Eye-to-Eye Calibration

Extrinsic Calibration of Multi-Camera Systems Using Hand-Eye Calibration Methods

Dipl.-Inf. Sandro Esquivel

Dissertation
zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
(Dr.-Ing.)
der Technischen Fakultät
der Christian-Albrechts-Universität zu Kiel
eingereicht im Jahr 2015

Kiel Computer Science Series (KCSS) 2015/4 v1.0 dated 2015-06-18

ISSN 2193-6781 (print version)
ISSN 2194-6639 (electronic version)

Electronic version, updates, errata available via https://www.informatik.uni-kiel.de/kcss

The author can be contacted via http://mip.informatik.uni-kiel.de

Published by the Department of Computer Science, Kiel University

Multimedia Information Processing Group

Please cite as:

▻ Sandro Esquivel. *Eye-to-Eye Calibration. Extrinsic Calibration of Multi-Camera Systems Using Hand-Eye Calibration Methods.* Number 2015/4 in Kiel Computer Science Series. Department of Computer Science, 2015. Dissertation, Faculty of Engineering, Kiel University.

```
@book{Esquivel15,
  author    = {Sandro Esquivel},
  title     = {Eye-to-Eye Calibration. Extrinsic Calibration of Multi-Camera Systems
               Using Hand-Eye Calibration Methods},
  publisher = {Department of Computer Science, Kiel University},
  year      = {2015},
  number    = {2015/4},
  series    = {Kiel Computer Science Series},
  note      = {Dissertation, Faculty of Engineering, Kiel University}
}
```

About this Series

The Kiel Computer Science Series (KCSS) covers dissertations, habilitation theses, lecture notes, textbooks, surveys, collections, handbooks, etc. written at the Department of Computer Science at the Christian-Albrechts-Universität zu Kiel. It was initiated in 2011 to support authors in the dissemination of their work in electronic and printed form, without restricting their rights to their work. The series provides a unified appearance and aims at high-quality typography. The KCSS is an open access series; all series titles are electronically available free of charge at the department's website. In addition, authors are encouraged to make printed copies available at a reasonable price, typically with a print-on-demand service.

Please visit `http://www.informatik.uni-kiel.de/kcss` for more information, for instructions how to publish in the KCSS, and for access to all existing publications.

1. Gutachter: Prof. Dr.-Ing. Reinhard Koch
Institut für Informatik
Christian-Albrechts-Universität zu Kiel

2. Gutachter: Prof. Dr.-Ing. Joachim Denzler
Lehrstuhl für Digitale Bildverarbeitung
Friedrich-Schiller-Universität Jena

Datum der mündlichen Prüfung: 10. Juni 2015

Zusammenfassung

Diese Arbeit beschäftigt sich mit der extrinsischen Kalibrierung von Mehrkamerasystemen ohne überlappende Sichtbereiche aus Bildfolgen. Die extrinsischen Parameter fassen dabei Lage und Orientierung der als starr-gekoppelt vorausgesetzten Kameras in Bezug auf ein gemeinsames Referenzkoordinatensystem zusammen. Die Minimierung der Redundanz der einzelnen Sichtfelder zielt dabei auf ein möglichst großes kombiniertes Sichtfeld aller Kameras ab. Solche Aufnahmesysteme haben sich in den letzten Jahren als hilfreich für eine Reihe von Aufgabenstellungen der Computer Vision erwiesen und sind daher von steigendem Interesse. Anwendungen, die von solchen Kamerakonfigurationen profitieren, finden sich etwa in den Bereichen der visuellen Navigation und der bildbasierten 3D-Szenenrekonstruktion. Um Messungen der einzelnen Kameras sinnvoll zusammenzuführen, müssen die Parameter der Koordinatentransformationen zwischen den Kamerakoordinatensystemen möglichst exakt bestimmt werden. Klassische Methoden zur extrinsischen Kamerakalibrierung basieren in der Regel auf räumlichen Korrespondenzen zwischen Kamerabildern, was ein überlappendes Sichtfeld voraussetzt.

Daher sollen in dieser Arbeit alternative Methoden zur Lagebestimmung von Kameras innerhalb eines Mehrkamerasystems untersucht werden, die auf der Hand-Auge-Kalibrierung basieren und Zwangsbedingungen starr-gekoppelter Bewegung ausnutzen. Das Problem soll dabei im Wesentlichen anhand von Bilddaten gelöst werden, also unter Verzicht auf zusätzliche Inertialsensoren oder odometrische Daten. Die daraus abgeleiteten extrinsischen Kalibrierverfahren werden in Anlehnung an die Hand-Auge-Kalibrierung als *Eye-to-Eye Calibration* bezeichnet. Es werden Lösungsverfahren vorgestellt, die ausschließlich auf Posemessdaten basieren und den Prozess der Poseschätzung von der eigentlichen Kalibrierung entkoppeln, sowie Erweiterungen, die direkt auf visuellen Informationen der einzelnen Kameras basieren.

Die beschriebenen Ansätze führen zu dem Entwurf eines Structure-from-Motion-Verfahrens, das Poseschätzung, Rekonstruktion der Szenengeometrie und extrinsische Kalibrierung der Kameras integriert. Bewegungskonfigurationen, die zu Singularitäten in den Kopplungsgleichungen führen, werden gesondert analysiert und es werden spezielle Lösungsstrategien zur partiellen Kalibrierung für solche Fälle entworfen. Dabei konzentrieren wir uns besonders auf den Fall von Bewegung in der Ebene, da dieser besonders häufig in Anwendungsszenarien auftritt, in denen sich das Kamerasystem in oder auf einem Fahrzeug befindet.

Abstract

The problem addressed in this thesis is the extrinsic calibration of embedded multi-camera systems without overlapping views, i. e., to determine the positions and orientations of rigidly coupled cameras with respect to a common coordinate frame from captured images. Such camera systems are of increasing interest for computer vision applications due to their large combined field of view, providing practical use for visual navigation and 3d scene reconstruction. However, in order to propagate observations from one camera to another, the parameters of the coordinate transformation between both cameras have to be determined accurately. Classical methods for extrinsic camera calibration relying on spatial correspondences between images cannot be applied here.

The central topic of this work is an analysis of methods based on hand-eye calibration that exploit constraints of rigidly coupled motions to solve this problem from visual camera ego-motion estimation only, without need for additional sensors for pose tracking such as inertial measurement units or vehicle odometry. The resulting extrinsic calibration methods are referred to as *eye-to-eye calibration*. We provide solutions based on pose measurements (*geometric eye-to-eye calibration*), decoupling the actual pose estimation from the extrinsic calibration, and solutions based on images measurements (*visual eye-to-eye calibration*), integrating both steps within a general Structure from Motion framework. Specific solutions are also proposed for critical motion configurations such as planar motion which often occurs in vehicle-based applications.

Acknowledgements

This thesis has come a long way – and it is not just my work but the outcome of a lot of helping hands and heads. I would like to take this opportunity to say thanks to a few of them.

During my time in the Multimedia Information Processing work group – first as student and research assistant, then as PhD student and research staff member – I was given the opportunity to attend a multitude of interesting, diverse and challenging projects including automatic measurement of sewers from video, engineering a modular software system for computer vision tasks, indoor reconstruction from panorama images and laser rangefinder data just to name a few. Beside these activities, I was able to take an active part in the development and realization of exercise courses for students which turned out as a matter of heart for me. These experiences allowed me multifaceted scientific insight into the theory, applications, and didactics of computer vision.

Therefore, first and foremost I would like to thank my supervisor Reinhard Koch for giving me the opportunity to work in such an inspiring topic area. This thesis would have never been finished without his helpful advice, considerate guidance, and encouragement.

I would also like to give acknowledgements to my former and present colleagues, especially Jan-Friso Evers-Senne and Kevin Köser for introducing me into the work group, Felix Woelk for supervising my diploma thesis, Ingo Schiller and Arne Petersen for tackling the KoSSE project with me, Bogumil Bartczak, Kristine Bauer, Markus Franke, Anatol Frick, Anne Jordt, Daniel Jung, Falko Kellner, Robert Wulff, and Lilian Zhang for personal and professional inspiration, and my current fellow researchers Johannes Brünger, Oliver Fleischmann, Andreas Jordt, Stefan Reinhold, Dominik Wolters, and Claudius Zelenka for their feedback and support during the recent time.

Furthermore, I owe a debt of gratitude to Renate Staecker, who kept me comfortably safe from most of the administrative duties and always held good advice in case of need, and Torge Storm, who reduced my technical workload to a minimum and skillfully counterbalanced my humble mechanical talents whenever a new camera system prototype had to be built.

Sincere thanks are dedicated to Joachim Denzler, Steffen Börm, and Dirk Nowotka for taking part in the assessment commission of this thesis.

Last but most vitally I would like to express my deepest appreciation for my family – Sybille, Mirko, Yanine, and Daniel – and friends who always backed me up during strenuous times, endured overtime hours and occupation by work, and grounded me when I was on the verge of going ballistic – reminding me of what is most important in life.

Sandro Esquivel
Kiel, June 2015

He stood in the center of Heaven and looked about it, having decided to have four eyes today. He noticed that with less than two looking in any one direction, he couldn't see as well as he ought. He resolved to set someone to discover the reason for this.

Steven Brust, "To Reign in Hell"

Illustration from: Andrea Alciati, *Emblemata*. Antwerpen, 1577, p. 106.

Contents

Symbols and Notations

Bold symbols are used for matrices, vectors, and points. Italic letters are used for vectors and scalars. In general, scalars are denoted by lowercase letters and matrices by uppercase letters. Coordinate frames, sets, and some special nonlinear functions are denoted by script letters.
The terms *position* and *translation*, *orientation* and *rotation*, resp. *pose* and *rigid motion* are used synonymously throughout this work.

Linear algebra

$\boldsymbol{x} \in \mathbb{R}^d$	point in d-dimensional Euclidean space
$\boldsymbol{x} \in \mathbb{P}^d$	point in d-dimensional projective space (i. e., $\boldsymbol{x} \in \mathbb{R}^{d+1} \backslash \{\mathbf{0}\}$)
$\boldsymbol{v} \in \mathbb{S}^d$	vector in d-dimensional unit sphere (i. e., $\boldsymbol{v} \in \mathbb{R}^{d+1}, \|\boldsymbol{v}\| = 1$)
$A_{i,j}$	entry at i-th row and j-th column of matrix $\mathbf{A}$
b_i	i-th entry of vector $\boldsymbol{b}$
$\boldsymbol{A}_i$, $\boldsymbol{a}_j$	i-th row or j-th column vector of matrix $\mathbf{A}$
$\mathbf{A}_{[i\ldots j,k\ldots \ell]}$	submatrix of $\mathbf{A}$ from i-th to j-th row and k-th to ℓ-th column
$\boldsymbol{b}_{[i\ldots j]}$	subvector of $\boldsymbol{b}$ from i-th to j-th entry
$\mathbf{I}_n$, $\mathbf{0}_{n\times m}$	$n \times n$ identity matrix ($\mathbf{I} = \mathbf{I}_3$) resp. $n \times m$ zero matrix
$[\boldsymbol{x}]_\times$	3×3 matrix describing cross product with $\boldsymbol{x} \in \mathbb{R}^3$

Camera model

$\boldsymbol{u} = (u, v\,[, 1])^\intercal$	2d pixel coordinates (Euclidean/homogeneous)
$\boldsymbol{x} = (x, y\,[, w])^\intercal$	normalized 2d point (Euclidean/homogeneous)
$\boldsymbol{X} = (X, Y, Z\,[, W])^\intercal$	3d point (Euclidean/homogeneous)
$\mathcal{K} : \mathbb{R}^3 \to \mathbb{P}^2$	camera function mapping 3d points to pixels
$\mathcal{P} : \mathbb{R}^3 \to \mathbb{P}^2$	perspective projection function
$\mathcal{S} : \mathbb{R}^3 \to \mathbb{S}^2$	spherical projection function
$\mathcal{D} : \mathbb{P}^2 \to \mathbb{P}^2$	image distortion function
$\mathcal{U} : \mathbb{P}^2 \to \mathbb{R}^3$	unprojection function

Poses and rigid motion

$[\mathbf{A} \mid \boldsymbol{b}] = \begin{pmatrix} \mathbf{A} & \boldsymbol{b} \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix}$	homogeneous affine transformation matrix with linear part $\mathbf{A}$ and translation $\boldsymbol{b}$
$\mathbf{T} = [\mathbf{R} \mid \boldsymbol{t}]$	rigid motion with rotation $\mathbf{R}$ and translation $\boldsymbol{t}$ or pose with orientation $\mathbf{R}$ and position $\boldsymbol{t}$
$\mathbf{S} = [\lambda\mathbf{R} \mid \boldsymbol{t}]$	similarity transformation with scale λ
$\mathbf{R}_{\boldsymbol{r},\alpha}$	rotation with angle α around axis $\boldsymbol{r}$
$\mathbf{q} = (\boldsymbol{q}, q)$	quaternion with vector part $\boldsymbol{q}$, scalar part q
$\check{\mathbf{q}} = (\mathbf{q}, \mathbf{p})$	dual quaternion with real part $\mathbf{q}$, dual part $\mathbf{p}$
$\mathbf{R}_{\mathbf{q}}$	rotation described by unit quaternion $\mathbf{q}$
$\mathbf{T}_{\check{\mathbf{q}}}$, $\mathbf{T}_{\mathbf{q},\mathbf{p}}$	rigid motion described by dual quaternion $\check{\mathbf{q}}$
$\mathcal{C}_i$, $\mathcal{W}_i$	reference/world coord. frame of i-th camera
$\mathbf{T}_{k,\ell}^{(i)} = [\mathbf{R}_{k,\ell}^{(i)} \mid \boldsymbol{t}_{k,\ell}^{(i)}]$	relative motion between ℓ-th and k-th pose of camera i
$\mathbf{T}_{k}^{(i)} = [\mathbf{R}_{k}^{(i)} \mid \boldsymbol{t}_{k}^{(i)}]$	k-th pose of camera i in its reference coordinate frame $\mathcal{C}_i$
$\mathbf{W}_{k}^{(i)} = [\mathbf{Q}_{k}^{(i)} \mid \boldsymbol{w}_{k}^{(i)}]$	k-th pose of camera i in its world coordinate frame $\mathcal{W}_i$
$\Delta\mathbf{T}_{i,j} = [\Delta\mathbf{R}_{i,j} \mid \Delta\boldsymbol{t}_{i,j}]$	*eye-to-eye transformation*, transformation from j-th to i-th camera coordinate frame in rig
$\Delta\mathbf{W}_{i,j} = [\Delta\mathbf{Q}_{i,j} \mid \Delta\boldsymbol{w}_{i,j}]$	*world-to-world transformation*, transformation from j-th to i-th world coordinate frame
$\boldsymbol{\mu} \in \mathbb{R}^{\mu}$, $\boldsymbol{\varrho} \in \mathbb{R}^{\varrho}$	general parameters for motion/rotation
$\mathbf{T}_{\boldsymbol{\mu}}$, $\mathbf{R}_{\boldsymbol{\varrho}}$	motion/rotation described by parameters $\boldsymbol{\mu}$, $\boldsymbol{\varrho}$
$\mathcal{M} : \mathbb{R}^{\mu} \times \mathbb{R}^3 \to \mathbb{R}^3$	3d point transform for motion parameters $\boldsymbol{\mu}$
$\mathcal{R} : \mathbb{R}^{\varrho} \times \mathbb{R}^3 \to \mathbb{R}^3$	3d point rotation for motion parameters $\boldsymbol{\varrho}$
$\mathcal{Q} : \mathbb{R}^3 \to \mathbb{S}^3$	lifting from minimal unit quaternion parameters $\boldsymbol{\varrho}$ to unit quaternion $\mathbf{q}_{\boldsymbol{\varrho}}$
$\Omega_{\mathrm{M}} \subset \mathbb{R}^{\mu}$, $\Omega_{\mathrm{R}} \subset \mathbb{R}^{\varrho}$	manifolds of valid motion/rotation parameter vectors for non-minimal parametrization
$\mathrm{proj}_{\mathrm{M}} : \mathbb{R}^{\mu} \to \Omega_{\mathrm{M}}$, $\mathrm{proj}_{\mathrm{R}} : \mathbb{R}^{\varrho} \to \Omega_{\mathrm{R}}$	projection onto manifold Ω_{M}, Ω_{R}, constraint enforcement for parameter vector $\boldsymbol{\mu}$, $\boldsymbol{\varrho}$

Chapter 1

Introduction

1.1 Motivation

In the recent years, embedded visual capturing systems composed of multiple cameras have proved useful in a variety of practical applications due to their large combined field of view. In the automotive industry, cameras are assembled on vehicles to merge information from front, rear, and side views for advanced driver assistent systems. Vehicle-mounted cameras are used for autonomous visual navigation and 3d reconstruction of urban scenes. Special camera rigs have gained popularity in filmmaking for the creation of digital visual effects. Further fields of application are edificial inspection, video surveillance and monitoring systems, and mobile devices for Augmented Reality. Major advantages of such devices with respect to omnidirectional cameras based on special lenses or mirrors are often lower costs, flexible configuration, and considerably higher resolution.

For many applications it is advantageous or even required to assemble cameras so that their fields of view are disjunct or have only minimal overlap (e. g., due to limitations on the number of cameras, specifications on camera locations, or small angular aperture of individual cameras) in order to provide the largest possible combined field of view. A well-known example is the *Point Grey Ladybug®5* spherical imaging system which is composed of six CCD cameras.[1] Multi-camera systems like these are especially helpful for *Structure from Motion* applications – i. e.,

[1] see Point Grey Research website at `http://www.ptgrey.com` for further details

3d reconstruction of an a priori unknown scene from images or video sequences captured under motion of the cameras – since the impact of critical motions is reduced and the problem of wide-baseline feature matching can be avoided in favor of temporal matching [FKK04].

However, in order to exchange observations between individual cameras or transform all measurements into a global coordinate system, the parameters of the coordinate transformations between camera images and the local 3d space have to be determined accurately. This process of *camera calibration* consists of finding the parameters of the imaging functions with respect to the camera models – the *intrinsic camera parameters* – and the relative poses of the cameras within the reference coordinate frame of the system – the *extrinsic camera parameters*. While the intrinsic parameters can be calibrated for each camera individually, estimation of the extrinsic parameters is in general based on mutual registration of cameras with respect to some jointly visible reference object.

1.2 Related Work

Classical multi-camera calibration extends the stereo calibration approach in a straightforward way. The prevalent strategy is to use a calibration pattern with known geometry that is visible in adjacent cameras and derive relative poses between cameras from absolute poses with respect to the calibration pattern [Zha99]. A more general approach is to detect corresponding image features in overlapping parts of simultaneously captured images that are used to compute the alignment of the cameras with each other [BKD09]. The latter approach is complicated by the fact that the stereo correspondence problem is in general difficult, especially if images of cameras with rather different resolution, distortion, and other imaging parameters have to be compared. Furthermore, approaches based on feature correspondences across cameras cannot be applied when the camera views have little or even no overlap.

Previous work on extrinsic camera calibration without overlapping views in the context of stationary camera networks approaches extrinsic cali-

bration by tracking moving objects between cameras and estimating the motion trajectory in unobserved areas [Jay04; RDD04; MEB04]). Such methods depend on the availability of a prior motion model of the observed objects. Another idea is to use a planar mirror to reflect a calibration object into the views of all cameras [Kum+08; HMR08]. Recently, Li et al. [Li+13] proposed to compose large feature-based calibration patterns that can also be detected when only small parts of them are visible. Both approaches are still limited by the physical setup of the camera system and depend largely on the construction of the calibration object resp. mirror. Liu et al. use a laser rangefinder with visible laser beam [LLZ14] or light planes created by a line laser projector [Liu+13] to locate cameras w.r.t. each other. Several proposed methods are dedicated explicitly to the case of car-mounted camera systems such as tracking features on the ground plane [KNS13] or traffic signs [Lam+07].

Motivated by results from structure and motion retrieval using stereo camera systems without stereo correspondences, such as first reported by Weng et al. [WH92], attempts were made to estimate the geometry of a multi-camera system from motion correspondences only. These approaches lead to a problem formulation that is closely related to hand-eye calibration.

Hand-eye calibration is a common problem originally addressed by the robotics community where a camera ("eye") is mounted on a mobile gripper ("hand") such that the relative position and orientation of the camera with respect to the gripper is fixed. The aim of hand-eye calibration is the estimation of the unknown coordinate transformation from the camera to the gripper coordinate frame or vice versa. Poses of the camera are estimated from images of a calibration object in the classical approach or more recently up to scale via Structure from Motion methods [AHE01]. This closely resembles the problem of extrinsic multi-camera calibration where the unknown but fixed coordinate transformation between the local camera coordinate frames has to be recovered from captured images.

Since the pioneering work on hand-eye calibration by Tsai & Lenz [TL89] and Shiu & Ahmad [SA89] in the late 1980s, the problem has been researched intensely first by the robotics community and later also by the

computer vision community in general. Several approaches and surveys, especially regarding critical configurations and motions, have been proposed so that hand-eye calibration is regarded as well understood by now although there is still active work on this topic. Although the classical hand-eye calibration was limited to a certain setup consisting of camera and robotic arm, it was stated in several works that the same method can be applied in fact for general coupled pose measurement devices (see for example [Iki00; DC03]).

The first notable approaches to extend hand-eye calibration in order to find the relative pose between two rigidly coupled cameras from ego-motion streams of each camera were made by Caspi & Irani [CI02] for the case of colocated cameras and Dornaika & Chung [DC03] for cameras in general spatial arrangement. In [EWK07], we generalized this approach by estimating the extrinsics of rigidly coupled cameras using a Structure from Motion approach, applying different extended hand-eye calibration methods, and considering solution strategies for degenerate motions. The idea of using Structure from Motion to estimate the camera position in hand-eye calibration was introduced by Andreff et al. [AHE01] and developed further by Schmidt et al. [SVN05]. In both approaches, the local ego-motion of the camera in the classical hand-eye setup is recovered using a monocular Structure from Motion approach instead of depending on a measured calibration pattern. This approach leads to a more complex problem since the absolute scale of camera translations is a priori unknown, but also relieves it from previous knowledge about the captured scene.

Since the publication of [EWK07], the proposed technique was extended into different directions, e. g., for vehicle-mounted camera systems [EG10; Pag10; PW10; PW11; Pag12b] or optimization of both camera motion and extrinsic parameters using a bundle adjustment approach [EG10; Léb+10]. More recent work on hand-eye calibration is also concerned with globally optimal calibration with respect to the L_∞-norm [RPK11; RPK12; HHP12], especially in the context of vehicle-mounted cameras.

