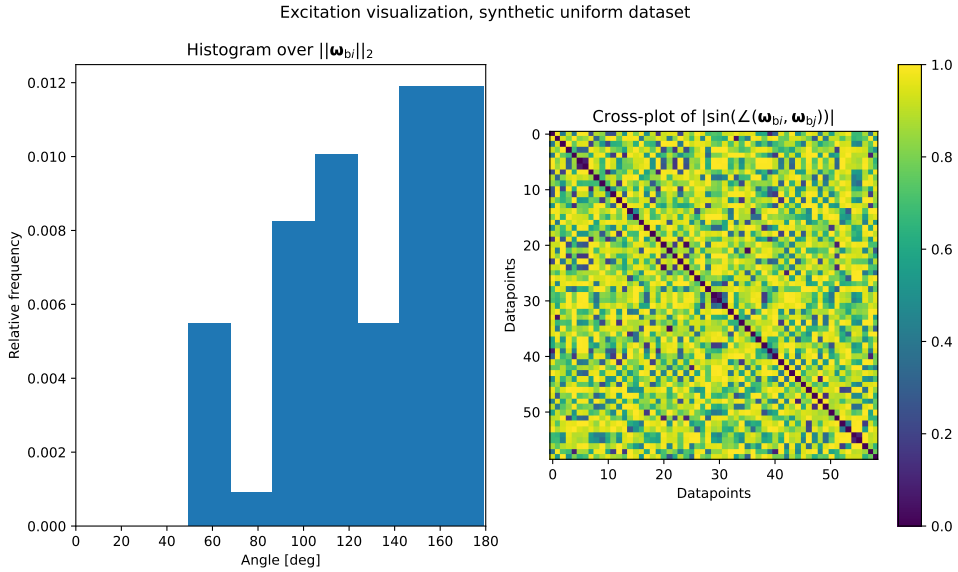


Figure 4.6: A test of the qualitative measure of Hand-Eye excitation present in the synthetic uniform dataset. The leftmost figure illustrates the type 1 excitation, the distribution of rotation magnitudes of the relative poses. The rightmost figure illustrates the type 2 excitation, the angle between each pair of relative poses.

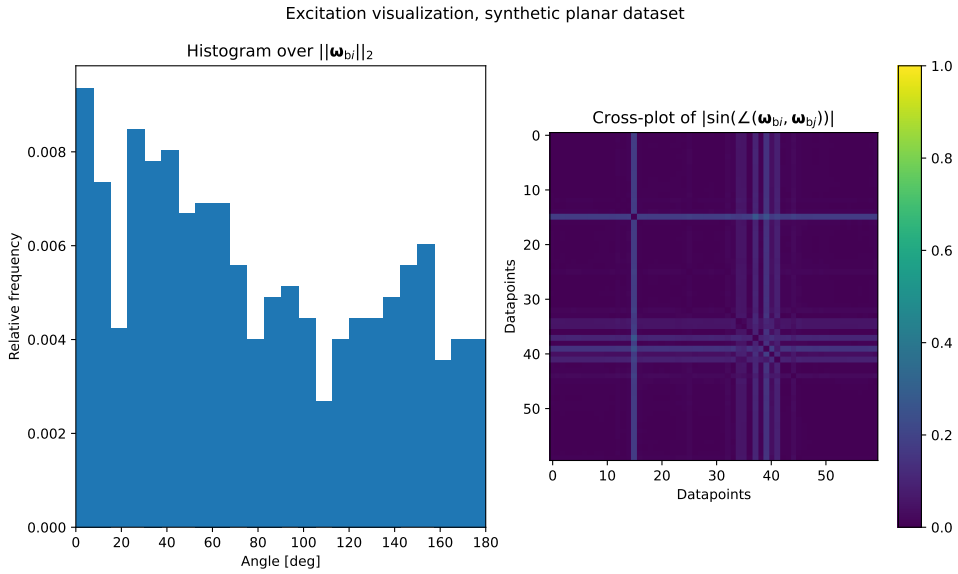


Figure 4.7: A test of the qualitative measure of Hand-Eye excitation present in the synthetic planar dataset. The leftmost figure illustrates the type 1 excitation, the distribution of rotation magnitudes of the relative poses. The rightmost figure illustrates the type 2 excitation, the angle between each pair of relative poses.

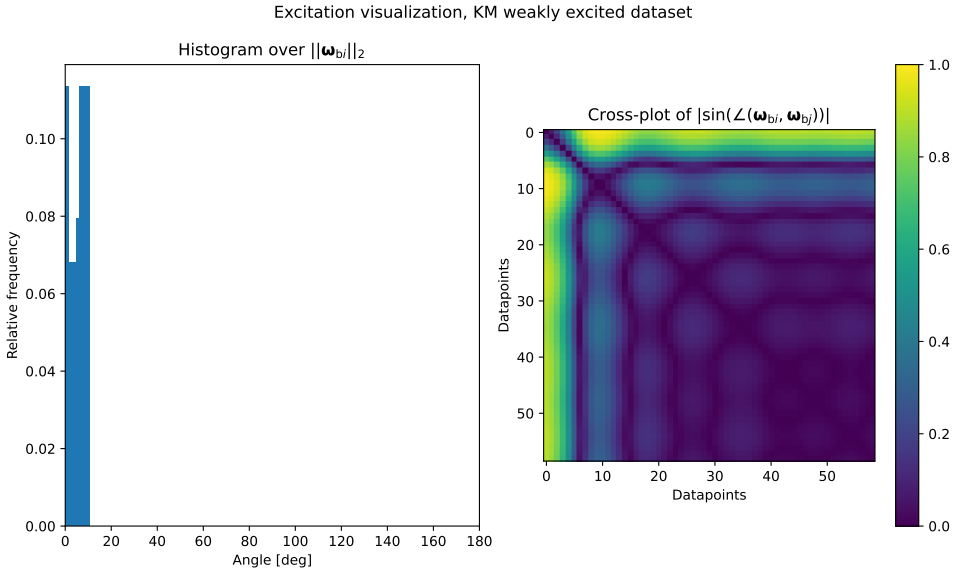


Figure 4.8: A test of the qualitative measure of Hand-Eye excitation present in the KM weakly excited dataset. The leftmost figure illustrates the type 1 excitation, the distribution of rotation magnitudes of the relative poses. The rightmost figure illustrates the type 2 excitation, the angle between each pair of relative poses.

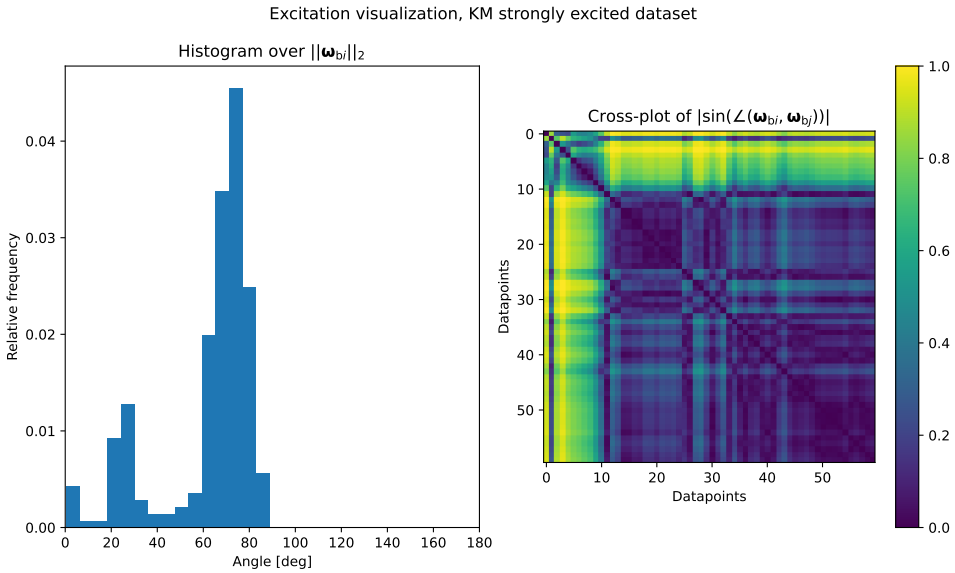


Figure 4.9: A test of the qualitative measure of Hand-Eye excitation present in the KM strongly excited dataset. The leftmost figure illustrates the type 1 excitation, the distribution of rotation magnitudes of the relative poses. The rightmost figure illustrates the type 2 excitation, the angle between each pair of relative poses.

4.5 Performance of Hand-Eye solvers

In the following section, the performance of different choices of Hand-Eye solvers in the algorithm pipeline is presented. Firstly, a *baseline* comparison using the synthetic datasets are performed, before the solvers are compared using the weakly excited KM real-world datasets. The figures showing the computed metrics are box-plots showing the mean in orange, the outer 25 percentiles outside the box and outliers as circles.

Noting the logarithmic scale on the Y-axis, Figures 4.11 and 4.13 shows that the synthetic planar dataset results in some of the solvers to generate estimates with higher errors than the results using the synthetic uniform dataset. It may also be observed in Figures 4.10 and 4.12 that the closed-form solvers, being entirely deterministic, do not yield variance in the measured metrics like the iterative optimization methods do.

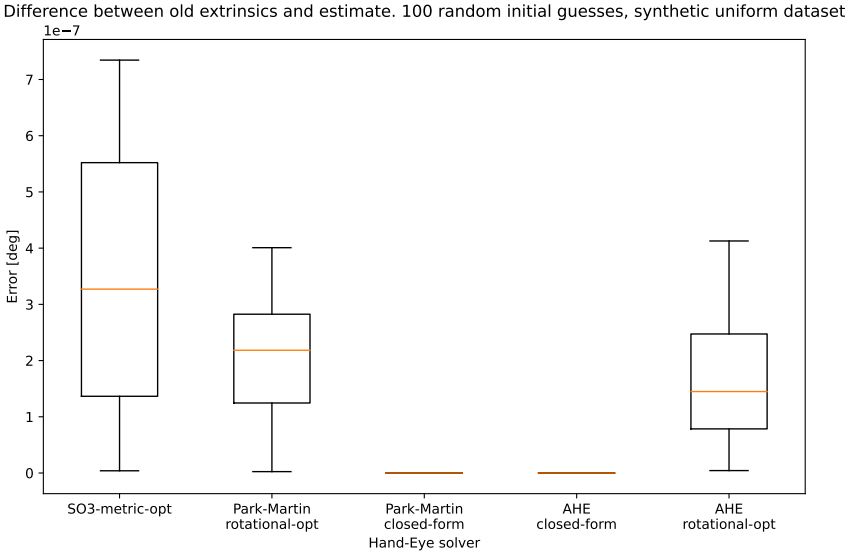


Figure 4.10: Comparison between choices of Hand-Eye solvers by measuring the difference between estimated extrinsics and the old parameters, with the synthetic uniform dataset as input.

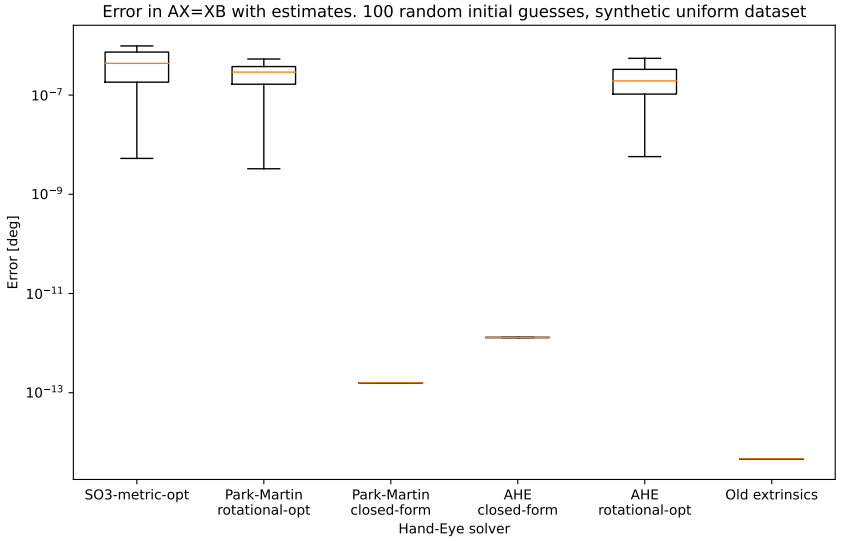


Figure 4.11: Comparison between choices of Hand-Eye solvers by measuring the difference between terms of the Hand-Eye equation when the estimate is inserted. The synthetic uniform dataset was used to generate the estimates.

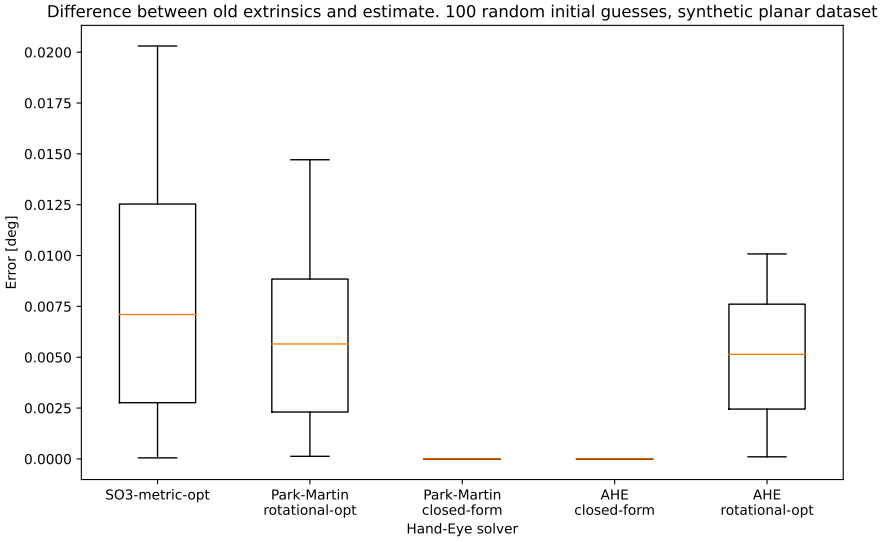


Figure 4.12: Comparison between choices of Hand-Eye solvers by measuring the difference between estimated extrinsics and the old parameters, with the synthetic planar dataset as input.