1.3 Contributions

The basic idea of the presented approach to extrinsic camera calibration is to replace the "hand" in classical hand-eye calibration by another "eye" and estimate local poses of the coupled cameras using purely image-based methods such as Structure from Motion. The resulting framework is denoted by the neologism "eye-to-eye calibration".

The recent publications of Muhle [Muh11] and Pagel [Pag12a] are considered as closest to this thesis. Both works utilize hand-eye calibration methods for extrinsic calibration of multi-camera systems with non-overlapping views. Muhle considers rigidly coupled subsystems consisting of calibrated stereo cameras and estimates visual odometry from 3d/3d correspondences, immediately providing absolute scale for all camera motions. Pagel's work is restricted to vehicle-mounted cameras with certain arrangement constraints, e. g., all cameras view part of the ground plane. Therefore, Pagel considers planar motion subject to nonholonomic motion constraints only, resulting from the Ackerman steering mechanism of cars.

In comparison to [Muh11] and [Pag12a], this thesis is not restricted to special motions or camera setups. We provide a thorough discussion of extrinsic camera calibration without overlapping views based on rigid motion constraints with respect to different motion parametrizations, error measures, and a priori unknown absolute scale. We provide an analysis of degenerate motions or system configurations and provide partial solutions for underconstrained situations. Individual camera ego-motion is estimated from images using general Structure from Motion techniques, allowing for a larger field of application than more specific techniques like visual odometry or optical flow estimation. An original method to stabilize Structure from Motion with rigidly coupled cameras by integrating partial rigid motion constraints is proposed. Finally, global refinement of camera ego-motion and eye-to-eye calibration parameters with respect to image measurements is described.

An overview of the basic eye-to-eye calibration algorithm proposed in this work is shown in Fig. 1.1.

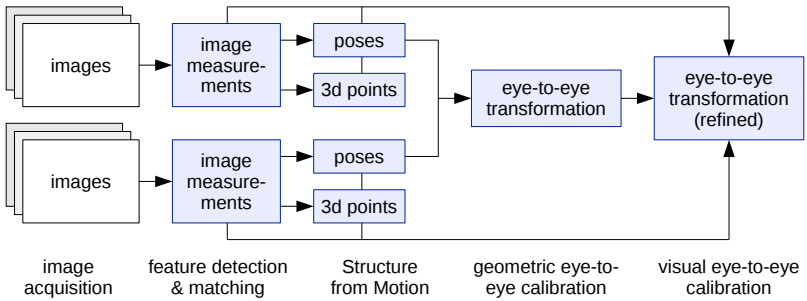

Figure 1.1. Basic overview of the proposed eye-to-eye calibration algorithm.

Parts of the contributions of this thesis have been described previously in:

- Sandro Esquivel, Felix Woelk, and Reinhard Koch: *Calibration of a Multi-Camera Rig from Non-Overlapping Views*, in: LNCS vol. 4713 (Proceedings of DAGM '07), 2007 [EWK07] (describes geometric eye-to-eye calibration and provides an analysis of critical motions)
- Sandro Esquivel, Reinhard Koch, and Heino Rehse: *Reconstruction of Sewer Shaft Profiles from Fisheye-Lens Camera Images*, in: LNCS vol. 5748 (Proceedings of DAGM '09), 2009 [EKR08] (applies geometric eye-to-eye calibration to a camera system consisting of two oppositely facing fisheye lens cameras)
- Sandro Esquivel and Stefan Gehrig: *Entwicklung eines Kalibrierverfahrens für fahrzeugmontierte Mehrkamerasysteme*, Abschlussbericht im Projekt AKTIV-AS, Teilprojekt KAS, Daimler AG, 07/2010 [EG10] (describes visual eye-to-eye calibration via bundle adjustment and partial eye-to-eye calibration from planar motion, not published)
- Sandro Esquivel and Reinhard Koch: *Structure from Motion Using Rigidly Coupled Cameras without Overlapping Views*, in: LNCS vol. 8142 (Proceedings of GCPR '13), 2013 [EK13] (describes Structure from Motion with partial rigid motion constraints)

1.4 Outline

The first part of this thesis is concerned with *geometric eye-to-eye calibration*, i. e., the estimation of the coordinate transformations between rigidly coupled cameras based on local ego-motion measurements, independent of the actual motion estimation process.

In **Chapter 2** we will describe the basics of coordinate transformations and rigid motion representation, introduce the nomenclature and typographic conventions used in this work, and refer to related problems.

In **Chapter 3** previous work on extrinsic calibration of rigidly coupled sensors is presented and algorithms for the solution of multi-camera calibration using hand-eye calibration techniques are described. We will first focus on the relative rotation between cameras. Afterwards we will consider the general relative pose and scale problem and discuss degenerate motion and camera configurations. Special care is taken of the case of planar motion which often occurs in practical applications, e. g., for vehicle-mounted multi-camera systems. The proposed methods are evaluated with synthetic data to analyze the impact of the motion configuration and accuracy on the calibration, leading to practical instruction for motion planning and selection.

The second part of this thesis is dedicated to *visual eye-to-eye calibration*, i. e., the estimation of the coordinate transformations between rigidly coupled cameras based on image measurements, in particular sparse feature point correspondences for individual cameras. The main topic is the integration of the calibration process into visual ego-motion estimation.

In **Chapter 4** we will describe the theoretical foundations of 3d reconstruction and pose estimation from camera images, i. e., the basic Structure from Motion problem and its subproblems, and give a brief overview of state-of-the-art methods.

Chapter 5 describes different applications of the rigid motion constraints in the context of Structure from Motion, leading to a general framework for extrinsic calibration of heterogeneous multi-camera systems. First, we will integrate eye-to-eye calibration into the general Structure from

Motion pipeline. Second, we will discuss the enforcement of rigid motion constraints in the camera pose estimation process. Third, joint optimization of camera poses and extrinsic camera parameters using a combined bundle adjustment approach is described. Finally, we consider globally consistent extrinsic calibration of multi-camera systems from pairwise calibration.

In **Chapter 6** we will evaluate the proposed framework by means of two real applications: First, we will consider pose tracking with a portable multi-camera system providing large motion variety. Second, we will consider the case of a vehicle-mounted multi-camera system which is limited to almost planar motion. A practical solution for this degenerate case is presented which includes partial knowledge about the captured scene and the camera rig configuration.

The results of this work and possible future work will finally be discussed and concluded in **Chapter 7**.

An elaborate description of the theoretical backgrounds and topics that go beyond the scope of the main work can be found in the appendices.

Appendix A goes into details of some aspects related to geometry such as quaternion algebra, rotation averaging, and absolute orientation.

Appendix B describes specific solutions for the subproblems of classical Structure from Motion and bundle adjustment for a single camera, and refers to extensions to other camera models such as spherical cameras.

Appendix C is dedicated to mathematical basics such as linear algebra, numerical optimization, and uncertainty handling.

Part I

Geometric Eye-to-Eye Calibration

Chapter 2

Spatial Motion and Poses

2.1 Introduction

The main goal of this part is to analyze rigid motions of rigidly coupled sensors – restrictively referred to as *cameras* in the following while the compound of cameras is denoted as *camera system* or *rig* – with respect to different coordinate frames and to defer the relative pose between these cameras from uncertain pose measurements. When we refer to motion of different cameras as "rigidly coupled" we mean synchronous motion under the assumption that the poses of the cameras in the rig w.r.t. each other do not change over time.

This chapter introduces the basic mathematical concepts that are essential for this analysis. First, we will describe coordinate frames and their notations used in this work. Second, we will describe different representations for coordinate transformations describing isometric scaling and motion in 3d space with special attention on rotations. Finally, we will discuss some calibration problems related to rigid coordinate transformations that are similar to our problem such as hand-eye calibration, base-world calibration, and extrinsic camera calibration.

We assume that the reader is familiar with basic concepts of linear algebra such as matrix properties, operations, and decompositions and least squares problems. A brief overview can be found in Appendix C.

2.2 Local and Global Coordinate Frames

In this thesis, both spatial motion and poses of objects are described in terms of transformations between Euclidean coordinate systems that are rigidly linked to these objects, denoted as *coordinate frames*. Every camera has a fixed coordinate frame associated with it that describes the 3d space local to it and moves through time and space with it. This coordinate frame is denoted as the *local coordinate frame* $\mathcal{C}_k^{(i)}$ for the i-th camera at the k-th time step.

The *pose* of a camera is identified by the location and orientation of the associated local coordinate frame within a *reference coordinate frame* $\mathcal{C}_i$ related with an initial pose of the rig, i. e., by its origin $\boldsymbol{C}_k^{(i)}$ and the rotation $\mathbf{R}_k^{(i)}$ that transforms the local coordinate axes into the reference coordinate frame. The coordinate transformation of 3d points from $\mathcal{C}_k^{(i)}$ to the reference coordinate frame is performed by a Euclidean transformation $\mathbf{T}_k^{(i)}$ so that $\boldsymbol{X}' = \mathbf{T}_k^{(i)}\boldsymbol{X}$ where $\boldsymbol{X}$ are the local coordinates in $\mathcal{C}_k^{(i)}$ and $\boldsymbol{X}'$ are the coordinates in the reference coordinate frame respectively. This transformation also consists of the rotation $\mathbf{R}_k^{(i)}$ and the translation vector $\boldsymbol{t}_k^{(i)} = \overrightarrow{\boldsymbol{OC}}_k^{(i)}$, hence we will refer to it as the *relative pose* of the camera with respect to its reference coordinate frame in the following. Similarly, the relative pose of the i-th camera from the ℓ-th to the k-th time step is given by the Euclidean transformation $\mathbf{T}_{k,\ell}^{(i)} = (\mathbf{T}_k^{(i)})^{-1}\mathbf{T}_\ell^{(i)}$. In general, a 3d point is transferred from the local coordinate frame of the j-th camera at the ℓ-th time step to the local coordinate frame of the i-th camera at the k-th time step by the Euclidean transformation $\mathbf{T}_{k,\ell}^{(i,j)} = (\mathbf{T}_k^{(i)})^{-1}\mathbf{T}_\ell^{(j)}$.

All reference coordinates frames $\mathcal{C}_i$ are embedded into a *global coordinate frame* $\mathcal{C}^{\mathrm{G}}$. We are mainly interested in the transformation from the j-th to the i-th reference coordinate frame – the *eye-to-eye transformation* $\Delta\mathbf{T}_{i,j}$ from camera j to i. These transformations are supposed to be constant over time and define the *extrinsic parameters* of the camera system we are interested in. Figure 2.1 illustrates the different nested coordinate frames and the respective coordinate transformations between them.

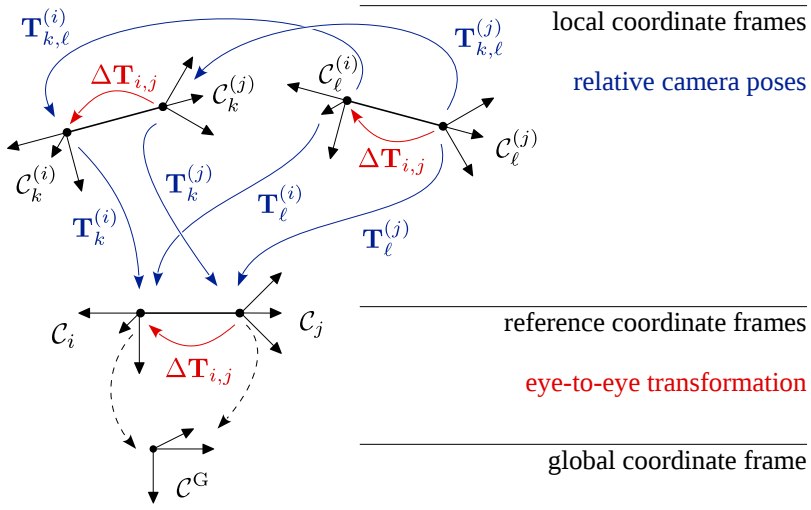

Figure 2.1. Local and global coordinate frames associated with rigidly coupled cameras and coordinate transformations between them.

The poses above relate to an initial pose of the camera system referred to as *reference pose*. In some situations the camera poses are known with respect to an individual world coordinate frame $\mathcal{W}_i$ (e. g., defined by certain 3d markers) embededd into the global world coordinate frame $\mathcal{W}^{\mathrm{G}}$ for each camera where the relative pose between the individual world coordinate frames is not a-priori known. These poses are referred to as *absolute poses* $\mathbf{W}_k^{(i)} = [\mathbf{Q}_k^{(i)} \mid \boldsymbol{w}_k^{(i)}]$ in the context of this work in order to distinguish them from the relative poses defined above. The transformation relating 3d points in the world coordinate frame $\mathcal{W}_j$ associated with camera j to the i-th camera's world coordinate frame $\mathcal{W}_i$ is denoted as *world-to-world transformation* $\Delta\mathbf{W}_{i,j}$. Figure 2.2 shows the relationship between camera coordinate frames and world coordinate frames in comparison to Fig. 2.1.

In the notations defined above, we use a superscript (i) and subscript i to distinguish poses and coordinate frames of different cameras within

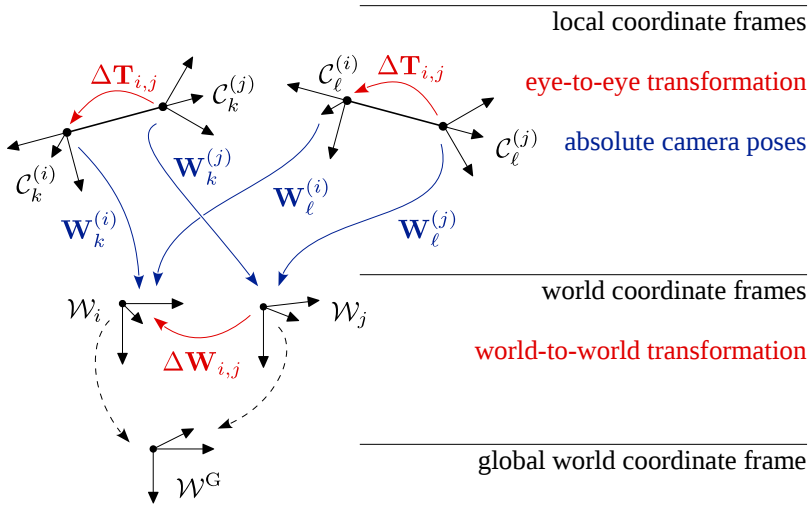

Figure 2.2. Absolute poses of rigidly coupled cameras with respect to individual world coordinate frames.

the rig, e. g., $\mathbf{T}_k^{(i)}$ vs. $\mathbf{T}_k^{(j)}$ and $\mathcal{C}_i$ vs. $\mathcal{C}_j$. However, if we consider only two cameras, we will omit these indices for sake of readability and write $\mathbf{T}_k$, $\mathbf{T}'_k$ and $\mathcal{C}$, $\mathcal{C}'$ instead. We will also drop the subscript i, j for transformations between cameras and write for example $\Delta\mathbf{T}$ instead of $\Delta\mathbf{T}_{i,j}$.

Coordinate system In robotics, kinematics, and the related body of theory, a multitude of different conventions how to specify 3d coordinates exist in parallel. Although the actual choice of the coordinate system is arbitrary, we will consistently use a specific coordinate system definition[1] for sake of clearness. Throughout this work we will always use a *right-handed coordinate system* to describe the local coordinate frames of cameras and scene objects as illustrated in Fig. 2.3:

[1] The coordinate system described here is commonly used in computer vision applications, e. g., in the well-known computer vision library OpenCV [Bra00].

- ▻ The origin coincides with the center of projection.
- ▻ The x-axis $\boldsymbol{e}_\mathrm{x}$ is the lateral axis pointing to the right.
- ▻ The y-axis $\boldsymbol{e}_\mathrm{y}$ is the vertical axis pointing down.
- ▻ The z-axis $\boldsymbol{e}_\mathrm{z}$ is the longitudinal axis pointing forwards.
- ▻ Rotation around $\boldsymbol{e}_\mathrm{x}$, $\boldsymbol{e}_\mathrm{y}$ and $\boldsymbol{e}_\mathrm{z}$ is denoted as *tilt*, *pan*, and *roll* rotation respectively. Positive tilt, pan, and roll angles rotate $\boldsymbol{e}_\mathrm{y}$ towards $\boldsymbol{e}_\mathrm{z}$, $\boldsymbol{e}_\mathrm{z}$ towards $\boldsymbol{e}_\mathrm{x}$, and $\boldsymbol{e}_\mathrm{x}$ towards $\boldsymbol{e}_\mathrm{y}$ respectively.

For perspective cameras, the plane of projection is located at $z = 1$ parallel to the x/y-plane within the local camera coordinate frame (see Sec. 4.2).

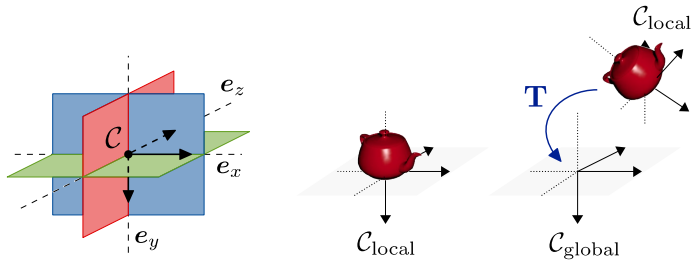

Figure 2.3. Right-handed coordinate system used in this work (left) and nested local and global coordinate frames (right).

2.3 Rigid Motion

In the following, we will have a closer look at the definition and proper parametrization of poses and motion, i. e., rigid coordinate transformations in 3d space.[2] We will discuss the parametrization of rotations in details and introduce metrics on poses that will be used in the formal description of

[2] Although the term "rigid motion" is commonly used in literature to restrict the class of transformations to rotation and translation without deformations, we will omit "rigid" here in general to avoid confusion with the term "rigidly coupled".

eye-to-eye calibration later. For an in-depth tutorial on the parametrization of rigid 3d transformations we refer the reader to [Bla12].

2.3.1 Basic Definitions

Motion[3] of a point $\boldsymbol{X} \in \mathbb{R}^3$ in 3d space is described in terms of linear algebra as a *Euclidean transformation* $\mathbf{T} \in \mathrm{SE}(3)$ consisting of a rotation matrix $\mathbf{R} \in \mathrm{SO}(3)$ and translation vector $\boldsymbol{t} \in \mathbb{R}^3$:

$$\boldsymbol{X}' = \mathbf{R}\boldsymbol{X} + \boldsymbol{t} \tag{2.1}$$

Using the homogeneous representation $\boldsymbol{X} = (X, Y, Z, 1)^\intercal$, a Euclidean transformation can be written as a linear transformation with the 4×4-matrix $\mathbf{T}$:

$$\boldsymbol{X}' = \mathbf{T}\boldsymbol{X} \quad \text{with } \mathbf{T} = \begin{pmatrix} \mathbf{R} & \boldsymbol{t} \\ \mathbf{0}^\intercal & 1 \end{pmatrix} \tag{2.2}$$

or in short: $\mathbf{T} = [\mathbf{R} \mid \boldsymbol{t}]$.

The rotational part is an orthonormal 3×3-matrix with determinant 1 describing the linear transformation of 3-vectors $\boldsymbol{v} \in \mathbb{R}^3$ under rotation, $\boldsymbol{v}' = \mathbf{R}\boldsymbol{v}$. Since a rotation matrix has 9 parameters and a rotation has only 3 degrees of freedom, there are 6 constraints on the entries of a rotation matrix given by the following quadratic equations between matrix rows $\boldsymbol{r}_i$, $i = 1, 2, 3$:

$$\boldsymbol{r}_i^\intercal \boldsymbol{r}_j = \begin{cases} 1 & \text{for } i = j \qquad \textit{(unit norm constraint)} \\ 0 & \text{for } i \neq j \qquad \textit{(orthogonality constraint)} \end{cases} \tag{2.3}$$

or equivalently: $\mathbf{R}^\intercal\mathbf{R} = \mathbf{I}$.

These matrices constitute the so-called *special orthogonal group* $\mathrm{SO}(3)$. The group of Euclidean transformations – the *special Euclidean group* $\mathrm{SE}(3)$ – can be understood as a manifold with structure $\mathrm{SO}(3) \times \mathbb{R}^3$ [Bla12].

[3] As mentioned in Sec. 2.2, we can model 3d motion (i. e., rotation and translation) and 3d pose (i. e., orientation and position) likewise using Euclidean transformations. Therefore, the terms "motion" and "pose" are used interchangeably throughout this work.

Composition of motions is performed by matrix multiplication of the corresponding Euclidean transformation matrices:

$$\mathbf{T} = \mathbf{T}_1\mathbf{T}_2 = [\mathbf{R}_1\mathbf{R}_2 \mid \mathbf{R}_1\boldsymbol{t}_1 + \boldsymbol{t}_2] \tag{2.4}$$

In terms of chronological order, $\mathbf{T}_2$ is performed first, then $\mathbf{T}_1$.

Since $\mathbf{R}$ is orthonormal, the inverse rotation $\mathbf{R}^{-1}$ is given by the transposed matrix $\mathbf{R}^\intercal$. The inverse Euclidean transformation matrix is hence given by:

$$\mathbf{T}^{-1} = [\mathbf{R}^\intercal \mid -\mathbf{R}^\intercal\boldsymbol{t}] \tag{2.5}$$

Given two Euclidean transformations describing *absolute motion* – i. e., coordinate transformations with respect to the same global coordinate frame – the *relative motion* is composed as follows:

$$\mathbf{T}_{1,2} = \mathbf{T}_1^{-1}\mathbf{T}_2 = \left[\mathbf{R}_1^\intercal\mathbf{R}_2 \mid \mathbf{R}_1^\intercal(\boldsymbol{t}_2 - \boldsymbol{t}_1)\right] \tag{2.6}$$

Similarities Similarity transformations[4] extend Euclidean transformations by an additional isometric scaling with a factor $\lambda \in \mathbb{R}_{>0}$. Considering a Euclidean transformation as a rotation followed by a translation, the scaling preceeds the translation. Hence, a similarity transformation can be represented by the homogeneous matrix $\mathbf{S} = [\lambda\mathbf{R} \mid \boldsymbol{t}]$. The partial transformation $\lambda\mathbf{R}$ is also denoted as *scaled rotation* here.

Obviously, a similarity transformation has 7 degrees of freedom, increasing the number by 1 with respect to a Euclidean transformation. The inverse similarity transformation is given by $\mathbf{S}^{-1} = [\frac{1}{\lambda}\mathbf{R}^\intercal \mid -\frac{1}{\lambda}\mathbf{R}^\intercal\boldsymbol{t}]$.

In the context of this work, a similarity transformation replaces a Euclidean transformation when coordinate frames with different metric units are related to each other.

Parametrization In order to abstract from the actual parametrization of a Euclidean transformation we will introduce the following notations.

[4] This denotes geometric similarity, not to be confused with matrix similarity.

Motion is described by a parameter vector $\boldsymbol{\mu} \in \mathbb{R}^{\mu}$ representing rotation by $\mathbf{R}_{\boldsymbol{\mu}}$ and translation by $\boldsymbol{t}_{\boldsymbol{\mu}}$. Rotation of a point $\boldsymbol{X} \in \mathbb{R}^3$ is described by a function $\mathcal{R}$:

$$\mathcal{R} : \mathbb{R}^{\mu} \times \mathbb{R}^3 \to \mathbb{R}^3, (\boldsymbol{\mu}, \boldsymbol{X}) \mapsto \mathbf{R}_{\boldsymbol{\mu}} \boldsymbol{X} \tag{2.7}$$

and translation is described by a function $\mathcal{T}$:

$$\mathcal{T} : \mathbb{R}^{\mu} \times \mathbb{R}^3 \to \mathbb{R}^3, (\boldsymbol{\mu}, \boldsymbol{X}) \mapsto \boldsymbol{X} + \boldsymbol{t}_{\boldsymbol{\mu}} \tag{2.8}$$

The complete Euclidean transformation $\mathbf{T}_{\boldsymbol{\mu}}$ is described by a function $\mathcal{M}$:

$$\mathcal{M} : \mathbb{R}^{\mu} \times \mathbb{R}^3 \to \mathbb{R}^3, (\boldsymbol{\mu}, \boldsymbol{X}) \mapsto \mathcal{R}(\boldsymbol{\mu}, \boldsymbol{X}) + \mathcal{T}(\boldsymbol{\mu}, \boldsymbol{X}) \tag{2.9}$$

For all motion parametrizations discussed in this work, rotation is described by a proper subset $\boldsymbol{\varrho} \in \mathbb{R}^{\varrho}$ of the motion parameters $\boldsymbol{\mu}$, so the domain of the rotation function can be reduced to:

$$\mathcal{R} : \mathbb{R}^{\varrho} \times \mathbb{R}^3 \to \mathbb{R}^3, (\boldsymbol{\varrho}, \boldsymbol{X}) \mapsto \mathbf{R}_{\boldsymbol{\varrho}} \boldsymbol{X} \tag{2.10}$$

The rotation matrix described by $\boldsymbol{\varrho}$ is referred to as $\mathbf{R}_{\boldsymbol{\varrho}}$ in this case.

Composing transformation parameters is described by a general function

$$\mathrm{comp}_{\mathrm{M}} : \mathbb{R}^{\mu} \times \mathbb{R}^{\mu} \to \mathbb{R}^{\mu} \quad \text{so that } \mathbf{T}_{\boldsymbol{\mu}_1} \mathbf{T}_{\boldsymbol{\mu}_2} = \mathbf{T}_{\mathrm{comp}_{\mathrm{M}}(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2)} \tag{2.11}$$

Computing parameters for the inverse transformation is described by:

$$\mathrm{inv}_{\mathrm{M}} : \mathbb{R}^{\mu} \to \mathbb{R}^{\mu} \quad \text{so that } \mathbf{T}_{\boldsymbol{\mu}}^{-1} = \mathbf{T}_{\mathrm{inv}_{\mathrm{M}}(\boldsymbol{\mu})} \tag{2.12}$$

The composition of $\mathrm{comp}_{\mathrm{M}}$ and $\mathrm{inv}_{\mathrm{M}}$ defines the relative transformation:

$$\mathrm{rel}_{\mathrm{M}} : \mathbb{R}^{\mu} \times \mathbb{R}^{\mu} \to \mathbb{R}^{\mu}, (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2) \mapsto \mathrm{comp}_{\mathrm{M}}(\mathrm{inv}_{\mathrm{M}}(\boldsymbol{\mu}_1), \boldsymbol{\mu}_2) \tag{2.13}$$

so that $\mathbf{T}_{\boldsymbol{\mu}_1}^{-1} \mathbf{T}_{\boldsymbol{\mu}_2} = \mathbf{T}_{\mathrm{rel}_{\mathrm{M}}(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2)}$. Generic functions $\mathrm{comp}_{\mathrm{R}}$, $\mathrm{inv}_{\mathrm{R}}$, $\mathrm{rel}_{\mathrm{R}}$ for composition, inversion, and relative transformation of rotation parameters $\boldsymbol{\varrho}$ are defined analogously.

For non-minimal parametrizations, i. e., $\mu > 6$, there are certain constraints for parameter vectors in order to represent a proper Euclidean transfor-

mation (e. g., the orthonormality constraints for the entries of a rotation matrix). Parameter vectors that satisfy these constraints are denoted as "valid" here. The manifold of valid pose parameter vectors is denoted as $\Omega_M \subset \mathbb{R}^\mu$. In general, there are different methods how to enforce these constraints for an invalid parameter vector $\hat{\boldsymbol{\mu}} \in \mathbb{R}^\mu$, e. g., via projection onto Ω_M. This is described by a general "projection" function

$$\text{proj}_M : \mathbb{R}^\mu \to \Omega_M \tag{2.14}$$

For the manifold of valid rotation parameters $\Omega_R \subset \mathbb{R}^\varrho$, the projection function proj_R is defined likewise.

2.3.2 Rotation Parametrization

The most common approach to motion parametrization in computer vision is to describe the translational part in terms of the translation vector $\boldsymbol{t} \in \mathbb{R}^3$, attended by some separate rotation parametrization $\boldsymbol{\varrho} \in \mathbb{R}^\varrho$. We will briefly introduce different rotation parametrizations here and point out their advantages and disadvantages for rotation estimation. Further discussion of this topic can be found in [Bla12; BT03; TC04]

Rotation matrices are simple to apply and compose, but are highly redundant and demand to keep track of a large number of constraints or come up with a reduced parametrization. Euler angles are minimal but suffer from singularities. The exponential map for rotation matrices is also a minimal representation but has to cope with singularities and is difficult to use for combining rotations. The same applies to angle/axis vector representations such as the Euler vector or Cayley-Gibbs-Rodrigues vector. Unit quaternions – or rather the Euler-Rodrigues vector representing unit quaternions – are considered as the optimal choice for rotation representation since they require only a single constraint and are easy to apply. It is also possible to represent them in reduced form for minimal parametrization.

Rotation matrix entries A rotation matrix can be described by the vector of its entries $\boldsymbol{\varrho} = \mathrm{vec}(\mathbf{R}) = (R_{1,1}, R_{1,2}, R_{1,3}, \ldots, R_{3,3})^\mathsf{T} \in \mathbb{R}^9$:

$$\mathbf{R}_{\boldsymbol{\varrho}} = \mathrm{mat}(\boldsymbol{\varrho}) = \begin{pmatrix} \varrho_1 & \varrho_2 & \varrho_3 \\ \varrho_4 & \varrho_5 & \varrho_6 \\ \varrho_7 & \varrho_8 & \varrho_9 \end{pmatrix}$$

This parametrization is well-defined and has no singularities. Another benefit is that composition, inversion, and vector rotation are given by simple linear transformations $\mathrm{comp}_\mathrm{R}(\boldsymbol{\varrho}_1, \boldsymbol{\varrho}_2) = \mathrm{vec}(\mathbf{R}_{\boldsymbol{\varrho}_1}\mathbf{R}_{\boldsymbol{\varrho}_2})$, $\mathrm{inv}_\mathrm{R}(\boldsymbol{\varrho}) = \mathrm{vec}(\mathbf{R}_{\boldsymbol{\varrho}}^\mathsf{T})$ and $\mathcal{R}(\boldsymbol{\varrho}, \boldsymbol{X}) = \mathbf{R}_{\boldsymbol{\varrho}}\boldsymbol{X}$.

However, a severe drawback is the amount of over-parametrization via $\varrho = 9$ parameters resulting in six quadratic constraints given by eq. (2.3). Hence, techniques for *orthonormalization* of an arbitrary non-zero matrix $\hat{\mathbf{R}} \in \mathbb{R}^{3\times3}$ have to be considered in the context of estimation from noise-corrupted data. A simple orthonormalization algorithm is given by the Gram–Schmidt method, although the result is not the optimal solution with respect to the matrix distance $d(\mathbf{A}, \mathbf{A}') = \|\mathbf{A} - \mathbf{A}'\|$. It can be shown that the orthonormal matrix $\mathbf{R}$ closest to $\hat{\mathbf{R}} \in \mathbb{R}^{3\times3}$ with respect to d can be obtained via singular value decomposition of $\hat{\mathbf{R}}$:

$$\mathbf{R} = \mathbf{U}\mathbf{V}^\mathsf{T} \tag{2.15}$$

where $\hat{\mathbf{R}} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\mathsf{T}$ is the singular value decomposition of $\hat{\mathbf{R}}$. This method can also be applied when the matrix $\hat{\mathbf{R}}$ is singular or nearly singular.

Euler angles The *Euler angles*[5] resp. *Tait-Bryan angles*[6] $\boldsymbol{\varrho} = (\alpha, \beta, \gamma) \in \mathbb{R}^3$ describe an arbitrary rotation in terms of rotation angles around the main axes $\boldsymbol{e}_\mathrm{x}$, $\boldsymbol{e}_\mathrm{y}$, and $\boldsymbol{e}_\mathrm{z}$ of the coordinate system, denoted as *tilt angle* α, *pan angle* β, and *roll angle* γ. There exist twelve possible sequences to apply these rotations in that must be specified.[7] For brevity, we consider only

[5] named after the German mathematician and physicist Leonard Euler (*1707, †1783)

[6] named after Peter Guthrie Tait (*1831, †1901) and George Hartley Bryan (*1864, †1928)

[7] Additional to the Tait-Bryan sequences XYZ, XZY, ... there are six sequences XYX, XZX, ... denoted as "classic" Euler angles.

the XYZ order ("roll first, then pan, final tilt") here which is given by:

$$\mathbf{R} = \mathbf{R}_{\boldsymbol{e}_x,\alpha}\mathbf{R}_{\boldsymbol{e}_\beta,\beta}\mathbf{R}_{\boldsymbol{e}_z,\gamma} = \begin{pmatrix} c_\beta c_\gamma & -c_\beta s_\gamma & s_\beta \\ c_\alpha s_\gamma + c_\gamma s_\alpha s_\beta & c_\alpha c_\gamma - s_\alpha s_\beta s_\gamma & -c_\beta s_\alpha \\ s_\alpha s_\gamma - c_\alpha c_\gamma s_\beta & c_\gamma s_\alpha + c_\alpha s_\beta s_\gamma & c_\alpha c_\beta \end{pmatrix} \quad (2.16)$$

where $c_\alpha := \cos(\alpha)$, $s_\alpha := \sin(\alpha)$, $c_\beta := \cos(\beta)$, $s_\beta := \sin(\beta)$, $c_\gamma := \cos(\gamma)$ and $s_\gamma := \sin(\gamma)$. To avoid singularities known as the *gimbal lock*, the Euler angles are usually limited to the range $\alpha, \gamma \in [-\pi, \pi)$ and $\beta \in [-\frac{\pi}{2}, \frac{\pi}{2})$.[8]

While Euler angles provide an intuitive minimal parametrization of rotations, they are not well suited for pose estimation due to the number of trigonometric function evaluations involved and the fact that deriving the parameters for composed rotations is cumbersome. Furthermore, metrics on Euler angles are not meaningful since rotations that are very close to each other can have very different Euler angle values.

Angle and axis Due to *Euler's rotation theorem*, any three dimensional rotation can be described by its *rotation axis*, i. e., a unit vector that is invariant under the rotation, and its rotation angle, i. e., the magnitude of rotation around the axis. The *angle/axis representation* $\boldsymbol{\varrho} = (\boldsymbol{r}, \alpha)$ using an angle α and a unit length vector $\boldsymbol{r} \in \mathbb{S}^2$ for 3d rotations is very common in robotics and computer vision. This parametrization is close to minimal with $\varrho = 4$ degrees of freedom and one quadratic constraint $\boldsymbol{r}^\mathsf{T}\boldsymbol{r} = 1$. To avoid ambiguities resulting from the facts that $(\boldsymbol{r}, \alpha)$, $(-\boldsymbol{r}, -\alpha)$, and $(\boldsymbol{r}, \alpha + 2n\pi)$ represent the same rotation, α is in general limited to the range $[0, \pi]$. The inverse rotation for $(\boldsymbol{r}, \alpha)$ is then trivially given by $(-\boldsymbol{r}, \alpha)$. However, a singularity for 180° rotations remains.

There are several methods to derive the rotation angle and axis from a rotation matrix. Since any vector along the axis remains fixed under the rotation (i. e., $\mathbf{R}\boldsymbol{r} = \boldsymbol{r}$), the rotation axis $\boldsymbol{r}$ for a rotation matrix $\mathbf{R}$ can be computed by solving the linear equation $(\mathbf{R} - \mathbf{I})\boldsymbol{r} = 0$ subject to the constraint $\|\boldsymbol{r}\|^2 = 1$ which yields the eigenvector of $\mathbf{R}$ related to the

[8] This applies to the given sequence XYZ only. In general, the angle of the second rotation is limited to $[-\frac{\pi}{2}, \frac{\pi}{2})$.

eigenvalue 1. Note however that $\boldsymbol{r}$ is undefined for $\alpha = 0$.

A closed-form solution to compute a rotation matrix for given rotation angle and axis can be derived from geometric considerations:

$$\mathbf{R} = \begin{pmatrix} c + r_x^2(1-c) & r_x r_y(1-c) - r_z s & r_x r_z(1-c) + r_y s \\ r_y r_x(1-c) + r_z s & c + r_y^2(1-c) & r_y r_z(1-c) - r_x s \\ r_z r_x(1-c) - r_y s & r_z r_y(1-c) + r_x s & c + r_z^2(1-c) \end{pmatrix} \tag{2.17}$$

where $c := \cos(\alpha)$ and $s := \sin(\alpha)$. We will denote the rotation matrix parametrized by its rotation axis $\boldsymbol{r}$ and rotation angle α by $\mathbf{R}_{\boldsymbol{r},\alpha}$ in the remainder of this work.

From eq. (2.17), the following extraction of the angle α and axis $\boldsymbol{r}$ from a rotation matrix $\mathbf{R}$ can be derived:

$$\operatorname{trace}(\mathbf{R}) = 1 + 2\cos(\alpha) \quad \text{i. e., } \alpha = \arccos\left(\frac{\operatorname{trace}(\mathbf{R}) - 1}{2}\right) \tag{2.18}$$

and

$$\boldsymbol{r} = \frac{1}{2\sin(\alpha)} \begin{pmatrix} R_{3,2} - R_{2,3} \\ R_{1,3} - R_{3,1} \\ R_{2,1} - R_{1,2} \end{pmatrix} \tag{2.19}$$

Equation (2.19) is undefined for $\alpha = 0$ and $\alpha = \pi$. For $\alpha = 0$, the rotation axis is not specified. For $\alpha = \pi$, the rotation axis is not unique and can only be computed up to sign from eq. (2.17): Set without loss of generality $r_x = \pm\sqrt{\frac{1}{2}(R_{1,1}+1)}$ assuming $R_{1,1}$ is the largest element on the matrix diagonal. Then the remaining vector elements can be computed as $r_y = r_x R_{2,3}/R_{1,3}$ and $r_z = r_x R_{2,3}/R_{1,2}$.

Rodrigues' rotation formula[9] is commonly used to rotate a vector $\boldsymbol{v}$ around a rotation axis $\boldsymbol{r}$ by an angle of α:

$$\boldsymbol{v}' = \cos(\alpha)\boldsymbol{v} + \sin(\alpha)(\boldsymbol{r} \times \boldsymbol{v}) + (1 - \cos(\alpha))(\boldsymbol{r}^\mathsf{T}\boldsymbol{v})\boldsymbol{r} \tag{2.20}$$

From eq. (2.20), another transformation from the angle/axis representation

[9] named after the French mathematician Olinde Rodrigues (*1795, †1851)

$(\boldsymbol{r}, \alpha)$ to a corresponding rotation matrix follows:

$$\mathbf{R}_{\boldsymbol{r},\alpha} = \cos(\alpha)\mathbf{I} + \sin(\alpha)[\boldsymbol{r}]_\times + (1 - \cos(\alpha))\boldsymbol{r}\,\boldsymbol{r}^\mathsf{T} \tag{2.21}$$

Using the identity $\boldsymbol{r}\,\boldsymbol{r}^\mathsf{T} = [\boldsymbol{r}]_\times^2 + \mathbf{I}$, eq. (2.21) can also be written as:

$$\mathbf{R}_{\boldsymbol{r},\alpha} = \mathbf{I} + \sin(\alpha)[\boldsymbol{r}]_\times + (1 - \cos(\alpha))[\boldsymbol{r}]_\times^2 \tag{2.22}$$

The geometrical interpretation of this algorithm is to transform a vector $\boldsymbol{v}$ by decomposing it into its components parallel and orthogonal to the rotation axis, i. e., *vector projection* $(\boldsymbol{r}^\mathsf{T}\boldsymbol{v})\boldsymbol{r}$ of $\boldsymbol{v}$ on $\boldsymbol{r}$ and *vector rejection* $\boldsymbol{v} - (\boldsymbol{r}^\mathsf{T}\boldsymbol{v})\boldsymbol{r}$ of $\boldsymbol{v}$ from $\boldsymbol{r}$, and rotating the orthogonal part within the plane orthogonal to $\boldsymbol{r}$.

A severe drawback of the angle/axis representation is the fact that the parameters are difficult to compute for rotation compositions. In general, we have to convert the angle/axis parameters to the corresponding rotation matrices or unit quaternions (see below) and convert the product back to the angle/axis parameters which is cumbersome to keep track of.

Rotation around fixed axis Given that the rotation axis $\boldsymbol{r}$ is fixed, a rotation matrix is described by the rotation angle α only via eq. (2.17). Since $\cos(\alpha)$ and $\sin(\alpha)$ can be described by a unit vector $\boldsymbol{\varrho} \in \mathbb{S}^1$ with $\varrho_1 = \cos(\alpha)$ and $\varrho_2 = \sin(\alpha)$, this provides a linear representation of a rotation around fixed axis $\boldsymbol{r}$ with a single quadratic parameter constraint $\boldsymbol{\varrho}^\mathsf{T}\boldsymbol{\varrho} = 1$. This is useful to parametrize rotations within a known plane.

Rotation vectors There are several minimal parametrizations describing rotations by a 3-vector $\boldsymbol{w}$ which are directly related to the angle/axis representation and Rodrigues' rotation formula (see [BT03; TC04] for a detailed discussion). An obvious choice is the rotation vector (a.k.a. *Euler vector*) $\boldsymbol{w} = \alpha\boldsymbol{r}$, i. e., the length of the vector indicates the rotation angle in radians while the direction is parallel to the rotation axis. Any 3-vector $\boldsymbol{w}$ describes a feasible rotation. For uniqueness, the valid domain is often restricted to the 3-dimensional ball with radius π, denoted as $\mathbb{B}_\pi^2$. Again,

this parametrization has a singularity at 180° since each rotation vector $\boldsymbol{w}$ with $\|\boldsymbol{w}\| = \pi$ represents the same rotation as $-\boldsymbol{w}$.

The transformation from the rotation vector to the corresponding rotation matrix is derived from eq. (2.22), replacing α by $\|\boldsymbol{w}\|$ and $\boldsymbol{r}$ by $\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$:[10]

$$\mathbf{R}_{\boldsymbol{w}} = \mathbf{I} + \sin(\|\boldsymbol{w}\|)[\tfrac{\boldsymbol{w}}{\|\boldsymbol{w}\|}]_\times + (1 - \cos(\|\boldsymbol{w}\|))[\tfrac{\boldsymbol{w}}{\|\boldsymbol{w}\|}]_\times^2 \tag{2.23}$$

The inverse mapping from a rotation matrix $\mathbf{R}$ to the corresponding rotation vector $\boldsymbol{w} = \alpha\boldsymbol{r}$ can be derived from eq. (2.18) and eq. (2.19) computing the rotation angle α and axis $\boldsymbol{r}$.[11]

Other common vectorial rotation representations combining the rotation angle and axis into a single vector are the *reduced Euler-Rodrigues vector* $\boldsymbol{w} = \sin(\frac{\alpha}{2})\boldsymbol{r}$ which are directly related to unit quaternions (see below), the *Cayley-Gibbs-Rodrigues vector*[12] $\boldsymbol{w} = \tan(\frac{\alpha}{2})\boldsymbol{r}$ which cannot represent rotations of $\pm 180°$, or the *Wiener-Milenković vector* $\boldsymbol{w} = \tan(\frac{\alpha}{4})\boldsymbol{r}$ also called *modified Rodrigues vector*.

All these representations have the benefit of being minimal parametrizations for rotations. Nevertheless, they exhibit discontinuities in the parameter space in the vicinity of 180° rotations when they are used to represent spatial orientations. Similar to the angle/axis parametrization, they are also not suitable for describing rotation compositions.

[10] Equation (2.23) describes the *matrix exponential* $\exp([\mathbf{w}]_\times)$ of the antisymmetric 3×3 matrix $[\mathbf{w}]_\times$ describing the cross product with vector $\boldsymbol{w}$.

[11] Equation (2.19) is related to the *matrix logarithm* of $\mathbf{R}$ in analogy to $\mathbf{R} = \exp([\mathbf{w}]_\times)$:

$$[\boldsymbol{w}]_\times = \log(\mathbf{R}) = \begin{cases} \mathbf{0}_{3\times 3} & \text{if } \alpha = 0, \\ \frac{\alpha}{2\sin(\alpha)}(\mathbf{R} - \mathbf{R}^\intercal) & \text{if } \alpha \in (0, \pi) \end{cases}$$

[12] named after the American scientist Josiah Willard Gibbs (*1839, †1903) and the British mathematician Arthur Cayley (*1821, †1895) beside Olinde Rodrigues

Unit Quaternions

Rotations can be described in a compact and only slightly over-parametrized way by *unit quaternions* $\mathbf{q} \in \mathbb{H}$ resp. the vector representation of the unit quaternion coefficients $\mathbf{q} \in \mathbb{R}^4$ known as the *Euler-Rodrigues vector* or *Euler symmetric parameters* (see A.1.1 for a detailed description of quaternions).