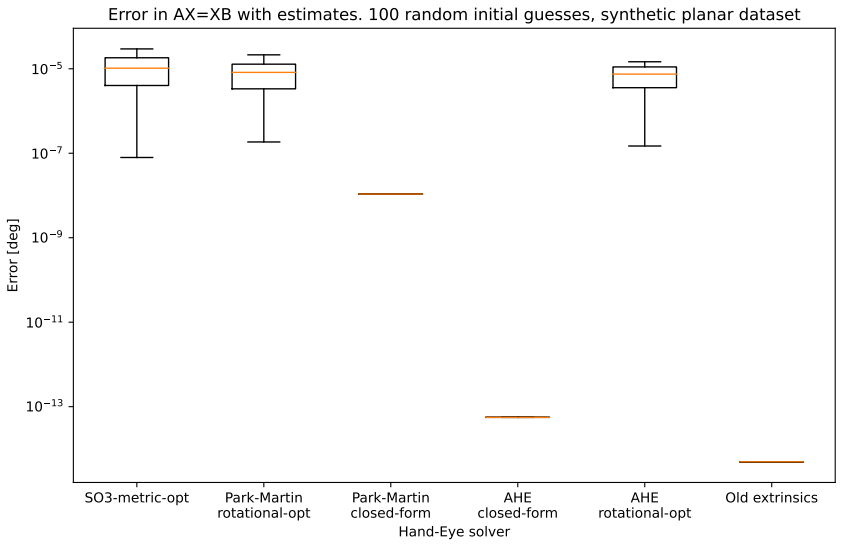


Figure 4.13: Comparison between choices of Hand-Eye solvers by measuring the difference between terms of the Hand-Eye equation when the estimate is inserted. The synthetic planar dataset was used to generate the estimates.

With baseline results established, the same analysis was performed on the Kongsberg Maritime weakly dataset, with results seen in Figures 4.14 and 4.15.

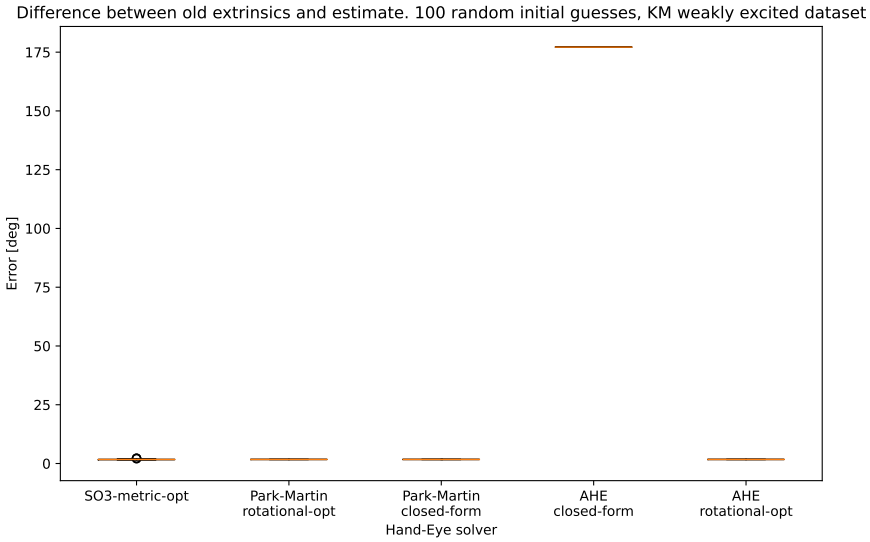


Figure 4.14: Comparison between choices of Hand-Eye solvers by measuring the difference between estimated extrinsics and the old parameters, with the KM weakly excited dataset as input. The relatively large error from the AHE closed-form-estimate makes the figure unreadable.

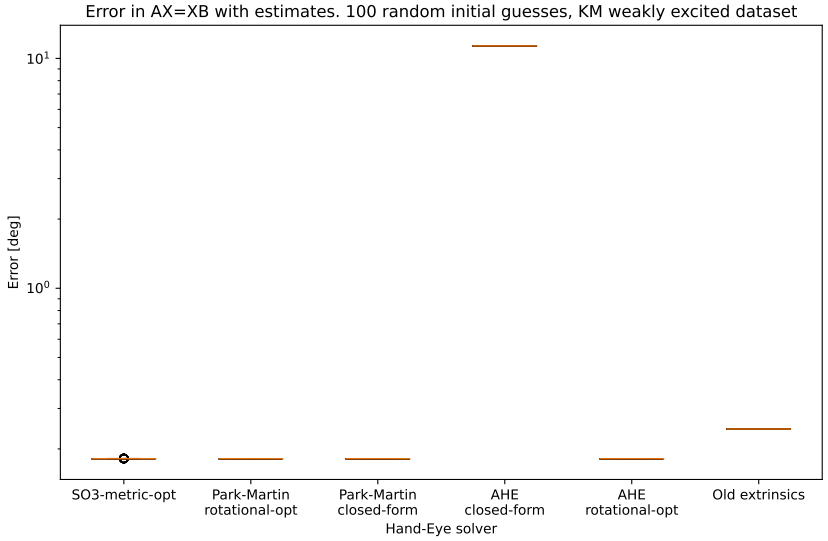


Figure 4.15: Comparison between choices of Hand-Eye solvers by measuring the difference between terms of the Hand-Eye equation when the estimate is inserted. The KM weakly excited dataset was used to generate the estimates. The relatively large error from the AHE closed-form-estimate makes the figure unreadable.

The results in Figures 4.14 and 4.15 using the KM weakly dataset are unreadable due to the high error from the estimate generated by the AHE closed-form method. Figures 4.16 and 4.17 display the same results, but without plotting results from the AHE closed-form method.

It is worth highlighting that the error shown in Figure 4.16 between the new estimate and old ground truth parameters is under 2° for all choices of Hand-Eye solver. In fact, the new estimates yield lower Hand-Eye error than the old extrinsics, as seen in Figure 4.17.

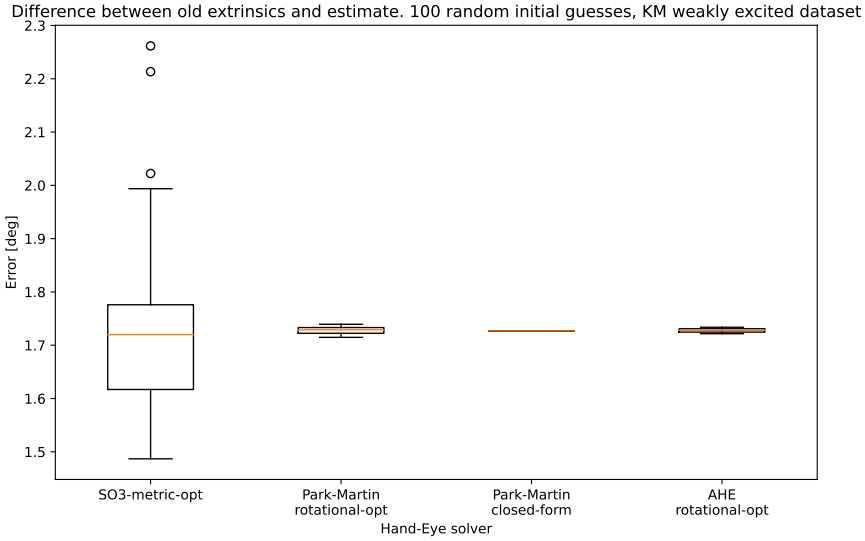


Figure 4.16: Comparison between choices of Hand-Eye solvers by measuring the difference between estimated extrinsics and the old parameters, with the KM weakly excited dataset as input. The figure has been excluded from plotting the results from the AHE closed-form solver due to high errors.