The unit quaternion $\mathbf{q} = (\boldsymbol{q}, q)$ representing a rotation by an angle $\alpha \in [0, \pi]$ around an axis given by the unit vector $\boldsymbol{r}$ is defined as:

$$\mathbf{q} = (\sin(\tfrac{\alpha}{2})\boldsymbol{r}, \cos(\tfrac{\alpha}{2})) \tag{2.24}$$

It is easy to verify that $\|\mathbf{q}\| = 1$, i. e., $\mathbf{q}$ is a unit quaternion:

$$\mathbf{q}^\intercal \mathbf{q} = \sin(\tfrac{\alpha}{2})^2 \boldsymbol{r}^\intercal \boldsymbol{r} + \cos(\tfrac{\alpha}{2})^2 = 1$$

Since unit quaternions are composed of $\varrho = 4$ parameters, there is only one constraint to keep track of, which is the quadratic unit length constraint $\mathbf{q}^\intercal \mathbf{q} = 1$. Given an invalid quaternion $\mathbf{q} \in \mathbb{R}^4$, this constraint is far more easy to enforce than the orthonormality constraints of a rotation matrix:

$$\mathrm{proj}_{\mathrm{R}}(\mathbf{q}) = \frac{\mathbf{q}}{\|\mathbf{q}\|} \tag{2.25}$$

Since both $\mathbf{q}$ and $-\mathbf{q}$ describe the same rotation, we further restrict the domain of unit quaternions to the hemisphere $q \geqslant 0$ for sake of uniqueness in the following considerations. Note that the ambiguity between $(\boldsymbol{q}, 0)$ and $(-\boldsymbol{q}, 0)$ remains.

Given a unit quaternion $\mathbf{q} = (\boldsymbol{q}, q)$, a vector $\boldsymbol{v} \in \mathbb{R}^3$ is rotated via left and right multiplication of the corresponding *pure quaternion* $\mathbf{v} = (\boldsymbol{v}, 0)$ with the quaternion $\mathbf{q}$ and its conjugate $\bar{\mathbf{q}} = (-\boldsymbol{q}, q)$:

$$\mathbf{v}' = \mathbf{q} \cdot \mathbf{v} \cdot \bar{\mathbf{q}} \tag{2.26}$$

where quaternion multiplication is defined as:

$$\mathbf{q}_1 \cdot \mathbf{q}_2 = (q_1 \boldsymbol{q}_2 + q_2 \boldsymbol{q}_1 + \boldsymbol{q}_1 \times \boldsymbol{q}_2,\ q_1 q_2 - \boldsymbol{q}_1^\intercal \boldsymbol{q}_2) \tag{2.27}$$

The rotation matrix $\mathbf{R}_\mathbf{q}$ corresponding to the unit quaternion $\mathbf{q}$ is given by the upper left 3×3 submatrix of the matrix product $\mathbf{M}_\mathbf{q}^\ell \mathbf{M}_{\bar{\mathbf{q}}}^r$ related to left and right quaternion multiplication with $\mathbf{q}$ and $\bar{\mathbf{q}}$:

$$\mathbf{M}_\mathbf{q}^\ell = \begin{pmatrix} q\mathbf{I} + [\boldsymbol{q}]_\times & \boldsymbol{q} \\ -\boldsymbol{q}^\mathsf{T} & q \end{pmatrix} = \begin{pmatrix} q & -q_z & q_y & q_x \\ q_z & q & -q_x & q_y \\ -q_y & q_x & q & q_z \\ -q_x & -q_y & -q_z & q \end{pmatrix} \tag{2.28}$$

and

$$\mathbf{M}_{\bar{\mathbf{q}}}^r = \begin{pmatrix} q\mathbf{I} + [\boldsymbol{q}]_\times & -\boldsymbol{q} \\ \boldsymbol{q}^\mathsf{T} & q \end{pmatrix} = \begin{pmatrix} q & -q_z & q_y & -q_x \\ q_z & q & -q_x & -q_y \\ -q_y & q_x & q & -q_z \\ q_x & q_y & q_z & q \end{pmatrix} \tag{2.29}$$

resulting in:

$$\mathbf{R}_\mathbf{q} = q^2\mathbf{I} + 2q[\boldsymbol{q}]_\times + [\boldsymbol{q}]_\times^2 + \boldsymbol{q}\boldsymbol{q}^\mathsf{T} =$$

$$\begin{pmatrix} q_x^2 - q_y^2 - q_z^2 + q^2 & 2(q_x q_y - q_z q) & 2(q_x q_z + q_y q) \\ 2(q_x q_y + q_z q) & -q_x^2 + q_y^2 - q_z^2 + q^2 & 2(q_y q_z - q_x q) \\ 2(q_x q_z - q_y q) & 2(q_x q + q_y q_z) & -q_x^2 - q_y^2 + q_z^2 + q^2 \end{pmatrix} \tag{2.30}$$

or in closer resemblance to Rodrigues' rotation formula (2.22):

$$\mathbf{R}_\mathbf{q} = (q^2 + 1)\mathbf{I} + 2q[\boldsymbol{q}]_\times + 2[\boldsymbol{q}]_\times^2$$

It is noteworthy that a similarity transformation can be representated with minimal number of parameters by using the quaternion for rotation representation and dropping the unit length constraint. Inserting an arbitrary length quaternion $\mathbf{q}$ into *eq.* (2.26) instead of a unit length quaternion results in an additional scaling by $\lambda = \|\mathbf{q}\|^2$.

The composition of rotations represented by quaternions $\mathbf{q}_1$ and $\mathbf{q}_2$ is given by the quaternion product $\mathbf{q}_1 \cdot \mathbf{q}_2$ which provides a computational

benefit over other angle/axis based parametrizations:

$$\mathbf{v}' = \mathbf{q}_1 \cdot (\mathbf{q}_2 \cdot \mathbf{v} \cdot \bar{\mathbf{q}}_2) \cdot \bar{\mathbf{q}}_1 = (\underbrace{\mathbf{q}_1 \cdot \mathbf{q}_2}_{\mathbf{q}}) \cdot \mathbf{v} \cdot (\underbrace{\bar{\mathbf{q}}_2 \cdot \bar{\mathbf{q}}_1}_{\bar{\mathbf{q}}}) \tag{2.31}$$

and

$$\mathrm{comp}_{\mathrm{R}}(\mathbf{q}_1, \mathbf{q}_2) = \mathbf{q}_1 \cdot \mathbf{q}_2 = \mathbf{M}^{\ell}_{\mathbf{q}_1} \mathbf{q}_2 = \mathbf{M}^{r}_{\mathbf{q}_2} \mathbf{q}_1 \tag{2.32}$$

The inverse rotation is simply given by the conjugate quaternion:

$$\mathrm{inv}_{\mathrm{R}}(\mathbf{q}) = \bar{\mathbf{q}} = (-\boldsymbol{q}, q) = \begin{pmatrix} -\mathbf{I} & \mathbf{0} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{pmatrix} \mathbf{q} \tag{2.33}$$

From eq. (2.24) we can easily derive the conversion from unit quaternions to the angle/axis representation:

$$\boldsymbol{r} = \frac{\boldsymbol{q}}{\|\boldsymbol{q}\|} \quad \text{and} \quad \alpha = 2 \arccos(q) \tag{2.34}$$

For $q = 0$, the rotation axis is not defined and can be set arbitrarily. The conversion from unit quaternion to rotation matrix is described in eq. (2.30). Converting a rotation matrix $\mathbf{R}$ into the corresponding unit quaternion is done by first extracting the rotation angle and axis from $\mathbf{R}$ via eq. (2.18) and eq. (2.19) and deriving the unit quaternion as defined in eq. (2.24). There are also numerically stable closed-form approaches for conversion considering different conditions of the rotation matrix (see [Ter+12]).

Reduced parametrization Unit quaternions are parametrized as 4-vectors $\mathbf{q} \in \mathbb{R}^4$ with one quadratic constraint $\mathbf{q}^{\mathsf{T}}\mathbf{q} = 1$. However, unconstrained parameters are often needed for nonlinear optimization. There are several approaches to represent unit quaternions with a minimal parameter vector $\boldsymbol{\varrho} \in \mathbb{R}^3$, often in the vicinity of some reference unit quaternion $\mathbf{p} \in \mathbb{R}^4$, based on homeomorphisms from $\mathbb{R}^3$ to $\mathbb{S}^3$. Discussions can be found in [Iki00], [SN01] and more recently in [Ter+12]. For further details we refer the reader to A.1.1.

In our work, a modified version of the mapping $\mathcal{Q} : \mathbb{R}^3 \to \mathbb{S}^3$ from [Ter+12]

is used:

$$\mathcal{Q}(\boldsymbol{\varrho}) = (2\boldsymbol{\varrho}, 1 - \boldsymbol{\varrho}^\mathsf{T}\boldsymbol{\varrho})/(\boldsymbol{\varrho}^\mathsf{T}\boldsymbol{\varrho} + 1) \quad \text{and} \quad \boldsymbol{\varrho} = \mathcal{Q}^{-1}(\mathbf{q}) = \tfrac{q}{q+1} \tag{2.35}$$

Given a reference unit quaternion $\mathbf{p}$, the minimal parametrization of a unit quaternion $\mathbf{q}$ is given by $\mathcal{Q}_\mathbf{p}^{-1}(\mathbf{q}) = \mathcal{Q}^{-1}(\bar{\mathbf{p}} \cdot \mathbf{q})$. The unit quaternion for a parameter vector $\boldsymbol{\varrho}$ is given by $\mathcal{Q}_\mathbf{p}(\boldsymbol{\varrho}) = \mathbf{p} \cdot \mathcal{Q}(\mathbf{q})$.

The lifting from minimal rotation parameters $\boldsymbol{\varrho} \in \mathbb{R}^3$ to unit quaternions $\mathbf{q} \in \mathbb{S}^3$ for a reference quaternion $\mathbf{p}$ is denoted by a mapping $\mathcal{Q}_\mathbf{p} : \mathbb{R}^3 \to \mathbb{S}^3$ in the remainder of this work. This mapping is considered to be bijective at least with respect to the hemisphere of unit quaternions $(\boldsymbol{q}, q)$ with $q \geqslant 0$. Note that virtually the same technique can be used to describe unit 3d vectors $\boldsymbol{v} \in \mathbb{S}^2$ within a given reference hemisphere.

2.3.3 Rigid Motion Parametrization

Screw motion In kinematics, *Chasles' theorem*[13] states that any rigid motion in 3d space can be accomplished in terms of a rotation around a unique 3d line – called the *screw axis* – and a translation along the same line. Describing motions this way is known as *screw motion* [Che91].

The screw axis can be described by a 3d line $\boldsymbol{L}(t) = \boldsymbol{c} + t\boldsymbol{r}$ with direction vector $\boldsymbol{r} \in \mathbb{S}^2$ and position $\boldsymbol{c} \in \mathbb{R}^3$. For sake of uniqueness we define $\boldsymbol{c}^\mathsf{T}\boldsymbol{r} = 0$. Given the magnitude of translation along the screw axis $p \in \mathbb{R}$ and the rotation angle α, motion of a 3d point $\boldsymbol{X}$ is given by:

$$\boldsymbol{X}' = \mathbf{R}_{\boldsymbol{r},\alpha}(\boldsymbol{X} - \boldsymbol{c}) + \boldsymbol{c} + p\boldsymbol{r} = \mathbf{R}_{\boldsymbol{r},\alpha}\boldsymbol{X} + (\mathbf{I} - \mathbf{R}_{\boldsymbol{r},\alpha})\boldsymbol{c} + p\boldsymbol{r} \tag{2.36}$$

Using this formulation, the translation $\boldsymbol{t} = \boldsymbol{t}_\perp + \boldsymbol{t}_\parallel$ is separated into a part $\boldsymbol{t}_\perp = (\mathbf{I} - \mathbf{R}_{\boldsymbol{r},\alpha})\boldsymbol{c}$ orthogonal to the rotation axis and a part $\boldsymbol{t}_\parallel = p\boldsymbol{r}$ parallel to the rotation axis.

An alternative representation uses the direction/moment parametrization $(\boldsymbol{r}, \boldsymbol{m})$ – also known as *Plücker line coordinates* – for the screw axis instead

[13] originally published by the French mathematician Michel Chasles in 1830

where the moment vector is given by $\boldsymbol{m} = \boldsymbol{c} \times \boldsymbol{r}$. However, both representations have the same disadvantages already discussed for the angle/axis representation of rotations.

Dual quaternions A convenient formulation of screw motion resembling the quaternion representation for rotations is provided by the use of *dual quaternions* (see A.1.2 for a detailed description of dual quaternions).

The dual quaternion $\check{\mathbf{q}} = (\mathbf{q}, \mathbf{p})$ related to a spatial displacement is computed from the screw axis $\check{\boldsymbol{r}} = (\boldsymbol{r}, \boldsymbol{m})$ and the dual angle $\check{\alpha} = (\alpha, d)$ where α is the rotation angle about the axis and d is the magnitude of translation along the axis, also referred to as *pitch*. The dual quaternion is given by:

$$\check{\mathbf{q}} = (\sin(\tfrac{\check{\alpha}}{2})\check{\boldsymbol{r}}, \cos(\tfrac{\check{\alpha}}{2})) \tag{2.37}$$

i. e., the real part $\mathbf{q}$ is given by eq. (2.24) and the dual part is:

$$\mathbf{p} = (\tfrac{d}{2}\cos(\tfrac{\alpha}{2})\boldsymbol{r} + \sin(\tfrac{\alpha}{2})\boldsymbol{m}, -\tfrac{d}{2}\sin(\tfrac{\alpha}{2})) \tag{2.38}$$

Note that the dual quaternion satisfies $\mathbf{q}^\mathsf{T}\mathbf{q} = 1$ and $\mathbf{q}^\mathsf{T}\mathbf{p} = 0$, hence it is a *unit dual quaternion*:

$$\begin{aligned}\mathbf{q}^\mathsf{T}\mathbf{p} &= \sin(\tfrac{\alpha}{2})\boldsymbol{r}^\mathsf{T}(\tfrac{d}{2}\cos(\tfrac{\alpha}{2})\boldsymbol{r} + \sin(\tfrac{\alpha}{2})\boldsymbol{m}) - \cos(\tfrac{\alpha}{2})\tfrac{d}{2}\sin(\tfrac{\alpha}{2}) \\ &= \sin(\tfrac{\alpha}{2})\tfrac{d}{2}\cos(\tfrac{\alpha}{2}) - \cos(\tfrac{\alpha}{2})\tfrac{d}{2}\sin(\tfrac{\alpha}{2}) = 0\end{aligned}$$

since $\boldsymbol{r}^\mathsf{T}\boldsymbol{r} = 1$ and $\boldsymbol{r}^\mathsf{T}\boldsymbol{m} = 0$.

Given this unit dual quaternion, a 3d line $\check{\boldsymbol{l}} = (\boldsymbol{l}, \boldsymbol{m})$ is rotated via left and right multiplication with $\check{\mathbf{q}}$ and its conjugate $\bar{\check{\mathbf{q}}} = (\bar{\mathbf{q}}, \bar{\mathbf{p}})$:[14]

$$\check{\mathbf{l}}' = \check{\mathbf{q}} \cdot \check{\mathbf{l}} \cdot \bar{\check{\mathbf{q}}} = (\mathbf{q} \cdot \mathbf{l} \cdot \bar{\mathbf{q}},\ \mathbf{q} \cdot \mathbf{m} \cdot \bar{\mathbf{q}} + \mathbf{p} \cdot \mathbf{l} \cdot \bar{\mathbf{q}} + \mathbf{q} \cdot \mathbf{l} \cdot \bar{\mathbf{p}}) \tag{2.39}$$

according to the definition of dual quaternion multiplication:

$$\check{\mathbf{q}}_1 \cdot \check{\mathbf{q}}_2 = (\mathbf{q}_1 \cdot \mathbf{q}_2,\ \mathbf{q}_1 \cdot \mathbf{p}_2 + \mathbf{p}_1 \cdot \mathbf{q}_2) \tag{2.40}$$

[14] As in eq. (2.26), the dual 3-vector $\check{\boldsymbol{l}}$ is considered as a dual pure quaternion $\check{\mathbf{l}}$ here.

This formula resembles eq. (2.26) describing rotation of 3d vectors using unit quaternions.

Representing a 3d point $\boldsymbol{X} \in \mathbb{R}^3$ by a dual quaternion $\check{\mathbf{X}} = (\mathbf{1}, \mathbf{X})$ with pure dual part $\mathbf{X} = (\boldsymbol{X}, 0)$ and real part $\mathbf{1} = (\mathbf{0}, 1)$, the transformation can be computed in a similar way to eq. (2.39), using the *mixed dual conjugate* $\bar{\check{\mathbf{q}}}^* = (\bar{\mathbf{q}}, -\bar{\mathbf{p}})$ instead of $\bar{\check{\mathbf{q}}}$ as described in [Bay03]:

$$\check{\mathbf{X}}' = \check{\mathbf{q}} \cdot \check{\mathbf{X}} \cdot \bar{\check{\mathbf{q}}}^* = (\mathbf{q} \cdot \bar{\mathbf{q}},\ \mathbf{q} \cdot \mathbf{X} \cdot \bar{\mathbf{q}} + \mathbf{p} \cdot \bar{\mathbf{q}} - \mathbf{q} \cdot \bar{\mathbf{p}}) \tag{2.41}$$

From the unit length constraint we can derive that $\mathbf{p} \cdot \bar{\mathbf{q}}$ and $\mathbf{q} \cdot \bar{\mathbf{p}}$ are pure quaternions and $\mathbf{p} \cdot \bar{\mathbf{q}} = -\mathbf{q} \cdot \bar{\mathbf{p}}$ (see eq. (A.6)). Hence, the result of eq. (2.41) is also a dual quaternion with real part $\mathbf{1}$ and pure dual part representing the transformed 3d point. Note that the dual part of eq. (2.41) describes in fact the rotation of $\boldsymbol{X}$ with $\mathbf{R}_\mathbf{q}$ and additional translation by the vector part of $\mathbf{p} \cdot \bar{\mathbf{q}} - \mathbf{q} \cdot \bar{\mathbf{p}}$.

In both cases, the translational part of the motion is given by:

$$\mathbf{t} = 2\mathbf{p} \cdot \bar{\mathbf{q}} \quad \text{i. e., } \boldsymbol{t} = 2(q\boldsymbol{p} - p\boldsymbol{q} + \boldsymbol{q} \times \boldsymbol{p}) \tag{2.42}$$

Hence, given a unit quaternion $\mathbf{q}$ representing rotation and a translation vector $\boldsymbol{t}$, the unit dual quaternion representing the spatial displacement can be derived directly as:

$$\check{\mathbf{q}} = (\mathbf{q}, \tfrac{1}{2}\mathbf{t} \cdot \mathbf{q}) \quad \text{i. e., } \mathbf{p} = \tfrac{1}{2}(\boldsymbol{t} \times \boldsymbol{q} + q\boldsymbol{t}, -\boldsymbol{t}^\mathsf{T}\boldsymbol{q}) \tag{2.43}$$

Composition and inversion is analogous to unit quaternions for rotation representation. The composition of motions described by dual quaternions $\check{\mathbf{q}}_1$, $\check{\mathbf{q}}_2$ is given by the dual quaternion product $\check{\mathbf{q}}_1 \cdot \check{\mathbf{q}}_2$ similar to eq. (2.32). The inverse transformation for $\check{\mathbf{q}}$ is given by the conjugate dual quaternion $\bar{\check{\mathbf{q}}}$ similar to eq. (2.33).

Since dual quaternions are composed of $\varrho = 8$ parameters, there are two parameter constraints resulting from the unit length constraint $\|\check{\mathbf{q}}\| = 1$, i. e., the real unit length constraint $\mathbf{q}^\mathsf{T}\mathbf{q} = 1$ and the real-dual orthogonality constraint $\mathbf{q}^\mathsf{T}\mathbf{p} = 0$. Given an invalid dual quaternion, these constraints

are enforced by:

$$\text{proj}_{\text{R}}(\mathbf{q}, \mathbf{p}) = \frac{1}{\|\mathbf{q}\|}\left(\mathbf{q},\ \mathbf{p} - \frac{\mathbf{p}^\intercal\mathbf{q}}{\mathbf{q}^\intercal\mathbf{q}}\mathbf{q}\right) \tag{2.44}$$

2.4 Rigid Motion Metrics

The most common way to measure distances between elements of the special Euclidean group $\text{SE}(3)$ is to combine measures for the rotational part and the translational part. While the Euclidean distance is in general used for translation vectors, comparison of rotations is not as straightforward.

Rotation metrics The following metrics are commonly used to measure distances between elements of the rotation group $\text{SO}(3)$. The actual computation of the metrics depends on the parametrization of the rotations [Huy09; Dai+10].

The *geodesic* or *angle metric* $d_\angle$ is the rotation angle α of the residual rotation $\mathbf{R}^\intercal\mathbf{R}'$ between rotations $\mathbf{R}$ and $\mathbf{R}'$. The range of the metric is $[0, \pi]$. For rotation matrices it can be computed as:

$$d_\angle(\mathbf{R}, \mathbf{R}') = \arccos(\tfrac{1}{2}(\text{trace}(\mathbf{R}^\intercal\mathbf{R}') - 1)) = \alpha \tag{2.45}$$

For unit quaternions $\mathbf{q}$, $\mathbf{q}'$ it can be computed either from the scalar part of the unit quaternion $\bar{\mathbf{q}} \cdot \mathbf{q}'$ representing the residual rotation or from the magnitude of its vector part:[15]

$$d_\angle(\mathbf{q}, \mathbf{q}') = 2\arccos(|\mathbf{q}^\intercal\mathbf{q}'|) = 2\arcsin(\|q\boldsymbol{q}' - q'\boldsymbol{q} - \boldsymbol{q} \times \boldsymbol{q}'\|) \tag{2.46}$$

The *chordal metric* $d_{\text{chord}}(\mathbf{R}, \mathbf{R}')$ measures the norm of the matrix difference:

$$\text{d}_{\text{chord}}(\mathbf{R}, \mathbf{R}') = \|\mathbf{R} - \mathbf{R}'\| = \|\mathbf{R}^\intercal\mathbf{R}' - \mathbf{I}\| = 2\sqrt{2}\sin(\alpha/2) \tag{2.47}$$

where $\|\cdot\|$ represents the Frobenius norm. Its range is $[0, 2\sqrt{2}]$.

[15] A numerically more stable computation is $2\,\text{atan2}(\|q\boldsymbol{q}' - q'\boldsymbol{q} - \boldsymbol{q} \times \boldsymbol{q}'\|, |\mathbf{q}^\intercal\mathbf{q}'|)$.

The *quaternion metric* $d_{\text{quat}}(\mathbf{q}, \mathbf{q}')$ measures the difference between the unit quaternion representations of $\mathbf{R}$ and $\mathbf{R}'$:

$$d_{\text{quat}}(\mathbf{q}, \mathbf{q}') = \min(\|\mathbf{q} - \mathbf{q}'\|, \|\mathbf{q} + \mathbf{q}'\|) = 2\sin(\alpha/4) \tag{2.48}$$

since both $\mathbf{q}$ and $-\mathbf{q}$ represent the same rotation.[16] Its range is $[0, \sqrt{2}]$. Note that $\|\mathbf{q} \pm \mathbf{q}'\| = \sqrt{2 \pm 2\mathbf{q}^\intercal\mathbf{q}'}$ for unit quaternions. Hence, the quaternion metric can also be written as $d_{\text{quat}}(\mathbf{q}, \mathbf{q}') = \sqrt{2 - 2|\mathbf{q}^\intercal\mathbf{q}'|}$.

Since the sine can be approximated as $\sin(\varphi) \approx \varphi$ for small angles φ, scaled versions of the chordal metric and quaternion metric can be used to approximate the angle metric: $\frac{1}{\sqrt{2}}\, d_{\text{chord}}(\mathbf{R}, \mathbf{R}') \approx \alpha$ and $2\, d_{\text{quat}}(\mathbf{q}, \mathbf{q}') \approx \alpha$.

2.5 Modeling Rigid Motion Uncertainty

In practical applications, rigid motion measurements are not ideal but afflicted with measurement errors resulting from the actual pose estimation process. A common way to model uncertainties of rigid motion parameters is to describe measurement errors as additive values following a Gaussian distribution with zero mean:

$$\boldsymbol{\mu} = \boldsymbol{\mu}^* + \boldsymbol{\mu}_\varepsilon \quad \text{with } \boldsymbol{\mu}_\varepsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\mu}_\varepsilon}) \tag{2.49}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{\mu}_\varepsilon} \in \mathbb{R}^{\mu \times \mu}$ is the *covariance matrix* of $\boldsymbol{\mu}_\varepsilon \in \mathbb{R}^\mu$, $\boldsymbol{\mu}$ are the measured parameters and $\boldsymbol{\mu}^*$ are the original parameters denoted as *ground truth* parameters.

A natural distance measurement between uncertain measurements $\boldsymbol{\mu}$ and predictions $\hat{\boldsymbol{\mu}}$ is given by the *Mahalanobis distance*[17]

$$d_{\text{Maha}}(\boldsymbol{\mu}, \hat{\boldsymbol{\mu}}) = \|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\|_{\boldsymbol{\Sigma}_{\boldsymbol{\mu}_\varepsilon}} = \sqrt{(\boldsymbol{\mu} - \hat{\boldsymbol{\mu}})^\intercal \boldsymbol{\Sigma}_{\boldsymbol{\mu}_\varepsilon}^{-1} (\boldsymbol{\mu} - \hat{\boldsymbol{\mu}})} \tag{2.50}$$

[16] The distinction between $\mathbf{q}$ and $-\mathbf{q}$ can be omitted when the unit quaternion are restricted to the same hemisphere of $\mathbb{S}^3$, for example by $q \geqslant 0$.

[17] named after the Indian scientist and applied statistician Prasanta Chandra Mahalanobis (*1893, †1972)

i. e., the Euclidean distance weighted by the inverse covariance matrix to take different uncertainties in the parameters into account. Depending on the rigid motion parametrization, $\mathbf{\Sigma}_{\boldsymbol{\mu}_\varepsilon}$ can be singular. This occurs when $\boldsymbol{\mu}$ contains parameters with internal constraints such as unit quaternions. In this case, the matrix inverse $\mathbf{\Sigma}_{\boldsymbol{\mu}_\varepsilon}^{-1}$ is replaced by the pseudoinverse $\mathbf{\Sigma}_{\boldsymbol{\mu}_\varepsilon}^{\dagger}$ in eq. (2.50) as explained in A.4.

For the parametrization of rigid motions with unit quaternion $\mathbf{q} \in \mathbb{S}^3$ and translation vector $\boldsymbol{t} \in \mathbb{R}^3$, the covariance matrix has the form:

$$\mathbf{\Sigma}_{\boldsymbol{\mu}_\varepsilon} = \begin{pmatrix} \mathbf{\Sigma}_{\mathbf{q}_\varepsilon} & \mathbf{\Sigma}_{\mathbf{q}_\varepsilon,\boldsymbol{t}_\varepsilon} \\ \mathbf{\Sigma}_{\mathbf{q}_\varepsilon,\boldsymbol{t}_\varepsilon}^{\mathsf{T}} & \mathbf{\Sigma}_{\boldsymbol{t}_\varepsilon} \end{pmatrix} \tag{2.51}$$

where $\mathbf{\Sigma}_{\mathbf{q}_\varepsilon,\boldsymbol{t}_\varepsilon} \in \mathbb{R}^{4\times 3}$ describes the correlation between rotation and translation parameter errors and $\mathbf{\Sigma}_{\mathbf{q}_\varepsilon} \in \mathbb{R}^{4\times 4}$, $\mathbf{\Sigma}_{\boldsymbol{t}_\varepsilon} \in \mathbb{R}^{3\times 3}$ describe the covariance of $\mathbf{q}_\varepsilon$ and $\boldsymbol{t}_\varepsilon$ respectively. Note that $\mathbf{\Sigma}_{\mathbf{q}_\varepsilon}$ is singular with $\operatorname{rank}(\mathbf{\Sigma}_{\mathbf{q}_\varepsilon}) = 3$ due to the unit length constraint.

In the simplified model we assume equally distributed uncorrelated translation errors, i. e., $\mathbf{\Sigma}_{\boldsymbol{t}_\varepsilon} = \sigma_{\boldsymbol{t}_\varepsilon}^2 \mathbf{I}$ with standard deviation $\sigma_{\boldsymbol{t}_\varepsilon}$, and rotation errors are described by $\mathbf{R}_{\boldsymbol{r}_\varepsilon,\alpha_\varepsilon}$ with rotation axes $\boldsymbol{r}_\varepsilon$ uniformly distributed over $\mathbb{S}^2$ and rotation angle $\alpha_\varepsilon \sim \mathcal{N}(0, \sigma_{\alpha_\varepsilon}^2)$ with standard deviation $\sigma_{\alpha_\varepsilon}$.

For details on propagation of uncertainty subject to nonlinear transformation of rigid motion parameters, e. g., from angle/axis to unit quaternion, we refer the reader to A.5.

2.6 Related Problems

2.6.1 Absolute Pose Alignment

In stereo calibration or multi-camera calibration with overlapping views the relative pose between rigidly coupled camera is estimated from 2d/3d correspondences of some calibration object with known geometry that is visible in multiple camera images at the same time.

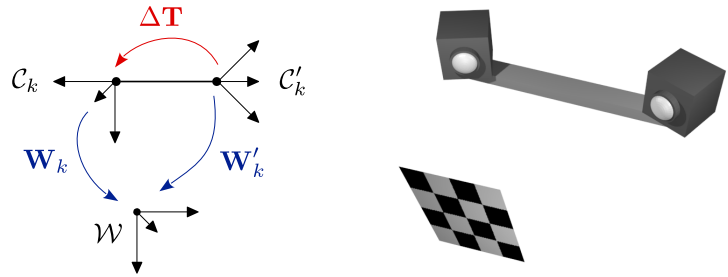

Figure 2.4. Extrinsic stereo camera calibration from absolute poses with respect to the same calibration object.

From a geometrical viewpoint, this problem can be formally described in terms of the coordinate frames and transformations introduced in Sec. 2.2 as computing the Euclidean transformations $\Delta\mathbf{T}$ between two rigidly coupled cameras given m absolute poses $\mathbf{W}_k, \mathbf{W}'_k$, $k = 1, \ldots, m$ as depicted in Fig. 2.4. $\mathbf{W}_k$ and $\mathbf{W}'_k$ can be interpreted as poses of the moving cameras within the world coordinate frame which is associated with the static calibration object, or vice versa as inverse pose transformations of the moving calibration object within the camera coordinate frame. Since both cameras relate to the same object, we have $\mathcal{W} = \mathcal{W}'$, i. e., $\Delta\mathbf{W} = [\mathbf{I} \mid \mathbf{0}]$. Thus, the problem of extrinsic multi-camera calibration from absolute poses is reduced to finding the Euclidean transformation $\Delta\mathbf{T} = (\mathbf{W}_k)^{-1}\mathbf{W}'_k$, e. g., via pose averaging.

However, this approach cannot be applied if the coupled cameras have minimal or no overlapping field of view as in our case.

2.6.2 Hand-Eye and Base-World Calibration

Hand-eye and simultaneous hand-eye and base-world calibration can be defined in terms of the coordinate frames and transformations introduced in Sec. 2.2 as a special case of eye-to-eye calibration. Here, the number

of sensors to align is in general limited to two where the first coordinate frame $\mathcal{C}$ is associated with the robot's arm ("hand") and the second $\mathcal{C}'$ with the camera mounted onto the arm ("eye") or vice versa. Absolute poses $\mathbf{W}_k$ of the hand are computed from the joint configuration and angles of the robot's arm with respect to the "base" coordinate frame $\mathcal{W}$ (e. g., the coordinate frame associated with the base joint of the arm in normal position or with the robot's body) while absolute poses $\mathbf{W}'_k$ of the camera are computed from images of some calibration object such as a checkerboard pattern, defining the "world" coordinate frame $\mathcal{W}'$. In this scenario, relative poses are typically derived from pairs of absolute poses, i. e., $\mathbf{T}_{k,\ell} = (\mathbf{W}_k)^{-1}\mathbf{W}_\ell$ and $\mathbf{T}'_{k,\ell} = (\mathbf{W}'_k)^{-1}\mathbf{W}'_\ell$ for $1 \leqslant k, \ell \leqslant m$, $k \neq \ell$. The relationship between these pose transformations is illustrated in Fig. 2.5 and Fig. 2.6.

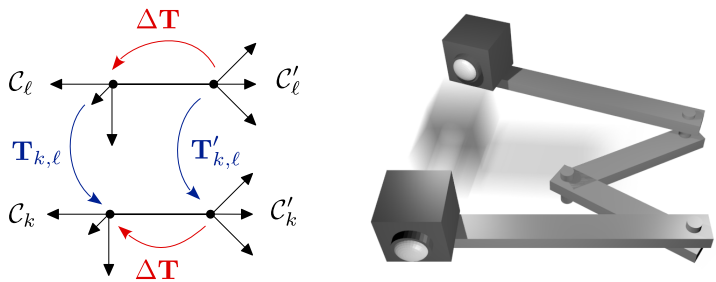

Figure 2.5. Hand-eye calibration from relative poses of robot arm ("hand") and camera ("eye").

Computation of the Euclidean transformation $\Delta\mathbf{T}$ given relative poses $\mathbf{T}_{k,\ell}, \mathbf{T}'_{k,\ell}$ is known as *hand-eye calibration*, sometimes also referred to as *tool-flange calibration*. If the absolute scale of the camera translations $\boldsymbol{t}'_{k,\ell}$ is unknown, $\Delta\mathbf{T}$ becomes a similarity transformation $\Delta\mathbf{S}$ and the problem is referred to as *extended hand-eye calibration*. This case occurs when relative camera poses are not computed from some calibration object but rather based on Structure from Motion which can recover camera ego-motion only up to scale.

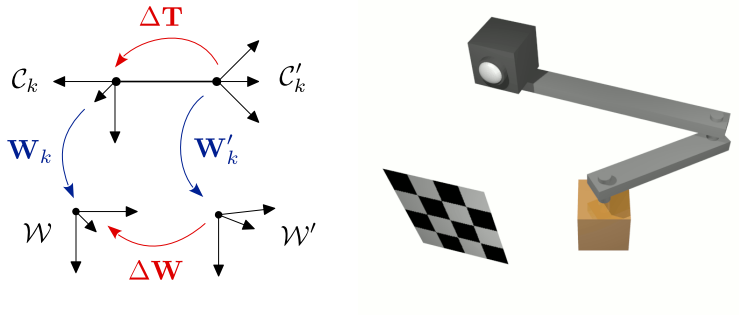

Figure 2.6. Simultaneous hand-eye and base-world calibration from absolute poses of robot arm and camera with respect to robot body ("base") and checkerboard coordinate frame ("world").

Computation of the Euclidean transformation $\Delta\mathbf{W}$ between the robot's base and the calibration object given absolute poses $\mathbf{W}_k, \mathbf{W}'_k$ is known as *base-world* or *robot-world calibration*. If both $\Delta\mathbf{T}$ and $\Delta\mathbf{W}$ are unknown, the problem is referred to as *simultaneous hand-eye and base-world calibration*.

Since the pioneering work by Tsai & Lenz [TL89], hand-eye and base-world calibration is in general derived from identities of pose transformations between relative positions of the hand and eye. As depicted in Fig. 2.5, for each relative pose the following equation holds:

$$\mathbf{T}_{k,\ell}\Delta\mathbf{T} = \Delta\mathbf{T}\mathbf{T}'_{k,\ell} \quad \forall\, k, \ell \tag{2.52}$$

This equation (also referred to as $\mathbf{AX} = \mathbf{XB}$ in the literature) is the foundation of modern hand-eye calibration methods. A similar equation (often referred to as $\mathbf{AX} = \mathbf{ZB}$ or likewise) can be derived for simultaneous hand-eye and base-world calibration:

$$\mathbf{W}_k\Delta\mathbf{T} = \Delta\mathbf{W}\mathbf{W}'_k \quad \forall\, k \tag{2.53}$$

However, this problem is often reduced to an instance of the hand-eye calibration problem by considering relative poses and eliminating the base-

world transformation as described above.[18] We will discuss the hand-eye calibration problem, its properties, and numerical solutions in detail in the next chapter.

2.7 Summary

In this chapter we described basic coordinate transformations such as Euclidean transformation, similarity transformation, rotation, and translatio, and we discussed the advantages and disadvantages of different parametrizations and metrics that are suitable for numerical estimation. For rotation parametrization, unit quaternions were considered as a reasonable compromise between minimal parametrization, simple constraints, and simple applicability.[19] Unit dual quaternion provide similar benefits for parametrization of rigid motion although their internal constraints are more demanding.

We introduced the basic structure of nested coordinate frames and coordinate transformations to describe rigid motion of a multi-camera system and fixed the nomenclature for recurring entities.

Finally, we introduced the classical hand-eye calibration problem in terms of the proposed coordinate transformations. Although hand-eye calibration was originally developed for a specific application – a camera mounted onto a robotic arm – the same approach can be used in general for any rigidly coupled pose measurement devices, i. e., devices that measure or estimate their position and orientation with respect to a fixed coordinate frame, as first mentioned by Ikits [Iki00]. In this work we will concentrate on rigidly coupled monocular cameras without overlapping views. Camera poses will be retrieved from analyzing the associated camera images

[18] Note also that this problem is reduced to the case of stereo calibration as described in Sec. 2.6.1 if either the hand-eye transformation or the base-world transformation is known as first noted by Wang et al. [Wan92].

[19] As Altmann states in [Alt86]: "Anyone who has ever used any other parametrization of the rotation group will, within hours of taking up the quaternion parametrization, lament his or her misspent youth."

only. We do not assume that additional pose measurement devices such as inertial measurement units or odometric sensors are present.

To reflect the adaption of hand-eye calibration methods to the case of rigidly coupled cameras, we will replace the terms “hand-eye” and “base-world” by “eye-to-eye” and “world-to-world” respectively. In the following, hand-eye calibration is replaced by the eponymous term *eye-to-eye calibration*. Simultaneous eye-to-eye calibration and world-to-world calibration is briefly descibed by *eye-and-world calibration*.

Chapter 3

Extrinsic Calibration From Relative Poses

In Sec. 2.6.2, we introduced hand-eye calibration for recovering the relative pose between two rigidly linked pose measuring systems from simultaneously captured relative motions. Abstracting from the actual pose estimation process, basically the same approach can be applied to estimate the relative poses between cameras of a multi-camera rig, i. e., *eye-to-eye calibration*. Since this process considers only geometrical measures between 3d motions to model the estimation error, we refer to these methods as *geometric eye-to-eye calibration*.[1]

In this chapter, we will present existing methods for hand-eye calibration, discuss their applicability for eye-to-eye calibration, and describe how to modify them properly for this purpose. Critical motion configurations are discussed and an analysis of the achievable estimation accuracy based on synthetic data is presented.

3.1 Problem Statement

In the following, we consider a camera system consisting of n rigidly coupled cameras. For simplicity, we will only consider a pair of cameras first and extend the problem to simultaneous calibration of all cameras later. Given are m corresponding pose transformations $\mathbf{T}_k$ and $\mathbf{T}'_k$, $k = 1, \ldots, m$

[1] [EWK07] S. Esquivel, F. Woelk, R. Koch: "Calibration of a multi-camera rig from non-overlapping views", Proc. of DAGM'07, 09/2007

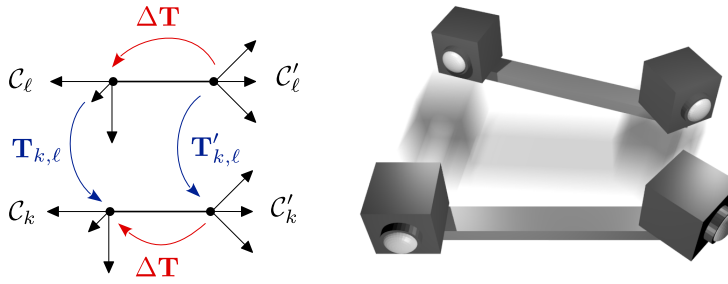

Figure 3.1. Local coordinate frames and relative coordinate transformations of two rigidly coupled cameras under motion.

for the first and second camera with respect to their reference coordinate frames $\mathcal{C}, \mathcal{C}'$ associated with a reference pose of the camera system resp. relative poses $\mathbf{T}_{k,\ell}$ and $\mathbf{T}'_{k,\ell}$ as defined in Sec. 2.2. These transformations are supposed to have been measured in a previous pose estimation process. The coordinate transformation $\Delta\mathbf{T}$ relating $\mathcal{C}'$ to $\mathcal{C}$, i. e., the *eye-to-eye transformation*, is considered as fixed for all motions.[2] The relationship between these pose transformations is illustrated in Fig. 3.1.

Each pose transformation consists of a rotation and a translation described by homogeneous transformation matrices $\mathbf{T} = [\mathbf{R} \mid \boldsymbol{t}]$ in the following. Rotations are described by rotation matrices $\mathbf{R}$ and translations by vectors $\boldsymbol{t}$. This notation is useful to facilitate understanding of the equations and to analyze degenerate cases. Nonetheless, we will also consider different parametrizations for rotations resp. pose transformations such as real and dual quaternions later.

Eye-to-eye calibration problem Measurements $\boldsymbol{X}'$ within the coordinate frame $\mathcal{C}'$ of the second camera are transferred into the coordinate frame $\mathcal{C}$ of the first camera via the eye-to-eye transformation $\Delta\mathbf{T}$ and vice versa as

[2] replacing the term "hand-eye transformation" used in the classical approach

follows:

$$\boldsymbol{X} = \Delta\mathbf{T}\boldsymbol{X}' \quad \text{i. e., } \boldsymbol{X} = \Delta\mathbf{R}\boldsymbol{X}' + \Delta\boldsymbol{t} \tag{3.1}$$

Given relative poses $\mathbf{T}_{k,\ell}$ of the first camera within $\mathcal{C}$, the corresponding relative poses $\mathbf{T}'_{k,\ell}$ of the second camera within $\mathcal{C}'$ are deduced by changing the reference coordinate frame via $\Delta\mathbf{T}$:

$$\mathbf{T}'_{k,\ell} = \Delta\mathbf{T}^{-1}\mathbf{T}_{k,\ell}\Delta\mathbf{T} = [\Delta\mathbf{R}^\intercal\mathbf{R}_{k,\ell}\Delta\mathbf{R} \mid \Delta\mathbf{R}^\intercal(\mathbf{R}_{k,\ell}\Delta\boldsymbol{t} + \boldsymbol{t}_{k,\ell} - \Delta\boldsymbol{t})] \tag{3.2}$$

or alternatively expressed within $\mathcal{C}$:

$$\begin{aligned} \mathbf{T}_{k,\ell}\Delta\mathbf{T} &= \Delta\mathbf{T}\mathbf{T}'_{k,\ell} \\ [\mathbf{R}_{k,\ell}\Delta\mathbf{R} \mid \mathbf{R}_{k,\ell}\Delta\boldsymbol{t} + \boldsymbol{t}_{k,\ell}] &= [\Delta\mathbf{R}\mathbf{R}'_{k,\ell} \mid \Delta\mathbf{R}\boldsymbol{t}'_{k,\ell} + \Delta\boldsymbol{t}] \end{aligned} \tag{3.3}$$

The translational part can be interpreted geometrically: The translation of the second camera within the coordinate frame of the first camera consists of the translation $\boldsymbol{t}_{k,\ell}$ of the first camera itself and the translation $\mathbf{R}_{k,\ell}\Delta\boldsymbol{t}$ resulting from rotating the second camera around the first.