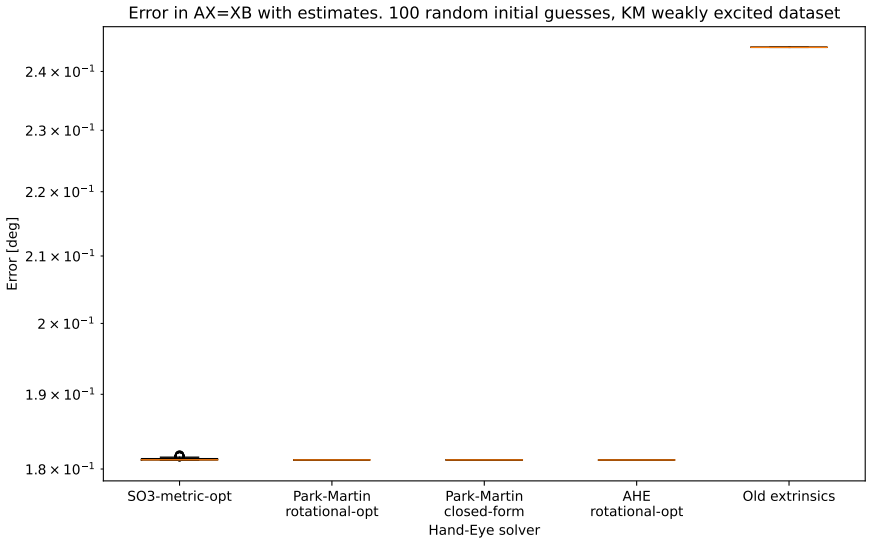
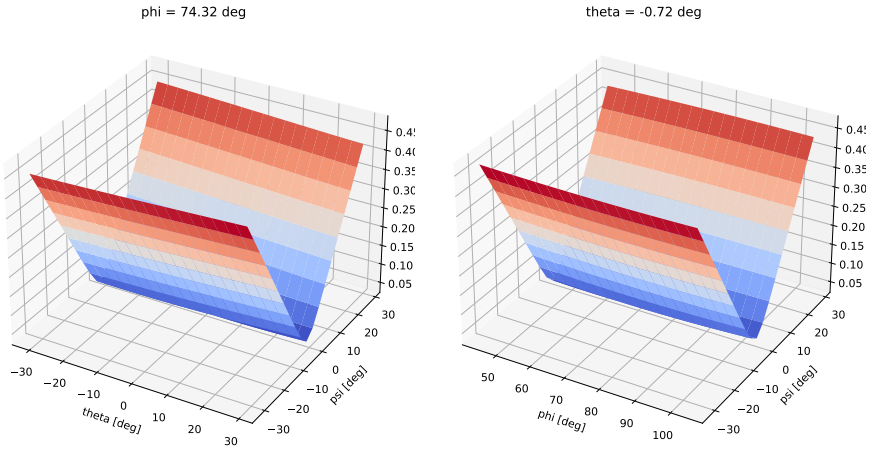


Figure 4.17: Comparison between choices of Hand-Eye solvers by measuring the difference between terms of the Hand-Eye equation when the estimate is inserted. The KM weakly excited dataset was used to generate the estimates. The figure has been excluded from plotting the results from the AHE closed-form solver due to high errors.

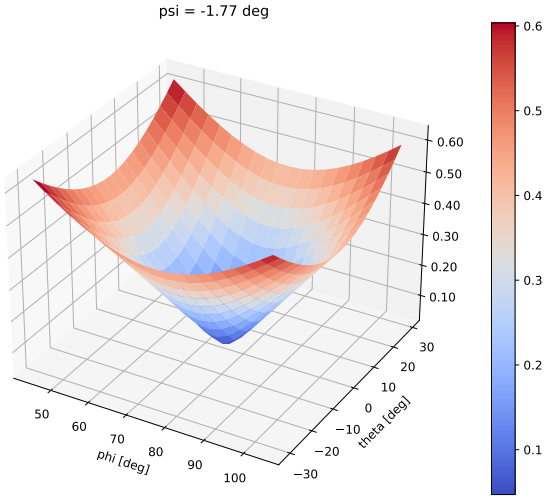
Focusing on the Hand-Eye optimization solver, the different choices for residual yielded similar results, but the SO(3)-metric yielded highest variance in the metrics but also the outer lowest error. Unlike the others, this residual was not explicitly derived from properties of the Hand-Eye equation. It was therefore analyzed further by sketching the cost-function.

Figure 4.18 sketches the cost-function given the weakly excited KM dataset as input. For each plot, one of the optimization variables is kept constant at the old assumed ground truth value, as to enable plotting the cost as the height at a given point in the parameter space. Note particularly how in Figures 4.18a and 4.18b, how the cost is almost only dependent on changes in the ψ Euler-angle.



(a) ϕ held constant at 74.32°

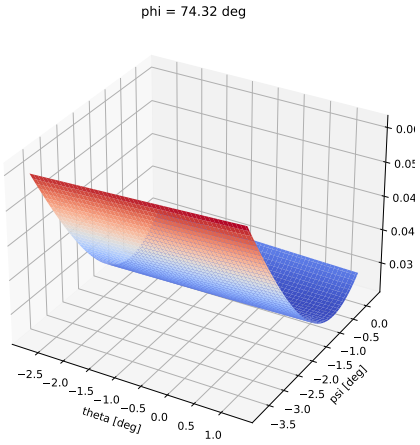
(b) θ held constant at −0.72°



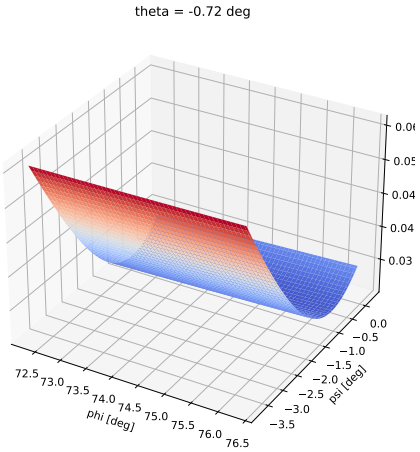
(c) ψ held constant at −1.77°

Figure 4.18: Sketch of the SO(3)-metric cost-function, using the weakly excited KM dataset as input data.

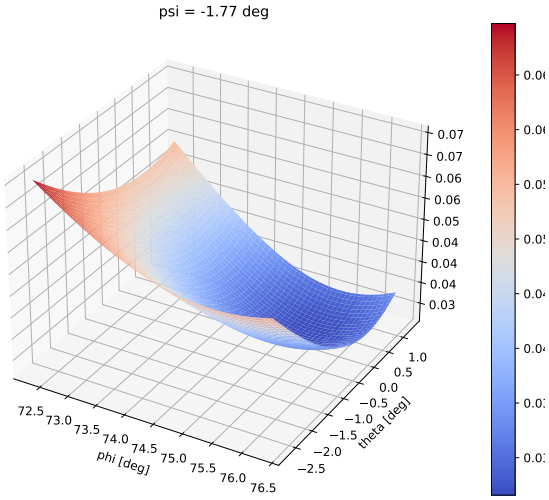
Figure 4.19 shows the same cost-function, closer around the assumed ground truth values and with higher resolution. Note the smoothness of the curve and the apparent lack of multiple isolated local minima.



(a) ϕ held constant at 74.32°



(b) θ held constant at −0.72°



(c) ψ held constant at −1.77°

Figure 4.19: Sketch of the SO(3)-metric cost-function, using the weakly excited KM dataset as input data. The cost-function has been zoomed in around the ground truth values.

The distinctive shape of the cost-function was analyzed further by attempting to eliminate the dataset as leading cause for the shape. This was done by performing the same analysis on a modified version of the uniformly random dataset where the ground truth extrinsics were all set to 0, seen in Figure 4.20. It is clear that this cost-function, as opposed to that in Figure 4.18, looks equal for all three Euler-angles.