Equation (3.3) can be decomposed into a rotational part

$$\mathbf{R}_{k,\ell}\Delta\mathbf{R} = \Delta\mathbf{R}\mathbf{R}'_{k,\ell} \quad \forall\, k, \ell \tag{3.4}$$

and a translational part

$$\mathbf{R}_{k,\ell}\Delta\boldsymbol{t} + \boldsymbol{t}_{k,\ell} = \Delta\mathbf{R}\boldsymbol{t}'_{k,\ell} + \Delta\boldsymbol{t} \quad \forall\, k, \ell \tag{3.5}$$

Methods to derive parameters of the eye-to-eye transformation $\Delta\mathbf{T}$ from eq. (3.3) are denoted as *(geometric) eye-to-eye calibration* in the following. In classical hand-eye calibration, a common approach is to solve eq. (3.4) for $\Delta\mathbf{R}$ first and use the estimated rotation to solve eq. (3.5) with respect to $\Delta\boldsymbol{t}$ only, reducing the latter to a linear equation system. This method is denoted as *decoupled estimation* while the simultaneous solution of eq. (3.4) and (3.5) for $\Delta\mathbf{R}$ and $\Delta\boldsymbol{t}$ is referred to as *combined estimation* here.[3]

[3] also referred to as *separable* resp. *simultaneous* estimation in the literature [SEH12]

Extended eye-to-eye calibration problem Given that the reference coordinate frames of the coupled cameras have different scaling, the eye-to-eye transformation relating $\mathcal{C}'$ to $\mathcal{C}$ is described by a similarity transformation $\Delta\mathbf{S}$ instead of a Euclidean transformation $\Delta\mathbf{T}$. An additional isometric scaling parameter $\Delta\lambda \in \mathbb{R}_{>0}$ is introduced:

$$\boldsymbol{X} = \Delta\mathbf{S}\boldsymbol{X}' \quad \text{i. e., } \boldsymbol{X} = \Delta\lambda\Delta\mathbf{R}\boldsymbol{X}' + \Delta\boldsymbol{t} \tag{3.6}$$

and

$$\begin{aligned} \mathbf{T}_{k,\ell}\Delta\mathbf{S} &= \Delta\mathbf{S}\mathbf{T}'_{k,\ell} \\ [\mathbf{R}_{k,\ell}\Delta\mathbf{R} \mid \mathbf{R}_{k,\ell}\Delta\boldsymbol{t} + \boldsymbol{t}_{k,\ell}] &= [\Delta\mathbf{R}\mathbf{R}'_{k,\ell} \mid \Delta\lambda\Delta\mathbf{R}\Delta\boldsymbol{t}'_{k,\ell} + \Delta\boldsymbol{t}] \end{aligned} \tag{3.7}$$

with translational part

$$\mathbf{R}_{k,\ell}\Delta\boldsymbol{t} + \boldsymbol{t}_{k,\ell} = \Delta\lambda\Delta\mathbf{R}\boldsymbol{t}'_{k,\ell} + \Delta\boldsymbol{t} \quad \forall\, k, \ell \tag{3.8}$$

The computation of $\Delta\mathbf{S}$ from eq. (3.7) is denoted as *extended (geometric) eye-to-eye calibration* here.

Eye-and-world calibration problem Given m corresponding absolute poses $(\mathbf{W}_k, \mathbf{W}'_k)$, $k = 1, \ldots, m$ for both cameras relating to different fixed world coordinate frames $\mathcal{W}$, $\mathcal{W}'$, we seek Euclidean transformations $\Delta\mathbf{T}$ relating $\mathcal{C}'$ to $\mathcal{C}$ and $\Delta\mathbf{W}$ relating $\mathcal{W}'$ to $\mathcal{W}$:

$$\begin{aligned} \mathbf{W}_k\Delta\mathbf{T} &= \Delta\mathbf{W}\mathbf{W}'_k \\ [\mathbf{Q}_k\Delta\mathbf{R} \mid \mathbf{Q}_k\Delta\boldsymbol{t} + \boldsymbol{w}_k] &= [\Delta\mathbf{Q}\mathbf{R}'_k \mid \Delta\mathbf{R}\Delta\boldsymbol{w}'_k + \Delta\boldsymbol{w}] \end{aligned} \tag{3.9}$$

To distinguish between pose transformations between camera coordinate frames and transformations from camera coordinate frames to world coordinate frames, we will identify the latter by $\mathbf{W} = [\mathbf{Q} \mid \boldsymbol{w}]$ instead of $\mathbf{T} = [\mathbf{R} \mid \boldsymbol{t}]$. The coordinate transformation $\Delta\mathbf{W} = [\Delta\mathbf{Q} \mid \Delta\boldsymbol{w}]$ between world coordinate frames is denoted as *world-to-world transformation*.[4] The relationship between these pose transformations is illustrated in Fig. 3.2.

[4] replacing the term "base-world transformation" used in classical hand-eye calibration

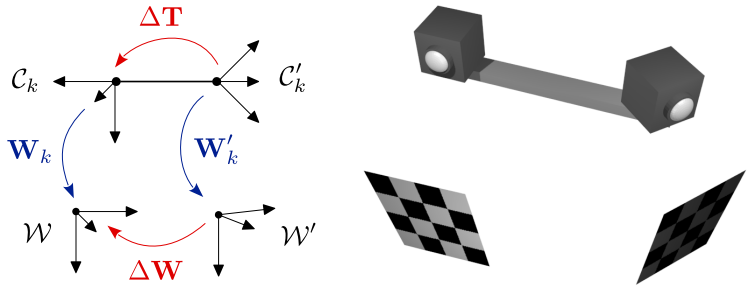

Figure 3.2. Local coordinate frames and absolute coordinate transformations of two rigidly coupled cameras with respect to different world coordinate frames.

In classical hand-eye calibration, eq. (3.9) is often solved for $\Delta\mathbf{T}$ first from the corresponding eye-to-eye calibration problem resulting from considering relative poses instead of absolute poses:

$$\underbrace{\mathbf{W}_k{}^{-1}\mathbf{W}_\ell}_{\mathbf{T}_{k,\ell}}\Delta\mathbf{T} = \Delta\mathbf{T}\underbrace{\mathbf{W}'_k{}^{-1}\mathbf{W}'_\ell}_{\mathbf{T}'_{k,\ell}} \quad \forall\, k, \ell \tag{3.10}$$

The remaining world-to-world transformation is found afterwards by inserting $\Delta\mathbf{T}$ into eq. (3.9) and solving for $\Delta\mathbf{W}$:

$$\Delta\mathbf{W} = \mathbf{W}_k\Delta\mathbf{T}\mathbf{W}'_k{}^{-1} \quad \forall\, k \tag{3.11}$$

which can be solved by averaging over all $\mathbf{W}_k\Delta\mathbf{T}\mathbf{W}'_k{}^{-1}$ (see A.2).

3.2 Related Work

A detailed overview of hand-eye and base-world calibration methods based on the matrix equations (3.3) and (3.9), also known as $\mathbf{AX} = \mathbf{XB}$ and $\mathbf{AX} = \mathbf{ZB}$ in the literature, can be found in [SEH12]. Table 3.1 and 3.2 give

an overview of the relevant methods that are described in the following.

Direct methods The seminal contribution to robotic hand-eye calibration was the decoupling of hand-eye calibration from conventional robot kinematic model calibration which was proposed first by Shiu & Ahmad [SA89] and simultaneously by Tsai & Lenz [TL89] in 1987.

Shiu & Ahmad [SA89] were the first authors who formulated the problem mathematically as the homogeneous matrix equation $\mathbf{AX} = \mathbf{XB}$ with the measured homogeneous transformation matrices $\mathbf{A}$ of the hand and $\mathbf{B}$ of the eye between the current and the previous location. $\mathbf{X}$ determines the unknown homogeneous coordinate transformation from the eye's coordinate frame $\mathcal{B}$ to the hand's coordinate frame $\mathcal{A}$. Almost all following approaches to hand-eye calibration are based on this equation, although the association of hand and eye with either $\mathbf{A}$ or $\mathbf{B}$ and the direction of the coordinate transformation $\mathbf{X}$ varies in the literature on hand-eye calibration. In our work we use the symbols $\Delta\mathbf{T}$, $\mathbf{T}$, $\mathbf{T}'$ instead of $\mathbf{X}$, $\mathbf{A}$, $\mathbf{B}$.[5]

The authors' approach to solve the matrix equation with respect to $\Delta\mathbf{T}$ is based on the geometric interpretation of the equation, in particular regarding the eigenvalues and eigenvectors of the rotation matrices involved. They provided as major contributions of this paper conditions on the uniqueness of the solution and the scheme of decoupled computation of the rotational and translational part of $\Delta\mathbf{T}$ using closed form-solutions. It is proved that the general solution has one degree of freedom in rotation and one degree of freedom in translation. Hence, at least two seperate motions must be given satisfying certain constraints such as non-zero rotation and non-parallel rotation axes. The ambiguity of the rotational part of the solution is solved here by computing general solutions from both motion pairs and finding particular solutions which are equal by solving a linear equation system. When the rotational part of $\Delta\mathbf{T}$ is known, the translational part can also be found by solving a linear equation system. While the theoretical formulation of the problem by the authors was setting the stage for all following approaches, the technical solution of the problem presented here was very inefficient and in need for improvement.

[5] resp. $\Delta\mathbf{T}_{i,j}$, $\mathbf{T}^{(i)}$, $\mathbf{T}^{(j)}$ when considering more than two cameras

Tsai & Lenz presented their attempt to decouple hand-eye calibration from robot calibration also in 1987 but reworked their approach after Shiu & Ahmad's publication in [TL89] where they also comment on the related work. Although starting from the same point, the technical details of their approach differ largely. While in the approach proposed by Shiu & Ahmad, the number of unknowns to be estimated increases linearly with the number of motions, Tsai & Lenz propose to estimate the rotational part of $\Delta\mathbf{T}$ from a linear equation system composed of all motion constraints at once, keeping the number of parameters fixed. The rotation $\Delta\mathbf{R}$ is derived from aligning the rotation axes of corresponding rotations with each other. A closed-form solution is provided by parametrizing $\Delta\mathbf{R}$ as a Cayley-Gibbs-Rodrigues vector. The translational part $\Delta\boldsymbol{t}$ of $\Delta\mathbf{T}$ is computed afterwards from a linear equation systems as in [SA89].

Wang [Wan92] proposed a similar solution for the rotational part based on the angle/axis parametrization and compared [SA89] and [TL89], concluding that Tsai & Lenz's method is the best on average.

Park & Martin [PM94] (described also by Baillot et al. [Bai+03]) proposed a solution for the rotational part based on the rotation matrix representation for the hand-eye rotation instead of angle/axis-based representations. They reduce the problem to finding the optimal rotation matrix aligning corresponding rotation vectors with each other and solve it efficiently via eigendecomposition (see A.3.2, first part).

Chou & Kamel [CK88; CK91] introduced the unit quaternion representation for rotations into hand-eye calibration. Minimizing the rotational part with respect to quaternions yields a linear equation system in the entries of a unit quaternion $\Delta\mathbf{q}$. A similar method was used by us in [Esq07].

Zhuang & Roth [ZR91] also use unit quaternions to derive a closed-form solution to the hand-eye rotation problem that is more or less identical to the solution proposed by Tsai & Lenz.

Another approach based on unit quaternions was proposed by Horaud & Dornaika [HD95]. The idea is similar to Park & Martin, but a unit quaternion is estimated instead of a rotation matrix to find the relative rotation between corresponding rotation vectors (see A.3.2, second part).

The authors were also the first to propose an iterative procedure for combined estimation of rotation and translation parameters using the quaternion parametrization via nonlinear least squares solution of eq. (3.3) and to address the issue of noisy measurements.

Andreff et al. [AHE99] proposed a linear solution for simultaneous rotation and translation estimation. They formulate the rotation constraint in terms of rotation matrices and the Kronecker product as a linear equation system. The constraints of the resulting rotation matrix are enforced afterwards via orthonormalization using singular value decomposition. However, the authors suggest to decouple rotation and translation estimation since the translational part is not recalculated after orthonormalization of the rotational part. This has been confirmed by Liang & Mao [LM08] who propose a very similar approach.

Decoupling rotation and translation estimation leads to simple problem formulations that can be solved with direct methods. A disadvantage of this technique is that errors in the estimation of the optimal hand-eye rotation $\Delta\mathbf{R}$ propagate into the estimation of the translation $\Delta\boldsymbol{t}$. Chen [Che91] suggests that direct methods estimating rotation and translation simulataneously are supposed to provide more accurate results. He describes the hand-eye calibration problem in terms of finding the relative pose between 3d lines given by the screw axes of rigidly coupled motions. Daniilidis, Sommer & Bayro-Corrochano [DB96; BDS97; Dan99] formulate Chen's approach algebraically using dual quaternions instead of screw motions which can be understood as the dual case of the approach by Chou & Kamel. Both approaches yield a linear equation system in the 8 parameters of a dual quaternion subject to the unit length constraint which is solved via singular value decomposition. Lu & Chou [LC95] derive the same linear equation system as Daniilidis et al. using quaternion algebra only and solve it using Gaussian elimination and Schur decomposition. Another very similar method based on screw motions was proposed by Zhao & Liu [ZL06]. Their results are virtually identical to Daniilidis et al.

Extension of hand-eye calibration by estimation of a relative scale $\Delta\lambda$ between the coupled coordinate frames was introduced by Andreff et al. [AHE01] w.r.t. their linear solution from [AHE99]. Schmidt, Vogt & Nie-

mann [SVN05] extended the dual quaternion based method, encoding the relative scale implicitly by the norm of the quaternion.

Nonlinear optimization Given an initial solution provided by a direct method as described above, iterative methods for nonlinear optimization such as the *Levenberg-Marquardt algorithm* [Mor78] can be used to solve eq. (3.3) simultaneously for rotation and translation based on minimizing an error metric on $\mathrm{SE}(3)$. The first approach in this direction was presented by Zhuang & Shiu [ZS93], minimizing the matrix difference $\|\mathbf{AX} - \mathbf{XB}\|$ via the Levenberg-Marquardt algorithm. Rotation matrices are parametrized using Euler angles. A very similar approach was presented by Fassi & Legnani [FL05] who also provide a geometric interpretation of the hand-eye calibration problem. Mao et al. [MHJ10] propose nonlinear optimization of their linear solution from [LM08] using Euler angles, resulting in another similar method.

Horaud & Dornaika combine error functions for the rotational part (3.4) and the translational part (3.5) using quaternions [HD95] and rotation matrices [DH98] for rotation representation. The resulting nonlinear error function is minimized via the Levenberg-Marquardt algorithm. Rotational and translational errors are weighted with respect to each other using heuristic weights. Constraints on the rotation parametrization are enforced via penalty terms. Strobl & Hirzinger [SH06] advise to use geometrically meaningful error measures such as the angle distance for rotations and improve Horaud & Dornaika's method by replacing the heuristic weights by adaptive weights derived from a stochastic error model. Kim et al. [Kim+10] extend this method furthing using the *Minimum Variance method*.

Ikits [Iki00] describes pose coregistration as a nonlinear least squares problem using a reduced parametrization for unit quaternions and weighting the residuals of the rotation and translation equations with the inverse covariance matrix of measured relative poses. The resulting problem can be solved in terms of an ordinary least squares problem or as a total least squares problem. This method is closest to our requirements since it addresses general pose measuring devices with potentially very different measurement error distributions explicitly.

Hand-eye and base-world calibration Estimation of hand-eye and base-world transformation has been addressed by straightforward extension of the solutions for the hand-eye calibration problem described above (see [SH06; Kim+10]). Zhuang, Roth & Sudhakar [ZRS94] were the first to describe a decoupled closed-form solution, using quaternions for rotation representation which has been rendered more precisely by Dornaika & Horaud [DH98]. Rémy et al. [Rém+97] described the first iterative solution via nonlinear optimization using the Levenberg-Marquardt algorithm. Further iterative solutions have been proposed by Hirsh, DeSouza & Kak [HDK01] and Ernst et al. [Ern+12].

Table 3.1. Overview of hand-eye calibration methods in chronological order (properties are **D**: decoupled, **C**: combined, **N**: nonlinear combined, **E**: extended).

Method	Parametrization	D	C	N	E
Tsai & Lenz [TL89]	rotation vector	•			
Shiu & Ahmad [SA89]	rotation axis	•			
Chou & Kamel [CK91]	unit quaternion	•			
Chen [Che91]	screw axis		•		
Wang [Wan92]	rotation axis	•			
Zhuang & Shiu [ZS93]	Euler angles			•	
Park & Martin [PM94]	rotation matrix	•			
Horaud & Dornaika [HD95]	unit quaternion	•		•	
Lu & Chou [LC95]	unit dual quaternion		•		
Daniilidis et al. [DB96]	unit dual quaternion		•		
Dornaika & Horaud [DH98]	rotation matrix			•	
Andreff et al. [AHE01]	rotation matrix, scale	•	•		•
Ikits [Iki00]	reduced unit quat.			•	
Schmidt et al. [SVN05]	scaled dual quaternion			•	•
Fassi & Legnani [FL05]	Euler angles			•	
Strobl & Hirzinger [SH06]	unit quaternion			•	
Liang & Mao [LM08]	rotation matrix	•	•		
Mao et al. [MHJ10]	Euler angles			•	

Table 3.2. Overview of hand-eye calibration methods w.r.t. parametrization of $\Delta\mathbf{T}$ and distance measure used in the error function (NL denotes nonlinear methods).

Parametrization	$\|\mathbf{R} - \hat{\mathbf{R}}\|$	$\|\mathbf{q} - \hat{\mathbf{q}}\|$	$d_{\angle}(\mathbf{R}, \hat{\mathbf{R}})$	$\|\boldsymbol{r} - \hat{\boldsymbol{r}}\|$
Unit quaternion		[CK91]	[SH06]NL	[HD95]NL
Dual quaternion		[DB96; LC95]		
Screw axis				[Che91]
Rotation matrix	[AHE01; LM08] [DH98]NL			[PM94]
Rotation vector				[TL89; SA89] [Wan92]
Euler angles	[ZS93; FL05]NL [MHJ10]NL			

3.3 Rigid Motion Constraints

In the following we will analyze the properties of rigidly coupled motions formally. Direct method solving the geometric eye-to-eye calibration problem are based on rigid motion constraints derived from eq. (3.3). We will revisit these internal constraints in Chapter 5 and discuss how to integrate them into the Structure from Motion process. For further information we refer the reader to Sec. II-D in [TL89].

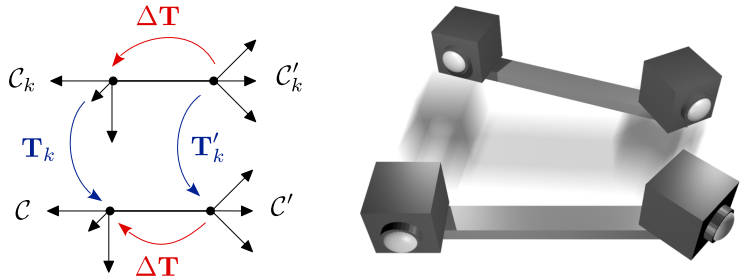

Figure 3.3. Local coordinate frames of two rigidly coupled cameras with respect to a reference pose.

Pose transformations are defined as in Sec. 3.1. For sake of readability we consider only relative poses w.r.t. the reference coordinate frames $\mathcal{C}$, $\mathcal{C}'$ here, denoted by $\mathbf{T}_k$, $\mathbf{T}'_k$. However, the same statements hold for relative poses $\mathbf{T}_{k,\ell}$, $\mathbf{T}'_{k,\ell}$ between any locations of the camera rig. Figure 3.3 illustrates the pose transformations under consideration. In order to simplify the notation of the following formulas further, we will drop the subscript indices completely when a single relative pose is considered.

The *rigid motion constraint* introduced in eq. (3.3) constitutes the primary theorem for eye-to-eye calibration:

Theorem 3.1 (Rigid Motion Constraint)**.** *Given rigidly coupled relative poses* $\mathbf{T}$ *and* $\mathbf{T}'$*, the following constraint holds:*

$$\mathbf{T}\Delta\mathbf{T} = \Delta\mathbf{T}\mathbf{T}' \tag{3.12}$$

which can be decomposed into a *rotation constraint*:

$$\mathbf{R}\Delta\mathbf{R} = \Delta\mathbf{R}\mathbf{R}' \tag{3.13}$$

and a *translation constraint*:

$$\mathbf{R}\Delta\boldsymbol{t} + \boldsymbol{t} = \Delta\mathbf{R}\boldsymbol{t}' + \Delta\boldsymbol{t} \tag{3.14}$$

Proof. The rigid motion constraint can be derived from the two possible shortest concatenations of coordinate transformations from $\mathcal{C}'_k$ to $\mathcal{C}$ as depicted in Fig. 3.3.

Corollary As a trivial consequence of eq. (3.14), the magnitude of translation is equal for all rigidly coupled cameras under pure translation or rotation around the offset vector $\Delta\boldsymbol{t}$:

$$(\mathbf{R} - \mathbf{I})\Delta\boldsymbol{t} = \mathbf{0} \quad \Rightarrow \quad \|\boldsymbol{t}\| = \|\boldsymbol{t}'\| \tag{3.15}$$

The same statement holds for collocated cameras, i. e., $\Delta\boldsymbol{t} = \mathbf{0}$.

A useful consequence of the rigid motion constraint is the fact that the

rotation axes of rigidly coupled rotations are also related by the eye-to-eye rotation $\Delta\mathbf{R}$. A second implication is that the angle of rotation and the magnitude of translation along the rotation axis ("pitch") are identical for rigidly coupled cameras. These constraints have been formulated by Chen as the *Screw Congruence Theorem* using screw motion theory [Che91]:

Lemma 3.2 (Rotation Axis Constraint)**.** *Given rigidly coupled rotations* $\mathbf{R}_{\boldsymbol{r},\alpha}$ *and* $\mathbf{R}_{\boldsymbol{r}',\alpha'}$*, the following constraint holds:*

$$\boldsymbol{r} \sim \Delta\mathbf{R}\boldsymbol{r}' \tag{3.16}$$

Proof. Since the rotation axis is invariant under rotation, we have $\mathbf{R}'\boldsymbol{r}' = \boldsymbol{r}'$. Using the identity from eq. (3.13) we obtain:

$$\Delta\mathbf{R}\boldsymbol{r}' = \Delta\mathbf{R}\mathbf{R}'\boldsymbol{r}' = \mathbf{R}\Delta\mathbf{R}\boldsymbol{r}'$$

Since $\Delta\mathbf{R}\boldsymbol{r}'$ is invariant under rotation by $\mathbf{R}$, it defines the rotation axis of $\mathbf{R}$ up to a sign ambiguity $\boldsymbol{r} = \pm\Delta\mathbf{R}\boldsymbol{r}'$.

Lemma 3.3 (Angle and Pitch Constraints)**.** *Given rigidly coupled pose transformations* $\mathbf{T} = [\mathbf{R}_{\boldsymbol{r},\alpha} \mid \boldsymbol{t}]$ *and* $\mathbf{T}' = [\mathbf{R}_{\boldsymbol{r}',\alpha'} \mid \boldsymbol{t}']$ *with* $\alpha, \alpha' \in (-\pi, \pi)$ *and* $\boldsymbol{r} = \Delta\mathbf{R}\boldsymbol{r}'$*, the following* angle constraint *holds:*

$$\alpha = \alpha' \tag{3.17}$$

and the following pitch constraint *holds:*

$$p = \boldsymbol{r}^\top\boldsymbol{t} = \boldsymbol{r}'^\top\boldsymbol{t}' = p' \tag{3.18}$$

Proof. The first part can be derived from the facts that the eigenvalues of a non-zero rotation matrix are given by 1 and $e^{\pm i\alpha}$ where α is the rotation angle and that the eigenvector corresponding to the eigenvalue 1 is given by the rotation axis. Since $\Delta\mathbf{R}$ is orthonormal, left and right multiplication of $\mathbf{R}$ with $\Delta\mathbf{R}^\top$ and $\Delta\mathbf{R}$ does not change the eigenvalues. Therefore, $\mathbf{R}' = \Delta\mathbf{R}^\top\mathbf{R}\Delta\mathbf{R}$ has the same eigenvalues as $\mathbf{R}$, i. e., $\alpha' = \pm\alpha$. Taking the sign of the corresponding eigenvector $\boldsymbol{r}' = \Delta\mathbf{R}\boldsymbol{r}$ into account,

we have $\alpha = \alpha'$ (see Sec. 4.1 in [Che91]).

The second part is proved using the identities $\boldsymbol{t} = \Delta\mathbf{R}\boldsymbol{t}' + \Delta\boldsymbol{t} - \mathbf{R}\Delta\boldsymbol{t}$ from eq. (3.14) and $\boldsymbol{r} = \Delta\mathbf{R}\boldsymbol{r}'$:

$$\begin{aligned} \boldsymbol{r}^\top\boldsymbol{t} &= \boldsymbol{r}^\top(\Delta\mathbf{R}\boldsymbol{t}' + (\mathbf{I} - \mathbf{R})\Delta\boldsymbol{t}) \\ &= \boldsymbol{r}'^\top\Delta\mathbf{R}^\top\Delta\mathbf{R}\boldsymbol{t}' + \boldsymbol{r}^\top(\mathbf{I} - \mathbf{R})\Delta\boldsymbol{t} \\ &= \boldsymbol{r}'^\top\boldsymbol{t}' + \boldsymbol{r}^\top\Delta\boldsymbol{t} - \boldsymbol{r}^\top\Delta\boldsymbol{t} = \boldsymbol{r}'^\top\boldsymbol{t}' \end{aligned}$$

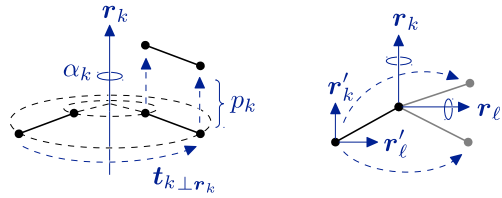

Figure 3.4. Rigidly coupled motions exhibit the same rotation angle and "pitch", i. e., magnitude of translation along the rotation axis (left). Rotation axes of different rigidly coupled motions confine the same angle (right).

From a geometric viewpoint, the congruence theorem as depicted in Fig. 3.4 (left) is quite obvious: Rigidly coupled cameras undergo rotation by the same absolute rotation angle within different coordinate frames, i. e., around rotation axes that are rotated with respect to each other. Since the rotation of the first camera causes only translation orthogonal to its rotation axis for the second camera, the magnitude of translation along the rotation axes must be the same for both cameras since it is produced by the same pure translational motion.

Note that for general rigidly coupled motions, the identities in the congruence theorem are only valid for the absolute values of angle α and pitch p. Choosing the direction of the rotation axes so that $\boldsymbol{r} = \Delta\mathbf{R}\boldsymbol{r}'$ holds, this sign ambiguity is fixed.[6]

[6] In fact, either $\boldsymbol{r} = \Delta\mathbf{R}\boldsymbol{r}'$, $\alpha = \alpha'$, $p = p'$ or $\boldsymbol{r} = -\Delta\mathbf{R}\boldsymbol{r}'$, $\alpha = -\alpha'$, $p = -p'$ holds for all rigidly coupled motions as shown by Chen [Che91].

Corollary From the rotation axis constraint and the angle constraint we can deduce that for any rotation vector representation $\boldsymbol{w}$, $\boldsymbol{w}'$ of rigidly coupled rotations such as the Euler vector, the reduced Euler-Rodrigues vector, or the Cayley-Gibbs-Rodrigues vector defined in Sec. 2.3.2, the constraint $\boldsymbol{w} = \Delta\mathbf{R}\boldsymbol{w}'$ holds, in particular $\|\boldsymbol{w}\| = \|\boldsymbol{w}'\|$.

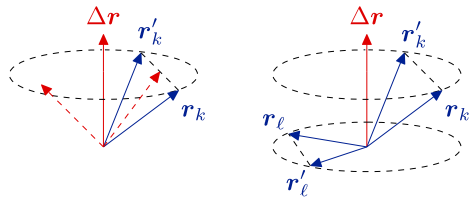

Figure 3.5. Rotation axes of rigidly coupled motions are related by the eye-to-eye rotation $\Delta\mathbf{R}$. For a single motion, the eye-to-eye rotation axis $\Delta\boldsymbol{r}$ is located in the bisecting plane between rotation axes (left), for two motions it is parallel to the intersection line of bisecting planes (right).

Shiu & Ahmad [SA89] and Tsai & Lenz [TL89] also identified the following constraint between the rotation axes of rigidly coupled rotations and the eye-to-eye rotation axis that is used in their proposed solutions:

Lemma 3.4 (Bisecting Plane Constraints)**.** *Given rigidly coupled rotations* $\mathbf{R}_{\boldsymbol{r},\alpha}$ *and* $\mathbf{R}_{\boldsymbol{r}',\alpha'}$ *related by* $\Delta\mathbf{R} = \mathbf{R}_{\Delta\boldsymbol{r},\Delta\alpha}$*, the following constraints holds:*

$$\boldsymbol{r}^\mathsf{T}\Delta\boldsymbol{r} = \boldsymbol{r}'^\mathsf{T}\Delta\boldsymbol{r} \quad \textit{resp.} \quad (\boldsymbol{r}' - \boldsymbol{r})^\mathsf{T}\Delta\boldsymbol{r} = 0 \tag{3.19}$$

$$(\boldsymbol{r}' - \boldsymbol{r}) \sim (\boldsymbol{r} + \boldsymbol{r}') \times \Delta\boldsymbol{r} \tag{3.20}$$

This lemma states that $\Delta\boldsymbol{r}$ is located within the bisecting plane[7] of rotation axes $\boldsymbol{r}$ and $\boldsymbol{r}'$ for each rigidly coupled rotation as depicted in Fig. 3.5.

Proof. Since $\Delta\mathbf{R}^\mathsf{T}\Delta\boldsymbol{r} = \Delta\boldsymbol{r}$ and $\boldsymbol{r} = \Delta\mathbf{R}\boldsymbol{r}'$, we obtain:

$$\boldsymbol{r}^\mathsf{T}\Delta\boldsymbol{r} = \boldsymbol{r}'^\mathsf{T}\Delta\mathbf{R}^\mathsf{T}\Delta\boldsymbol{r} = \boldsymbol{r}'^\mathsf{T}\Delta\boldsymbol{r}$$

[7] i. e., the plane orthogonal to $\boldsymbol{r} - \boldsymbol{r}'$ that contains $\boldsymbol{r} + \boldsymbol{r}'$ and $\boldsymbol{r} \times \boldsymbol{r}'$ in particular

The second part follows from $(\boldsymbol{r}' - \boldsymbol{r}) \perp \Delta\boldsymbol{r}$ and $(\boldsymbol{r}' - \boldsymbol{r}) \perp (\boldsymbol{r} + \boldsymbol{r}')$. The latter can be easily proved:

$$(\boldsymbol{r}' - \boldsymbol{r})^\mathsf{T}(\boldsymbol{r} + \boldsymbol{r}') = \boldsymbol{r}'^\mathsf{T}\boldsymbol{r}' - \boldsymbol{r}^\mathsf{T}\boldsymbol{r} = 0$$

The angle and pitch constraints described above are useful to constrain individual rigidly coupled motions since they can be expressed without explicit knowlegde of the eye-to-eye transformation. For multiple rigidly coupled poses, rotation axes are pairwise constrained to confine the same angle (see Fig. 3.4, right):

Lemma 3.5 (Inter-Axis Angle Constraint)**.** *Given m rigidly coupled rotations $\mathbf{R}_{\boldsymbol{r}_k,\alpha_k}$ and $\mathbf{R}_{\boldsymbol{r}'_k,\alpha'_k}$, $k = 1,\ldots,m$, the following constraint holds for all $k,\ell = 1,\ldots,m$:*

$$d_\angle(\boldsymbol{r}_k, \boldsymbol{r}_\ell) = d_\angle(\boldsymbol{r}'_k, \boldsymbol{r}'_\ell) \tag{3.21}$$

Proof. Since there exists a rotation $\Delta\mathbf{R}$ with $\boldsymbol{r}_k = \Delta\mathbf{R}\boldsymbol{r}'_k$, we have:

$$\boldsymbol{r}_k^\mathsf{T}\boldsymbol{r}_\ell = \boldsymbol{r}_k'^\mathsf{T}\Delta\mathbf{R}^\mathsf{T}\Delta\mathbf{R}\boldsymbol{r}'_\ell = \boldsymbol{r}_k'^\mathsf{T}\boldsymbol{r}'_\ell$$

Finally, the transformation of rigidly coupled motions and the rigid coupling parameters subject to change of the reference coordinate frames is described by the following theorem:

Theorem 3.6 (Change of Reference Coordinate Frame)**.** *Given rigidly coupled relative poses $\mathbf{T}$, $\mathbf{T}'$ and similarity transformations $\mathbf{S}$, $\mathbf{S}'$ describing the change of reference coordinate frame for the first and second camera, consider the rigid motion equation w.r.t. the target reference coordinate frames $\tilde{\mathcal{C}}$, $\tilde{\mathcal{C}}'$:*

$$(\mathbf{S}\mathbf{T}\mathbf{S}^{-1})\Delta\tilde{\mathbf{T}} = \Delta\tilde{\mathbf{T}}(\mathbf{S}'\mathbf{T}'\mathbf{S}'^{-1}) \tag{3.22}$$

Now $\Delta\tilde{\mathbf{T}}$ is a solution to eq. (3.22) if and only if $\Delta\mathbf{T} = \mathbf{S}^{-1}\Delta\tilde{\mathbf{T}}\mathbf{S}'$ is a solution to the original equation $\mathbf{T}\Delta\mathbf{T} = \Delta\mathbf{T}\mathbf{T}'$.

Proof. Assume first that $\Delta\tilde{\mathbf{T}}$ is a solution to eq. (3.22). Inserting the

substitution $\Delta\mathbf{T} = \mathbf{S}^{-1}\Delta\tilde{\mathbf{T}}\mathbf{S}'$ into eq. (3.12) yields:

$$\begin{aligned}\mathbf{T}\Delta\mathbf{T} &= \mathbf{T}\mathbf{S}^{-1}\Delta\tilde{\mathbf{T}}\mathbf{S}' = \mathbf{S}^{-1}(\mathbf{S}\mathbf{T}\mathbf{S}^{-1})\Delta\tilde{\mathbf{T}}\mathbf{S}' = \mathbf{S}^{-1}\Delta\tilde{\mathbf{T}}(\mathbf{S}'\mathbf{T}'\mathbf{S}'^{-1})\mathbf{S}' \\ &= \mathbf{S}^{-1}\Delta\tilde{\mathbf{T}}\mathbf{S}'\mathbf{T}' = \Delta\mathbf{T}\mathbf{T}'\end{aligned}$$

The converse is proved by assuming first that $\Delta\mathbf{T}$ is a solution to eq. (3.12) and inserting $\Delta\tilde{\mathbf{T}} = \mathbf{S}\Delta\mathbf{T}\mathbf{S}'^{-1}$ into eq. (3.22):

$$\begin{aligned}(\mathbf{S}\mathbf{T}\mathbf{S}^{-1})\Delta\tilde{\mathbf{T}} &= \mathbf{S}\mathbf{T}\mathbf{S}^{-1}\mathbf{S}\Delta\mathbf{T}\mathbf{S}'^{-1} = \mathbf{S}\mathbf{T}\Delta\mathbf{T}\mathbf{S}'^{-1} = \mathbf{S}\Delta\mathbf{T}\mathbf{T}'\mathbf{S}'^{-1} \\ &= \mathbf{S}\Delta\mathbf{T}\mathbf{S}'^{-1}\mathbf{S}'\mathbf{T}'\mathbf{S}'^{-1} = \Delta\tilde{\mathbf{T}}(\mathbf{S}'\mathbf{T}'\mathbf{S}'^{-1})\end{aligned}$$

This strategy, illustrated schematically in Fig. 3.6, can be used to reduce the number of eye-to-eye transformation parameters when partial information about the absolute poses of the cameras is available. An exemplification is given in Sec. 3.6.2 for the case that the orientation and position of a common reference plane within the local camera coordinate frames $\mathcal{C}, \mathcal{C}'$ has been identified. In this case, Euclidean transformations $\mathbf{S}, \mathbf{S}'$ can be found so that $\Delta\tilde{\mathbf{T}}$ can be written as a planar Euclidean transformation within the common reference plane, reducing the number of eye-to-eye transformation parameters to be found by 3.

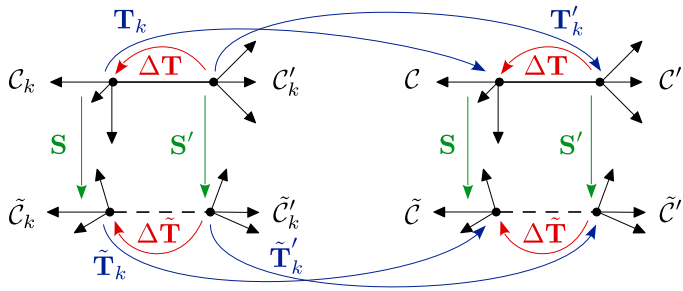

Figure 3.6. Rigidly coupled motions under change of reference coordinate frames via linear tranformations $\mathbf{S}$ and $\mathbf{S}'$ respectively, e. g., relative poses $\mathbf{T}_k$ in $\mathcal{C}$ are transferred to $\tilde{\mathbf{T}}_k = \mathbf{S}\mathbf{T}_k\mathbf{S}^{-1}$ in $\tilde{\mathcal{C}}$. The rigid coupling becomes $\Delta\tilde{\mathbf{T}} = \mathbf{S}\Delta\mathbf{T}\mathbf{S}'^{-1}$.

3.4 Solving the Rigid Motion Equation

3.4.1 Rotation Estimation

In the decoupled approach, eye-to-eye rotation $\Delta\mathbf{R}$ is estimated first from either the rotation constraint (3.13) [CK91; AHE99] or the rotation axis constraint (3.16) [SA89; TL89; PM94; HD95], given at least two distinct non-zero rotations.

For the first case, the resulting problem is similar to eq. (3.4) with respect to specific rotation parametrizations and metrics. Given m rigidly coupled rotations $\mathbf{R}_k, \mathbf{R}'_k$, $k = 1, \ldots, m$, optionally parametrized by unit quaternions $\mathbf{q}_k, \mathbf{q}'_k$ or angle/axis representation $(\boldsymbol{r}_k, \alpha_k)$, $(\boldsymbol{r}'_k, \alpha'_k)$, we will briefly describe the direct methods proposed in the literature:

Minimizing quaternion distance Chou & Kamel [CK91] minimize the residual error of eq. (3.13) in terms of the quaternion distance between the left and right rotation:[8]

$$\min_{\Delta\mathbf{q}\in\mathbb{R}^4} \sum_{k=1}^{m} \|\mathbf{q}_k \cdot \Delta\mathbf{q} - \Delta\mathbf{q} \cdot \mathbf{q}'_k\|^2 \quad \text{subject to } \|\Delta\mathbf{q}\| = 1 \tag{3.23}$$

which can be written using the matrix representation for quaternion multiplication as:

$$\min_{\Delta\mathbf{q}\in\mathbb{R}^4} \sum_{k=1}^{m} \| \left(\mathbf{M}^{\ell}_{\mathbf{q}_k} - \mathbf{M}^{r}_{\mathbf{q}'_k} \right) \Delta\mathbf{q}\|^2 \quad \text{subject to } \|\Delta\mathbf{q}\| = 1$$

Equation (3.23) can be solved efficiently via singular value decomposition of $\mathbf{M} = \sum_{k=1}^{m} (\mathbf{M}^{\ell}_{\mathbf{q}_k} - \mathbf{M}^{r}_{\mathbf{q}'_k})^{\mathsf{T}} (\mathbf{M}^{\ell}_{\mathbf{q}_k} - \mathbf{M}^{r}_{\mathbf{q}'_k})$ as described in C.3.2.

[8] In [Esq07] we minimize the residual error of eq. (3.13) w.r.t. the distance measure $d(\mathbf{q}_1, \mathbf{q}_2) = \sqrt{1 - \mathbf{q}_1^{\mathsf{T}}\mathbf{q}_2}$ instead. The results are identical to the ones obtained from eq. (3.23). In fact, both error metrics are algebraically identical.

Minimizing rotation matrix distance Andreff et al. [AHE99] minimize the residual error of eq. (3.13) in terms of the Frobenius norm of the residual matrix:

$$\min_{\Delta\mathbf{R}\in\mathbb{R}^{3\times3}} \sum_{k=1}^{m} \|\mathbf{R}_k\Delta\mathbf{R} - \Delta\mathbf{R}\mathbf{R}'_k\|^2 \tag{3.24}$$

which can be written using the Kronecker product and the parameter vector $\Delta\boldsymbol{\varrho} = \mathrm{vec}(\Delta\mathbf{R})$ as the following linear least squares system:

$$\min_{\Delta\boldsymbol{\varrho}\in\mathbb{R}^9} \left\| \begin{pmatrix} \mathbf{I}_9 - \mathbf{R}_1 \otimes \mathbf{R}'_1 \\ \vdots \\ \mathbf{I}_9 - \mathbf{R}_m \otimes \mathbf{R}'_m \end{pmatrix} \Delta\boldsymbol{\varrho} \right\|^2 \quad \text{subject to } \|\Delta\boldsymbol{\rho}\| = 1$$

They solve the linear least squares problem using standard methods and enforce the constraints on the resulting vector $\boldsymbol{\varrho}$ to provide a valid rotations matrix afterwards. The rotation matrix $\Delta\mathbf{R}$ closest to $\mathrm{mat}(\boldsymbol{\varrho})$ is computed via orthonormalization as described in Sec. 2.3.2.
Liang & Mao [LM08] propose a similar approach using the equivalent constraint $(\mathbf{R}_k \otimes \mathbf{I} - \mathbf{I} \otimes \mathbf{R}'^{\mathsf{T}}_k)\Delta\boldsymbol{\varrho} = \mathbf{0}$ instead of $(\mathbf{I}_9 - \mathbf{R}_k \otimes \mathbf{R}'_k)\Delta\boldsymbol{\varrho} = \mathbf{0}$.

Rotation vector alignment Different authors such as Park & Martin [PM94] or Horaud & Dornaika [HD95] propose solutions based on (3.16) by finding the optimal rotation $\Delta\mathbf{R}$ between corresponding rotation axes $\boldsymbol{r}_k, \boldsymbol{r}'_k$ or rotation vectors such as the Euler vectors $\boldsymbol{w} = \alpha\boldsymbol{r}$:

$$\min_{\Delta\mathbf{R}} \sum_{k=1}^{m} \|\boldsymbol{w}_k - \Delta\mathbf{R}\boldsymbol{w}'_k\|^2 \tag{3.25}$$

See A.3.2 for numerical solutions for this relative rotation problem using rotation matrices [PM94] or unit quaternions [HD95] to describe $\Delta\mathbf{R}$.

Intersection of rotation axes bisecting planes The original approaches by Shiu & Ahmad [SA89] and Tsai & Lenz [TL89] can be interpreted as estimating the eye-to-eye rotation axis from the intersection between the bisecting plane between corresponding rotation axes from eq. (3.19).

Tsai & Lenz propose a closed-form solution based on the second constraint in eq. (3.19):

$$\min_{\Delta \boldsymbol{w} \in \mathbb{R}^3} \sum_{k=1}^{m} \| [\boldsymbol{r}_k + \boldsymbol{r}'_k]_\times \Delta \boldsymbol{w} - (\boldsymbol{r}'_k - \boldsymbol{r}_k) \|^2 \tag{3.26}$$

and show that $\Delta \boldsymbol{w} = \tan(\frac{\Delta \alpha}{2}) \Delta \boldsymbol{r}$ is the Cayley-Gibbs-Rodrigues vector of the relative rotation $\Delta \mathbf{R}$.

Shiu & Ahmad compute for each pair of rigidly coupled rotation axes $(\boldsymbol{r}_k, \boldsymbol{r}'_k)$ an exact solution to eq. (3.16) as $\Delta \hat{\mathbf{R}}_k$ with rotation axis $\Delta \hat{\boldsymbol{r}}_k = \boldsymbol{r}'_k \times \boldsymbol{r}_k$ and rotation angle $\Delta \hat{\alpha}_k = d_\angle(\boldsymbol{r}_k, \boldsymbol{r}'_k)$. They show that $\mathbf{R}_{\boldsymbol{r}_k, \beta_k} \Delta \hat{\mathbf{R}}_k$ is also a feasible solution for every rotation angle β_k.[9] For two rotations, this leads to finding angles β_1, β_2 that satisfy:

$$\mathbf{R}_{\boldsymbol{r}_1, \beta_1} \Delta \hat{\mathbf{R}}_1 = \mathbf{R}_{\boldsymbol{r}_2, \beta_2} \Delta \hat{\mathbf{R}}_2 \tag{3.27}$$

which can be written using Rodrigues' rotation formula (2.22) as a linear equation system with 9 equations in 4 unknowns $s_k = \sin(\beta_k), c_k = \cos(\beta_k)$, $k \in \{1, 2\}$ of the form:

$$\min_{s_1, c_1, s_2, c_2} \| \underbrace{(\mathbf{A}_1 + s_1 \mathbf{B}_1 + c_1 \mathbf{C}_1)}_{\mathbf{R}_{\boldsymbol{r}_1, \beta_1}} \Delta \hat{\mathbf{R}}_1 - \underbrace{(\mathbf{A}_2 + s_2 \mathbf{B}_2 + c_2 \mathbf{C}_2)}_{\mathbf{R}_{\boldsymbol{r}_2, \beta_2}} \Delta \hat{\mathbf{R}}_2 \|^2 \tag{3.28}$$

with $\mathbf{A}_k = \mathbf{I} + [\boldsymbol{r}_k]_\times^2$, $\mathbf{B}_k = [\boldsymbol{r}_k]_\times$ and $\mathbf{C}_k = -[\boldsymbol{r}_k]_\times^2$. The final solution is retrieved from averaging the resulting rotation matrices $\mathbf{R}_{\boldsymbol{r}_k, \beta_k} \Delta \hat{\mathbf{R}}_k$ with $\beta_k = \operatorname{atan2}(s_k, c_k)$ for $k \in \{1, 2\}$.