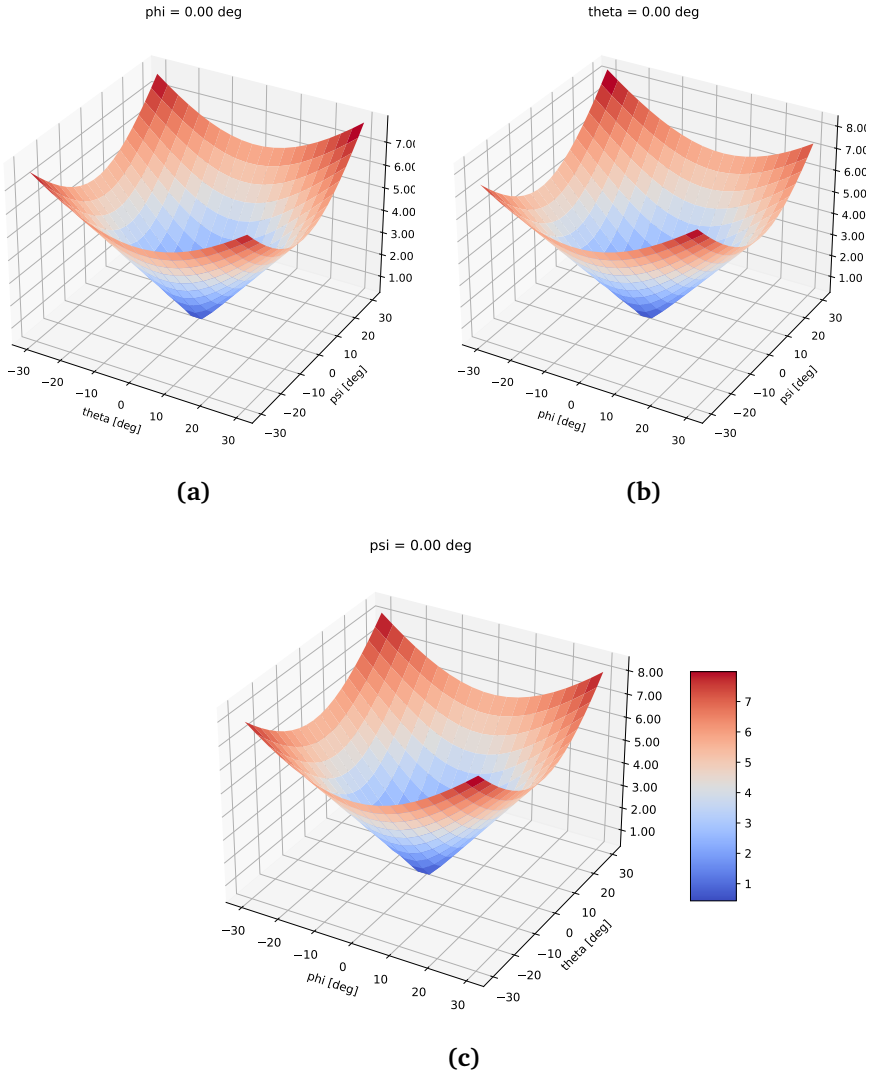


Figure 4.20: Sketch of the SO(3)-metric cost-function, using the synthetic uniform dataset as input data.

An analysis was performed to see if using a dataset which in theory is more excited may improve the estimates. Figures 4.21 and 4.22 show the same comparison of Hand-Eye solvers as done previously, but using the strongly excited KM dataset. Note how the error both between the extrinsic and in the Hand-Eye equation seemingly are higher than that of the weakly excited dataset. Note also the high amount of outliers present in the results generated by the Park Martin and AHE optimization functions in Figure 4.21, and how both Park-Martin solvers' results are worse than that of the SO(3)-metric. The AHE closed-form solver once more yielded too high errors to be included without obscuring the other results.

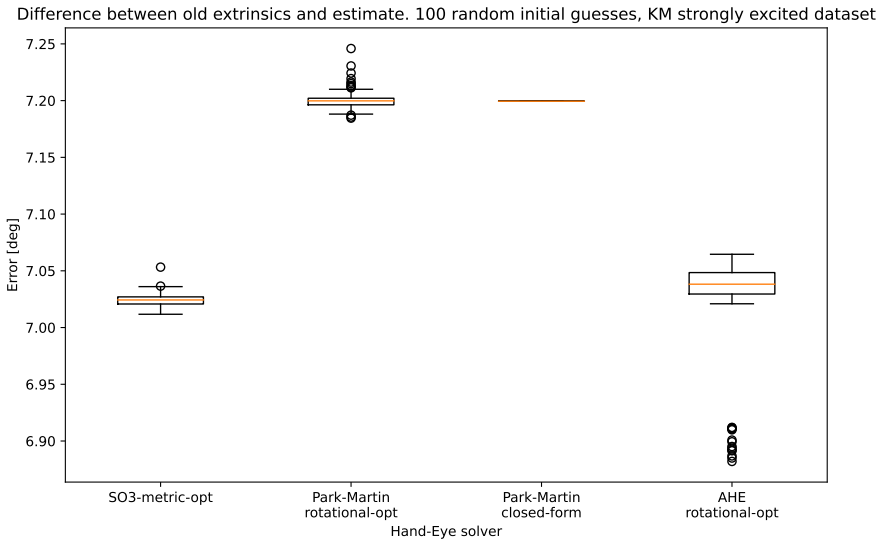


Figure 4.21: Comparison between choices of Hand-Eye solvers by measuring the difference between estimated extrinsics and the old parameters, with the KM strongly excited dataset as input. The figure has been excluded from plotting the results from the AHE closed-form solver due to high errors.

Lastly, it can be argued that using the same dataset for both generating an estimate and testing its performance is bad practice. The estimate will tend to overfit to the data, and especially the error in Hand-Eye equation metric is therefore expected to be low. To test for this, a run of comparing the different Hand-Eye solvers was performed were every second datapoint was used for estimating and every other datapoint was used for calculating the metrics. This way, estimation and evaluation was separated. This was performed with the KM weakly excited dataset, due to reliable results using this dataset earlier in this section.

The results may be seen in Figure 4.23. Comparing the result from splitting the dataset with the results from not doing so, Figure 4.17, the computed errors are about 3 times higher when splitting the dataset. The error is still somewhat lower than the results obtained when using the old extrinsics as input to the metric.

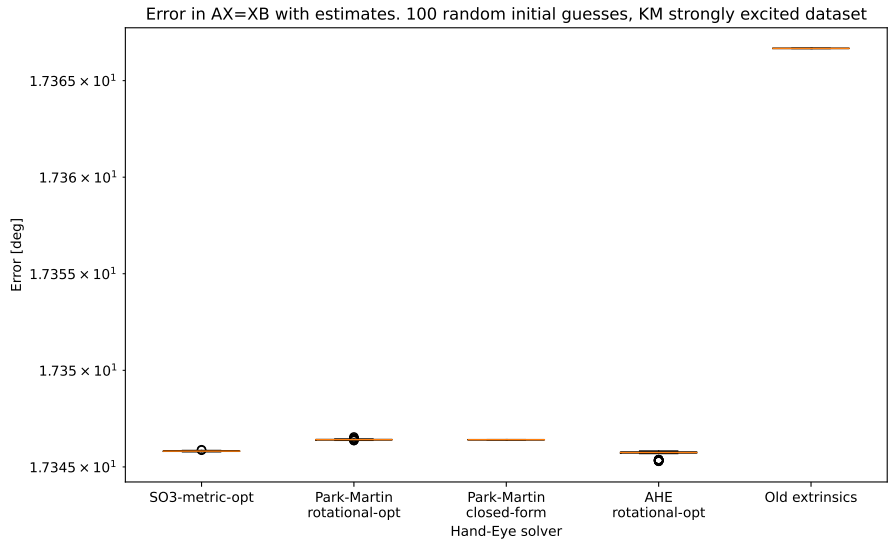


Figure 4.22: Comparison between choices of Hand-Eye solvers by measuring the difference between terms of the Hand-Eye equation when the estimate is inserted. The KM strongly excited dataset was used to generate the estimates. The figure has been excluded from plotting the results from the AHE closed-form solver due to high errors.

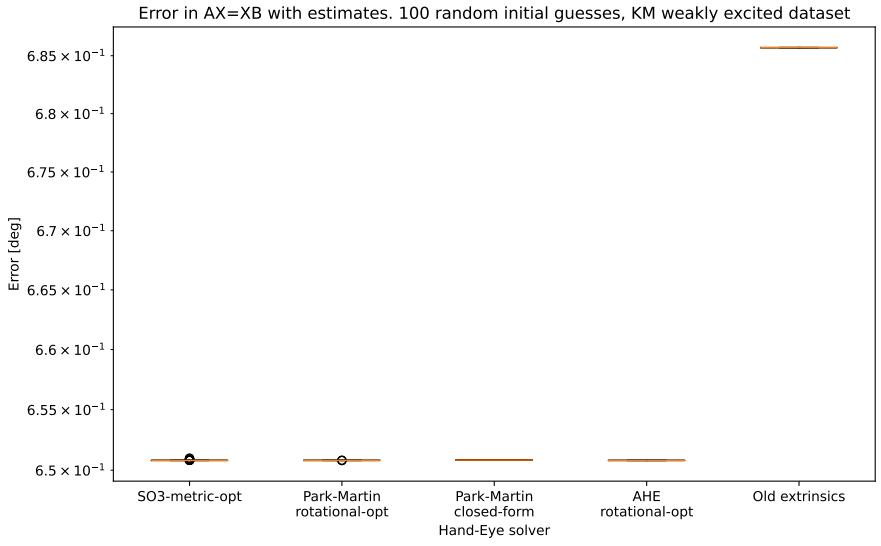


Figure 4.23: Comparison between choices of Hand-Eye solvers by measuring the difference between terms of the Hand-Eye equation when the estimate is inserted. Every second datapoint of the KM weakly excited dataset was used to generate the estimates, and every other datapoint was used to calculate the metrics.

Chapter 5

Discussion

5.1 Validity of reconstructions

When comparing the reconstructed camera poses with each other and with the movement of the ship, it was shown in Figure 4.5 that the reconstructions drift up to 25° over time. The COLMAP-reconstruction seemed to drift less, with error around 15°. The large amount of drift is expected to affect the accuracy of the Hand-Eye solvers. At the same time, it should be noted that the Hand-Eye solvers yield satisfactory results even with the drift present. The metric used for generating these results is not a proven method, and it may be erroneously representing the drift as worse than it actually is, if it was compared to the ground-truth camera poses. This last point is especially of interest when observing how all three sets of comparisons in Figure 4.5 have the same general shape, a fact one would not expect if these were truly independent. As discussed in Chapter 3, steps to minimize this drift in further work could be to use a VSLAM-method, potentially aided by the ship-measurements, as well as choosing shorter timespans when selecting batches of data to use in the estimation.

5.2 Using the qualitative measure of excitation

The presented qualitative measure of Hand-Eye excitation was tested on all four datasets used in this project. The results in Figures 4.6 and 4.7 showed high amounts of type 1 excitation for both synthetic datasets, while type 2 excitation was shown to be higher in the synthetic uniform than the synthetic planar dataset. These results accurately reflect the behaviour of the datasets, in that both the planar and uniform datasets are expected to have a good spread of poses, but the planar dataset only contains poses in the plane. This last point may be quickly identified by observing the rightmost plot of Figure 4.7, where the lack of any color reflects the parallel orientation axes.

The results obtained using the real-world datasets in Figures 4.8 and 4.9 similarly displayed expected results regarding the type 1 excitation. The weakly excited

dataset resulted in a histogram of angles shifted further away from the theoretical optimum of 180° than the strongly excited dataset did. The fact that the results reflect the intuitively expected outcome corroborates the presented method for evaluating the type 1 excitation present in datasets.

When it comes to using the presented method to evaluate the type 2 excitation present in the datasets, the results obtained with real-world data are not as consistent with intuition. Both real-world datasets display similar amounts of type 2 excitation, which is measured to be highest in the early datapoints and decreasing with time. The expected result would be for the weakly excited KM dataset to show lower type 2 excitation than the strongly excited. The fact that the excitation decreases over time is also dubious.

One potential source of this error could be the following: Since ship-movements are generally smooth, poses near each other in time will be very similar. When all the relative poses are computed relative the first pose, the first few poses are expected to have very small rotations due to them being so similar to the first. Numerical instability of the SO(3)-logarithm about the identity could lead to small rotation axes, whose direction are easily affected even by weak noise. This could then be the cause of the rotation axes seemingly being at an angle of 90° relative each other for the first few datapoints, even though the actual orientation difference is very small due to the vectors having such small lengths.

This source of error could be attempted to be compensated for in further work by finding some other mathematical expression for evaluating the angle between rotation axes which foregoes computing the SO(3)-logarithm.

It is expected for these results to change if the strategy for computing relative poses is changed, but without further theoretical frameworks to lean on it is difficult to predict exactly how. Choosing to use ship-measurements over camera-reconstructions as a baseline for these figures is only expected to change whether noise in the measurements or reconstructions are allowed to corrupt the results.

Nevertheless, it is clear that the weakly excited KM dataset is less excited than the synthetic uniform dataset, when evaluation of excitation is performed in the way presented in this report. The results obtained with the weakly excited KM dataset may then be interpreted as worst-case results, and that employing this pipeline on a system with higher possibility of excitation than the large cruise-ship in the weakly excited dataset should lead to better results, given that levels of noise between them are comparable.

5.3 Choice of Hand-Eye solvers

The comparison of Hand-Eye solvers on the synthetic datasets in Figures 4.10 to 4.13 showed all tested Hand-Eye solvers to perform more than adequately on the noise-free and highly excited synthetic uniform dataset, with slightly worse, but still acceptable, results on the noise-free synthetic planar dataset. This proves the Hand-Eye solvers to be functional for this project and shows a theoretical upper limit to performance one can expect. This conclusion is not a surprise, how-

ever, since all the solvers except the SO(3)-metric are proven methods in literature. The results on the synthetic datasets also give a pointer to how much the performance is expected to decrease when using a dataset consisting of mainly planar poses.

Inputting the worst-case weakly excited KM dataset with COLMAP-reconstruction gave resulted in an error between old extrinsics and the estimate of about 2°, and none of the methods provided a singular lower estimate than the SO(3)-metric. The results in Figures 4.16 and 4.17 show expectedly worse performance than that achieved using the synthetic planar dataset, as the real-world dataset is corrupted by both noise on the ship-measurements, drift in camera egomotion reconstruction and simplifications like the flat-earth approximation made in the pipeline.

The AHE closed-form solution diverges when used on the weakly excited KM dataset. This is an expected result due to the nature of it being a simultaneous solver means the rotational estimate diverges as the solver at the same time tries to estimate the unobservable height of the camera. This point is supported by the fact that the purely rotational AHE optimization solver does not diverge. An unexpected result, however, is that the AHE simultaneous solver does not diverge for the synthetic planar dataset, as seen in Figures 4.12 and 4.13. No further analyses were performed to attempt explaining this phenomenon, but one possible explanation could be that the AHE-solver just barely tackles the planarity in the synthetic planar dataset, but fails when the planarity is in combination with the noise and drift found in the weakly excited KM dataset. This is just conjecture however, and the exact cause of this inconsistency should be further explored by analyzing the outputs of the AHE simultaneous solver more closely.

For future work in real-time operations, robustness and consistency of estimates is more important than marginally lower errors. The results in this project then suggests that use of iterative optimization solvers, which to a lesser degree diverges for degenerate data, is preferred. Of the tested residuals, the Park Martin rotational residual and AHE rotational residual provide the results with least variance, a property which yields robustness in real-time operations. Further, the Park Martin residual also has a closer connection to the qualitative measure of excitation, in that the residual features the rotation axes of the poses. It may therefore be advantageous to see if a closer and more concrete connection between excitation and the solver's estimates may be made. One the other hand, the results from using the seemingly outlier-prone KM strongly excited reconstruction in Figure 4.21 may point to the Park Martin-residual being more sensitive to noise in input-data, an undesirable trait for robust real-time use.

5.3.1 Validity of comparisons

In this project, error bounds on the supposedly ground truth extrinsic calibration was not known for the real-world data. A consequence of this is that when the estimate generated by the pipeline is shown to be within 2° of the ground truth, it is not possible to know if this error is due to a wrongly generated estimate or if

the ground truth is 2° off from the *actual* orientation.

The second metric evaluated, the error between Hand-Eye terms, was an attempt to alleviate this and provide an objective measure of the performance of the pipeline. Evaluating this metric, Figure 4.17 showed the generated estimate consistently provided lower error than the old extrinsic parameters. However, an argument may be made that evaluating an estimate on the same baseline from which it is generated is misleading, as the estimate always will be made to best fit the data. The results shown in Figure 4.23 was therefore generated, where every second datapoint was used for estimation and every other for evaluation. The fact that the result presented in Figure 4.23 also yielded lower errors for the estimates than the ground truth value strengthens belief in that the generated estimates may in actuality be more accurate than the old assumed ground truth parameters.

Further, the analyses performed on synthetic datasets have exactly known values of the ground truth extrinsics by construction, and may therefore be more trustworthy when comparing Hand-Eye solvers. On the other hand, it has been shown in this report that the synthetic datasets do not accurately reflect properties of real-world data, and so these results should not be expected to be extendable to the real-world data.

5.3.2 Further analyses

The previous discussion of results has given baseline performance and critical analysis of the algorithm pipeline. What follows is a discussion of what may be considered as additional results.

With the algorithm having been evaluated on the weakly excited KM dataset, a similar analysis was performed using the strongly excited dataset to analyse the effect of excitation on the results. The strongly excited dataset yielded estimates which gave higher error in both of the measured metrics than when applying the weakly excited dataset, as illustrated in Figures 4.21 and 4.22. As pointed out when presenting the illustration of this datasets' camera motion reconstruction in Figure 4.4, it may seem that the reconstruction exhibits discontinuous jumps. Due to the level of noise in the reconstruction, it was difficult to conclude anything of value on the effect of better excitation for improving estimates of the extrinsic parameters. This conclusion also points to the importance of developing some method for strategically choosing the data with an optimal balance between noise and excitation, as well as the need for theory describing how noise on both ship-poses and camera-reconstruction propagates to the Hand-Eye solution.

The analysis of the SO(3)-metric cost-function over the weakly excited KM dataset in Figure 4.18 showed the shape to be convex, but less so in the degrees of freedom associated with pitch and roll. The large variance in results obtained using this cost-function may be explained by its valley-like shape, where the iterative optimization may get stuck at a point along the valley. The fact that sketching the same cost-function over the synthetic uniform dataset in Figure 4.20 does not results in the same distinctive valley-shape may point to the shape being a con-

sequence of the planarity of the data. It should also be noted that a linear change in Euler-angles not necessarily results in linear perturbation over the respective orientations' Lie algebra, meaning attempting to assign intuitive cause-and-effect to the shape of the cost-functions should be done with care.

5.4 Considerations when applying the presented method

The algorithm pipeline presented in this report is presented as a general framework for performing extrinsic calibration of ship-mounted camera. In theory, the pipeline should be able to compute reasonable estimates no matter what specific camera, structure-from-motion algorithm or ship attitude sensors are used. In reality however, any result achieved is highly dependent on the specifics of the experimental setup. One example of this is the fact that the results presented in this report are achieved using a Kongsberg Maritime Seapath-unit for measuring the ship poses. This unit is known for very good accuracy and a low amount of latency, which in turn leads to better results than for instance an IMU which was simply preintegrated. Even though the methodology presented in this report *in theory* should work when combined with any chosen method for measuring ship-poses, this bias towards high accuracy should be addressed. That said, most ships used in industrial applications are expected to have a navigational system of at least some accuracy, and the accuracy of these systems is also expected to improve with increased reliance on autonomy.

To achieve the results in this project, parts of the datasets where the ship was near structured land was chosen. This clearly leads to better reconstruction of the camera egomotion than if data from when the ship is at full sea is chosen, and one could argue the results thereby are skewed. This assumption is however deemed necessary to be able to use this methodology, supported by the fact that recalibration of extrinsic parameters today is done at shore by maintenance staff. It can also be defined as a prerequisite of the presented method, that it must be performed close to land.

Considering all this, the overall performance of the pipeline is still good, providing useable estimates of the extrisics which may even be better than the supposed ground truth. Figure 4.17 shows that it indeed is possible to employ Hand-Eye Calibration solvers on data from ships with rigidly mounted cameras to estimate the orientation of the camera. The simplifications and assumptions made are considered to be relatively reasonable for the application in question.

5.5 Additional ideas for future work

Moments for future work has been presented throughout the report, but some main ideas are presented here.

An obvious venue for future work is further analysis and handling of the degenerate nature of planar ship-data. If methods for detecting and avoiding de-

generacy in the estimation was possible, one may imagine the possibility of also estimating the camera position using the presented method.

Currently, the algorithm pipeline performs Structure from Motion over the captured images without using the measured ship-poses as priors. Aiding the reconstruction with the use of pose-priors are generally considered to provide better estimates, at least when these measurements are not very noisy. One may argue that separating the estimation of camera poses from the ship-pose measurements should be advantageous in separating sources of error, since the SfM reconstruction in theory should be just as consistent. In this project, however, the SfM reconstruction has been shown to be drifting to such a degree that it should be considered a major source of error. At any rate, it would be interesting to examine whether a better method to performing on-line SfM-based Hand-Eye Calibration could be based on iteratively building a camera egomotion reconstruction, aided by ship-measurements, while at the same time enforcing the Hand-Eye equation constraint for all relative pose pairs.

For the results presented in this report, much consideration has been made to be able to separate different sources of error on the estimated extrinsic parameters. These being error due to noise on ship-measurements, error due to drifting camera reconstructions and error due to degeneracy. The field of using egomotion reconstruction algorithms for the purpose of solving the Hand-Eye equation would be greatly benefitted if theoretical groundwork existed on how covariance in the different components propagate into the estimate extrinsic parameters.

It is also a fact that egomotion estimation may be performed on many different kinds of sensors, not only on cameras with the use of SfM. Another interesting direction to take research would then be to unify the method into a general framework for extrinsic calibration by combining egomotion algorithms with Hand-Eye solvers. Further, extending the methodology presented in this paper to work in real-time could present opportunities for generalizing the algorithm. One may for instance imagine formulating the Hand-Eye equation as a constraint to be optimized by a Factor Graph-framework, which inspired by current VSLAM-methods could lead to fusing uncertainties, different sensor models and egomotion algorithms.

Lastly, as noted multiple times in this report, it is the opinion of the author that the literature is critically lacking theory on the optimal strategy for constructing relative poses when large amounts of data is available. Solving this problem is key to enabling online calibration using the presented method, when higher amounts, but also more noisy, data than necessary is available.

Chapter 6

Conclusion

This project tested combining egomotion reconstruction-algorithms of camera-movement with solvers of the Hand-Eye Calibration problem, for the purposes of estimating the orientation of ship-mounted cameras.

The performance of the developed algorithm was evaluated through comparing the generated estimates to old ground truth values of the extrinsic calibration. The tests showed the algorithm was capable of estimating the orientation of cameras within 2° of the old extrinsics, and with lower error in a data-driven metric. This was achieved despite the camera reconstructions being shown to drift up to 15° over time in the evaluated dataset. The best choice of Hand-Eye solution method for ship-data was considered to be iterative optimization of a residual inspired by the work of Park and Martin, but more in-depth analyses should be performed to be able to conclude that this holds for the general case.

A qualitative measure was also presented for the purposes of enabling easier understanding of the amount of excitation in data input to the Hand-Eye calibration methods. Testing the method gave overall expected results, but further work can be done on improving the robustness of the measure.

Bibliography

[1] Y. Shiu and S. Ahmad, 'Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX=XB,' *IEEE Transactions on Robotics and Automation*, vol. 5, no. 1, pp. 16–29, 1989. DOI: 10.1109/70.88014.

[2] A. Tabb and K. M. A. Yousef, 'Solving the robot-world hand-eye(s) calibration problem with iterative methods,' *CoRR*, vol. abs/1907.12425, 2019. arXiv: 1907.12425. [Online]. Available: http://arxiv.org/abs/1907.12425.

[3] S. Ma and Z. Hu, 'Hand-eye calibration,' in *Computer Vision: A Reference Guide*, K. Ikeuchi, Ed. Boston, MA: Springer US, 2014, pp. 355–358, ISBN: 978-0-387-31439-6. DOI: 10.1007/978-0-387-31439-6_168. [Online]. Available: https://doi.org/10.1007/978-0-387-31439-6_168.

[4] N. Andreff, R. Horaud and B. Espiau, 'Robot hand-eye calibration using structure-from-motion,' *The International Journal of Robotics Research*, vol. 20, no. 3, pp. 228–248, 2001. DOI: 10.1177/02783640122067372. eprint: https://doi.org/10.1177/02783640122067372. [Online]. Available: https://doi.org/10.1177/02783640122067372.

[5] N. H. Khan and A. Adnan, 'Ego-motion estimation concepts, algorithms and challenges: An overview,' *Multimedia Tools and Applications*, vol. 76, pp. 16 581–16 603, 2017.

[6] D. Yang, B. He, M. Zhu and J. Liu, 'An extrinsic calibration method with closed-form solution for underwater opti-acoustic imaging system,' *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 9, pp. 6828–6842, 2020. DOI: 10.1109/TIM.2020.2976082.

[7] N. Roy, P. Newman and S. Srinivasa, 'Extrinsic calibration from per-sensor egomotion,' in *Robotics: Science and Systems VIII*. 2013, pp. 25–32.

[8] 'Kinematics,' in *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, Ltd, 2011, ch. 2, pp. 15–44, ISBN: 9781119994138. DOI: https://doi.org/10.1002/9781119994138.ch2. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119994138.ch2. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119994138.ch2.

[9] [Online]. Available: https://opensfm.org/.

[10] R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed. Springer Cham, 2022. DOI: 10.1007/978-3-030-34372-9.

[11] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004. DOI: 10.1017/CBO9780511811685.

[12] J. Solà, J. Deray and D. Atchuthan, 'A micro lie theory for state estimation in robotics,' *CoRR*, vol. abs/1812.01537, 2018. arXiv: 1812.01537. [Online]. Available: http://arxiv.org/abs/1812.01537.

[13] F. Park and B. Martin, 'Robot sensor calibration: Solving ax=xb on the euclidean group,' *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 717–721, 1994. DOI: 10.1109/70.326576.

[14] J. Fuentes-Pacheco, J. R. Ascencio and J. M. Rendon-Mancha, 'Visual simultaneous localization and mapping: A survey,' *Artificial Intelligence Review*, vol. 43, pp. 55–81, 2012.

[15] B. Lucas and T. Kanade, 'An iterative image registration technique with an application to stereo vision (IJCAI),' vol. 81, Apr. 1981.

[16] K. Daniilidis, 'Hand-eye calibration using dual quaternions,' *The International Journal of Robotics Research*, vol. 18, no. 3, pp. 286–298, 1999. DOI: 10.1177/02783649922066213. eprint: https://doi.org/10.1177/02783649922066213. [Online]. Available: https://doi.org/10.1177/02783649922066213.

[17] H. Longuet-Higgins, 'A computer algorithm for reconstructing a scene from two projections,' in *Readings in Computer Vision*, M. A. Fischler and O. Firschein, Eds., San Francisco (CA): Morgan Kaufmann, 1987, pp. 61–62, ISBN: 978-0-08-051581-6. DOI: https://doi.org/10.1016/B978-0-08-051581-6.50012-X. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B978008051581650012X.

[18] R. Tsai and R. Lenz, 'A new technique for fully autonomous and efficient 3d robotics hand/eye calibration,' *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, pp. 345–358, 1989. DOI: 10.1109/70.34770.

[19] J. Schmidt and H. Niemann, 'Data selection for hand-eye calibration: A vector quantization approach,' *The International Journal of Robotics Research*, vol. 27, no. 9, pp. 1027–1053, 2008. DOI: 10.1177/0278364908095172. eprint: https://doi.org/10.1177/0278364908095172. [Online]. Available: https://doi.org/10.1177/0278364908095172.

[20] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt and SciPy 1.0 Contributors, 'SciPy 1.0: Fundamental Algorithms for Scientific

Computing in Python,' *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: 10.1038/s41592-019-0686-2.

[21] J. Maye, P. Furgale and R. Siegwart, 'Self-supervised calibration for robotic systems,' in *2013 IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 473–480. DOI: 10.1109/IVS.2013.6629513.

[22] J. L. Schönberger and J.-M. Frahm, 'Structure-from-motion revisited,' in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[23] R. E. Miles, 'On random rotations in R^3,' *Biometrika*, vol. 52, no. 3/4, pp. 636–639, 1965, ISSN: 00063444. DOI: 10.2307/2333716. [Online]. Available: http://www.jstor.org/stable/2333716 (visited on 11/11/2022).

[24] K. Shoemake, 'Iii.6 - uniform random rotations,' in *Graphics Gems III (IBM Version)*, D. KIRK, Ed., San Francisco: Morgan Kaufmann, 1992, pp. 124–132, ISBN: 978-0-12-409673-8. DOI: https://doi.org/10.1016/B978-0-08-050755-2.50036-1. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780080507552500361.

[25] *Seapath 380 series*, accessed from https://www.kongsberg.com/globalassets/maritime/km-products/product-documents/seapath-380---utilising-gps-glonass-galileo-beidou-and-qzss, Kongsberg Maritime, Dec. 2018.

[26] S. Bianco, G. Ciocca and D. Marelli, 'Evaluating the performance of structure from motion pipelines,' *Journal of Imaging*, vol. 4, no. 8, 2018, ISSN: 2313-433X. DOI: 10.3390/jimaging4080098. [Online]. Available: https://www.mdpi.com/2313-433X/4/8/98.