Although originally described for exactly two poses, the approach can be adapted in a straightforward way for $m > 2$ rotations. However, since the number of parameters increases linearly with the number of motions, this is not advisable. Note also that Shiu & Ahmad's method as proposed in [SA89] estimates in fact a scaled rotation for each motion since they do not enforce the constraints $s_k^2 + c_k^2 = 1$ in the linear least squares problem.

[9] Note that the rotation axes of the potential eye-to-eye rotations form the bisecting plane.

Singularities in rotation estimation By examination of the linear equation systems solved for the eye-to-eye rotation parameters, singularities arise in two cases: for zero rotation and for rotation around a single axis. The first case occurs for pure translation of the camera rig, the latter for example for purely planar motion. The ambiguities of the solution associated with the second case are described formally by the following theorem:

Lemma 3.7. *Given are rigidly coupled rotations* $\mathbf{R}$, $\mathbf{R}'$ *related by* $\Delta\mathbf{R}$. *Then for any rotation* $\mathbf{R}_{\boldsymbol{r},\beta}$, $\beta \in [-\pi, \pi]$ *around the rotation axis* $\boldsymbol{r}$ *of* $\mathbf{R}$ *the following equation holds:*

$$\mathbf{R}\mathbf{R}_{\boldsymbol{r},\beta}\Delta\mathbf{R} = \mathbf{R}_{\boldsymbol{r},\beta}\Delta\mathbf{R}\mathbf{R}' \tag{3.29}$$

i. e., $\mathbf{R}_{\boldsymbol{r},\beta}\Delta\mathbf{R}$ *is also a solution to the rotation equation eq. (3.13).*

Proof. Since rotations around the same axis commute, we obtain:

$$\mathbf{R}\mathbf{R}_{\boldsymbol{r},\beta}\Delta\mathbf{R} = \mathbf{R}_{\boldsymbol{r},\beta}\mathbf{R}\Delta\mathbf{R} = \mathbf{R}_{\boldsymbol{r},\beta}\Delta\mathbf{R}\mathbf{R}'$$

Corollary From a single rotation of the rig or multiple rigidly coupled rotations with parallel rotation axes, the eye-to-eye rotation $\Delta\mathbf{R}$ can only be determined up to an unknown rotation around the common rotation axis from eq. (3.13).

To determine a unique solution for the eye-to-eye rotation $\Delta\mathbf{R}$ from eq. (3.13), at least two rigidly coupled non-zero rotation pairs $\mathbf{R}_1$, $\mathbf{R}_2$ and $\mathbf{R}'_1$, $\mathbf{R}'_2$ with non-parallel rotation axes $\boldsymbol{r}_1$, $\boldsymbol{r}_2$ resp. $\boldsymbol{r}'_1$, $\boldsymbol{r}'_2$ are needed.

Accuracy of rotation estimation In the general case, the accuracy of the estimated eye-to-eye rotation depends on the error of the rigidly coupled rotations $\mathbf{R}_k$, $\mathbf{R}'_k$ relative to the absolute rotation angles α_k, α'_k. Additionally, the accuracy depends on the minimal angle between rotation axes $\boldsymbol{r}_k$, $\boldsymbol{r}'$. An experimental analysis of sensitivity of different rotation estimation methods w.r.t. input errors is presented in Sec. 3.8.1.

3.4.2 Translation Estimation

Once $\Delta\mathbf{R}$ is known, the linear equation system resulting from eq. (3.14) can be solved for the hand-eye translation $\Delta\boldsymbol{t}$ from rigidly coupled motions with non-zero rotation:

$$(\mathbf{R}_k - \mathbf{I})\Delta\boldsymbol{t} = \Delta\mathbf{R}\boldsymbol{t}'_k - \boldsymbol{t}_k \quad \text{for } k = 1, \ldots, m \tag{3.30}$$

resp.

$$\min_{\Delta\boldsymbol{t}\in\mathbb{R}^3} \sum_{k=1}^{m} \|(\mathbf{R}_k - \mathbf{I})\Delta\boldsymbol{t} - (\Delta\mathbf{R}\boldsymbol{t}'_k - \boldsymbol{t}_k)\|^2$$

Note that the matrix $\mathbf{R}_k - \mathbf{I}$ is rank deficient with its nullspace consisting of all vectors parallel to the rotation axis $\boldsymbol{r}_k$. Therefore, at least $m = 2$ motions with non-parallel rotation axes are needed for a unique solution.

Singularities in translation estimation If $\Delta\mathbf{R}$ is known, the linear equation system eq. (3.30) becomes singular in two cases: either the right side of the equation is zero, i. e., $\Delta\mathbf{R}\boldsymbol{t}'_k = \boldsymbol{t}_k \; \forall k = 1, \ldots, m$, or the matrix composed of $(\mathbf{R}_k - \mathbf{I})$ has rank 2. Apart from the trivial case $\Delta\boldsymbol{t} = \mathbf{0}$, both situations occur only for pure translation of the rig or rotation around a single axis, i. e., under the same conditions where rotation estimation exhibits singularities.

The ambiguities of the solution to the singular equation system are formalized by the following theorem:

Lemma 3.8. *Given are rigidly coupled motions* $[\mathbf{R} \mid \boldsymbol{t}]$, $[\mathbf{R}' \mid \boldsymbol{t}']$ *with non-zero rotation related by* $[\Delta\mathbf{R} \mid \Delta\mathbf{t}]$. *Then for any translation* $p\boldsymbol{r}$ *along the rotation axis* $\boldsymbol{r}$ *of* $\mathbf{R}$ *the following equation holds:*

$$(\mathbf{R} - \mathbf{I})(\Delta\boldsymbol{t} + p\boldsymbol{r}) = \Delta\mathbf{R}\boldsymbol{t}' - \boldsymbol{t} \tag{3.31}$$

i. e., $\Delta\hat{\boldsymbol{t}} = \Delta\boldsymbol{t} + p\boldsymbol{r}_i$ *is also a solution to the translation equation eq. (3.14).*

Proof. Since $p\boldsymbol{r}$ is invariant under rotation by $\mathbf{R}$, we obtain:

$$\begin{aligned}(\mathbf{R}-\mathbf{I})(\Delta\boldsymbol{t}+p\boldsymbol{r}) &= (\mathbf{R}-\mathbf{I})\Delta\boldsymbol{t}+(\mathbf{R}-\mathbf{I})p\boldsymbol{r} \\ &= (\mathbf{R}-\mathbf{I})\Delta\boldsymbol{t}+p\boldsymbol{r}-p\boldsymbol{r}=\Delta\mathbf{R}\boldsymbol{t}'-\boldsymbol{t}\end{aligned}$$

Corollary From a single motion of the rig or multiple rigidly coupled rotations with parallel rotation axes, the eye-to-eye translation $\Delta\boldsymbol{t}$ can only be determined up to an unknown translation along the common rotation axis from eq. (3.14), assuming that $\Delta\mathbf{R}$ is known.

To determine a unique solution for the eye-to-eye transformation $[\Delta\mathbf{R} \mid \Delta\boldsymbol{t}]$ from eq. (3.13) and (3.14), at least two rigidly coupled motions with non-zero rotation and non-parallel rotation axes are needed.

Accuracy of translation estimation In the general case, the accuracy of the estimated eye-to-eye translation depends on the error of the rigidly coupled translations $\boldsymbol{t}_k$, $\boldsymbol{t}'_k$ relative to the absolute translation magnitudes $\|\boldsymbol{t}_k\|$, $\|\boldsymbol{t}'_k\|$. The accuracy depends also on the error of rotation $\mathbf{R}_k$ of the first camera but is independent of rotation $\mathbf{R}'_k$. For decoupled estimation, errors of the previously estimated eye-to-eye rotation $\Delta\mathbf{R}$ are propagated to eye-to-eye translation estimation via the term $\Delta\mathbf{R}\boldsymbol{t}'$. The resulting error is proportional to the angle error of $\Delta\mathbf{R}$ and the length of $\boldsymbol{t}'$.

3.4.3 Combined Rotation and Translation Estimation

Direct methods for the simultaneous solution of eq. (3.3) w.r.t. eye-to-eye rotation and translation are based on a linear formulation of the problem using Euclidean transformation matrices or dual quaternions.

Using Euclidean transformation matrices The linear approach by Andreff et al. [AHE99] can be used for combined estimation:

$$\min_{\Delta\mathbf{R}\in\mathbb{R}^{3\times3},\Delta\boldsymbol{t}\in\mathbb{R}^3}\sum_{k=1}^{m}\left(\|\mathbf{R}_k\Delta\mathbf{R}-\Delta\mathbf{R}\mathbf{R}'_k\|^2+\|(\mathbf{R}_k-\mathbf{I})\Delta\boldsymbol{t}-\Delta\mathbf{R}\boldsymbol{t}'_k+\boldsymbol{t}_k\|^2\right)$$

which can be written similar to eq. (3.24) as a linear least squares system:

$$\min_{\Delta\boldsymbol{\varrho}\in\mathbb{R}^9,\Delta\boldsymbol{t}\in\mathbb{R}^3} \left\| \begin{pmatrix} \mathbf{I}_9 - \mathbf{R}_1 \otimes \mathbf{R}_1' & \mathbf{0}_{9\times 3} \\ \mathbf{I} \otimes \boldsymbol{t}_1'^{\mathsf{T}} & \mathbf{I} - \mathbf{R}_1 \\ \vdots & \vdots \\ \mathbf{I}_9 - \mathbf{R}_m \otimes \mathbf{R}_m' & \mathbf{0}_{9\times 3} \\ \mathbf{I} \otimes \boldsymbol{t}_m'^{\mathsf{T}} & \mathbf{I} - \mathbf{R}_m \end{pmatrix} \begin{pmatrix} \Delta\boldsymbol{\varrho} \\ \Delta\boldsymbol{t} \end{pmatrix} - \begin{pmatrix} \mathbf{0} \\ \boldsymbol{t}_1 \\ \vdots \\ \mathbf{0} \\ \boldsymbol{t}_m \end{pmatrix} \right\|^2 \tag{3.32}$$

However, solving eq. (3.32) using standard methods and enforcing the constraints on the resulting rotation matrix afterwards results in non-optimal estimation of $\Delta\boldsymbol{t}$. Note also that the equation system is singular for $\boldsymbol{t}_m = \mathbf{0}$, i. e., for the case of pure rotation of the first camera.

The implementation of this method is denoted as `Tmat` in the following.

Using dual quaternions Using the quaternion parametrization for rotations, the translation constraint (3.14) can be written equivalently as:

$$\begin{aligned} & \mathbf{q}\cdot\Delta\mathbf{t}\cdot\bar{\mathbf{q}} + \mathbf{t} &&= \Delta\mathbf{q}\cdot\mathbf{t}'\cdot\Delta\bar{\mathbf{q}} + \Delta\mathbf{t} && \mid \cdot\,\Delta\mathbf{q} \\ \Leftrightarrow\quad & \mathbf{q}\cdot\Delta\mathbf{t}\cdot\bar{\mathbf{q}}\cdot\Delta\mathbf{q} + \mathbf{t}\cdot\Delta\mathbf{q} &&= \Delta\mathbf{q}\cdot\mathbf{t}' + \Delta\mathbf{t}\cdot\Delta\mathbf{q} \\ \Leftrightarrow\quad & \mathbf{q}\cdot\underbrace{\Delta\mathbf{t}\cdot\Delta\mathbf{q}}_{\Delta\mathbf{p}}\cdot\bar{\mathbf{q}}' + \mathbf{t}\cdot\Delta\mathbf{q} &&= \Delta\mathbf{q}\cdot\mathbf{t}' + \underbrace{\Delta\mathbf{t}\cdot\Delta\mathbf{q}}_{\Delta\mathbf{p}} \end{aligned}$$

By replacing $\Delta\mathbf{t}\cdot\Delta\mathbf{q}$ with a quaternion $\Delta\mathbf{p}$, we obtain the following linear least squares problem from the rotation and translation constraints:

$$\min_{\Delta\mathbf{q},\Delta\mathbf{p}\in\mathbb{R}^4} \left\| \begin{pmatrix} \vdots & \vdots \\ \mathbf{M}^{\ell}_{\mathbf{q}_k} - \mathbf{M}^{r}_{\mathbf{q}_k'} & \mathbf{0}_{4\times 4} \\ \mathbf{M}^{\ell}_{\mathbf{t}_k} - \mathbf{M}^{r}_{\mathbf{t}_k'} & \mathbf{M}^{\ell}_{\mathbf{q}_k}\mathbf{M}^{r}_{\bar{\mathbf{q}}_k'} - \mathbf{I}_4 \\ \vdots & \vdots \end{pmatrix} \begin{pmatrix} \Delta\mathbf{q} \\ \Delta\mathbf{p} \end{pmatrix} \right\|^2 \tag{3.33}$$

subject to $\|\Delta\mathbf{q}\| = 1$ and $\Delta\mathbf{q}^{\mathsf{T}}\Delta\mathbf{p} = 0$. The second constraint takes account of the fact that the scalar part of $\Delta\mathbf{t}$ is bound to be zero. The eye-to-eye translation can be recovered from the result as $\Delta\mathbf{t} = \Delta\mathbf{p}\cdot\Delta\bar{\mathbf{q}}$.

A very similar equation system is derived from the dual part of eq. (3.12) using dual quaternions to represent pose transformations:

$$\begin{aligned} \check{\mathbf{q}} \cdot \Delta\check{\mathbf{q}} &= \Delta\check{\mathbf{q}} \cdot \check{\mathbf{q}}' \\ \Leftrightarrow \quad \mathbf{q} \cdot \Delta\mathbf{q} &= \Delta\mathbf{q} \cdot \mathbf{q}' \quad \text{and} \\ \mathbf{q} \cdot \Delta\mathbf{p} + \mathbf{p} \cdot \Delta\mathbf{q} &= \Delta\mathbf{q} \cdot \mathbf{p}' + \Delta\mathbf{p} \cdot \mathbf{q}' \end{aligned}$$

leading to the direct method proposed by Daniilidis et al.:[10]

$$\min_{\Delta\mathbf{q},\Delta\mathbf{p}\in\mathbb{R}^4} \left\| \begin{pmatrix} \vdots & \vdots \\ \mathbf{M}^{\ell}_{\mathbf{q}_k} - \mathbf{M}^{r}_{\mathbf{q}'_k} & \mathbf{0}_{4\times 4} \\ \mathbf{M}^{\ell}_{\mathbf{p}_k} - \mathbf{M}^{r}_{\mathbf{p}'_k} & \mathbf{M}^{\ell}_{\mathbf{q}_k} - \mathbf{M}^{r}_{\mathbf{q}'_k} \\ \vdots & \vdots \end{pmatrix} \begin{pmatrix} \Delta\mathbf{q} \\ \Delta\mathbf{p} \end{pmatrix} \right\|^2 \tag{3.34}$$

subject to $\|\Delta\mathbf{q}\| = 1$ and $\Delta\mathbf{q}^{\intercal}\Delta\mathbf{p} = 0$. Conversion from dual quaternion to Euclidean transformation yields $\Delta\mathbf{t} = 2\Delta\mathbf{p} \cdot \Delta\bar{\mathbf{q}}$. Lu & Chou derived virtually the same equation system from the translation constraint by right-multiplying both sides with $\frac{1}{2}\mathbf{q}\Delta\mathbf{q}$ instead of $\Delta\mathbf{q}$. Methods for the solution of eq. (3.33) and (3.34) can be found in C.3.2.

The implementation of the dual quaternion based approach according to Daniilidis et al. is denoted as `dualquat` in the following.

3.4.4 Nonlinear Rotation and Translation Estimation

The direct methods described in Sec. 3.4.1–3.4.3 rely on linear error functions, e. g., the residuals of the rotational equation represent Euclidean distances between rotation matrix elements, quaternion vectors or rotation axes approximating the actual angle error between estimated and observed rotations. Decoupling rotation and translation estimation is a common strategy to provide a linear formulation of the translation equation. Combined estimation w.r.t. nonlinear error functions leads to nonlinear least squares problems that can be solved iteratively via *Nonlinear Optimization*

[10] Note that Daniilidis et al. omit the scalar equations, leaving $6m$ equations for m motions.

strategies. Details on the solution of unconstrained and constrained NLLS problems can be found in C.3.3 and C.3.4.

Given an initial solution for the eye-to-eye transformation $\Delta\mathbf{T}$ represented by a parameter vector $\Delta\boldsymbol{\mu} \in \mathbb{R}^{\mu}$, eye-to-eye rotation $\Delta\mathbf{R} = \mathbf{R}_{\Delta\boldsymbol{\mu}}$ and translation $\Delta\boldsymbol{t} = \boldsymbol{t}_{\Delta\boldsymbol{\mu}}$ can be estimated jointly by minimizing a error function composed of the rotational term based on eq. (3.4) and translational term base on eq. (3.5) as suggested in [HD95], leading to the following nonlinear least squares problem:

$$\begin{aligned}\min_{\Delta\boldsymbol{\mu}} \sum_{k=1}^{m} & \xi_{\text{rot}}\, d_{\text{rot}}(\mathbf{R}_k, \Delta\mathbf{R}\mathbf{R}'_k\Delta\mathbf{R}^{\mathsf{T}})^2 + && \textit{(rotation term)} \\ & \xi_{\text{pos}}\, d_{\text{pos}}(\boldsymbol{t}_k, \Delta\mathbf{R}\boldsymbol{t}'_k + (\mathbf{I} - \mathbf{R}_k)\Delta\boldsymbol{t})^2 && \textit{(translation term)}\end{aligned} \tag{3.35}$$

where d_{rot}, d_{pos} are distance measures for orientations and positions respectively and ξ_{rot}, ξ_{pos} are weighting factors to bring both error measurements together. Method differ by the choice of the parametrization $\Delta\boldsymbol{\mu}$ and the distance measures d_{rot}, d_{pos}. We will only consider methods using a unit quaternion $\Delta\mathbf{q}$ to parametrize $\Delta\mathbf{R}$ due to their computational efficiency.[11] Translation is parametrized as a 3d vector and translation errors are described by Euclidean distance.

▷ Strobl & Hirzinger [SH06] use the angle distance to describe the rotation error term, representing the geometrically most meaningful error measure:

$$\min_{\Delta\mathbf{q},\Delta\boldsymbol{t}} \sum_{k=1}^{m} \xi_{\text{rot}}\,(2\arccos((\mathbf{q}_k \cdot \Delta\mathbf{q})^{\mathsf{T}}(\Delta\mathbf{q} \cdot \mathbf{q}'_k)))^2 + \xi_{\text{pos}}\,\|(\mathbf{R}_k - \mathbf{I})\Delta\boldsymbol{t} - \mathbf{R}_{\Delta\mathbf{q}}\boldsymbol{t}'_k + \boldsymbol{t}_k\|^2$$

▷ Horaud & Dornaika [HD95] use the Euclidean distance between transformed rotation axes, approximating the angle between both vectors:

$$\min_{\Delta\mathbf{q},\Delta\boldsymbol{t}} \sum_{k=1}^{m} \xi_{\text{rot}}\,\|\boldsymbol{r}_k - \mathbf{R}_{\Delta\mathbf{q}}\boldsymbol{r}'_k\|^2 + \xi_{\text{pos}}\,\|(\mathbf{R}_k - \mathbf{I})\Delta\boldsymbol{t} - \mathbf{R}_{\Delta\mathbf{q}}\boldsymbol{t}'_k + \boldsymbol{t}_k\|^2$$

[11] As stated in Sec. 3.2, almost all existing methods use either unit quaternions or Euler angles to represent rotation.

▷ Using the Euclidean distance between unit quaternions as an approximation to the angle distance yields a novel method that can be considered as the combined version of Chou & Kamel's method [CK91]:

$$\min_{\Delta\mathbf{q},\Delta\boldsymbol{t}} \sum_{k=1}^{m} \xi_{\text{rot}} \left\| \mathbf{q}_k \cdot \Delta\mathbf{q} - \Delta\mathbf{q} \cdot \mathbf{q}'_k \right\|^2 + \xi_{\text{pos}} \left\| (\mathbf{R}_k - \mathbf{I})\Delta\boldsymbol{t} - \mathbf{R}_{\Delta\mathbf{q}}\boldsymbol{t}'_k + \boldsymbol{t}_k \right\|^2$$

All these error functions must be minimized subject to the unit length constraint $\|\Delta\mathbf{q}\| = 1$. In the original approach of Horaud & Dornaika, this constraint is enforced by adding a penalty term $\xi_{\text{pen}} \cdot (1 - \Delta\mathbf{q}^\mathsf{T}\Delta\mathbf{q})^2$ with $\xi_{\text{pen}} = 2 \cdot 10^6$ which might dominate the outcome. Other recommended options are to use either constrained nonlinear optimization methods (see C.3.4) if available, to enforce the constraint by normalizing $\Delta\mathbf{q}$ within the error function, or to apply the reduced quaternion parametrization from eq. (2.35) when unconstrained NLLS methods such as the Levenberg-Marquardt algorithm are used.

The implementations of these methods are denoted as $\texttt{angle}_{\texttt{NL}}$, $\texttt{qvec}_{\texttt{NL}}$ and $\texttt{quat}_{\texttt{NL}}$ for unit weights resp. $\texttt{W-angle}_{\texttt{NL}}$, $\texttt{W-qvec}_{\texttt{NL}}$ and $\texttt{W-quat}_{\texttt{NL}}$ when statistical weights are used as described in the next section.[12]

3.4.5 Weighting Rotation and Translation Errors

As Strobl & Hirzinger emphasize in [SH06], careful weighting of the immanently disproportionate error functions is crucial for nonlinear optimization. In the original approach of Horaud & Dornaika, heuristic weights $\xi_{\text{rot}} = \xi_{\text{pos}} = 1$ are used without further indication how to select these. Both Strobl & Hirzinger [SH06] and Ikits [Iki00] derive the weights from the error distribution of the input poses, leading to robust and statistically meaningful error functions.

Based on the simplified error model from Sec. 2.5, we compute weights $\xi_{\text{rot}} = \kappa/(\sigma^2_{\alpha_\varepsilon} + \sigma^2_{\alpha'_\varepsilon})$ and $\xi_{\text{pos}} = 1/(\sigma^2_{\boldsymbol{t}_\varepsilon} + \sigma^2_{\boldsymbol{t}'_\varepsilon})$ where $\sigma_{\alpha_\varepsilon}$, $\sigma_{\alpha'_\varepsilon}$ and $\sigma_{\boldsymbol{t}_\varepsilon}, \sigma_{\boldsymbol{t}'_\varepsilon}$ are

[12] The prefix `W-` indicates weighted methods, the subscript `NL` refers to the nonlinear error functions and sets these methods apart from direct methods.

the assumed standard deviations of the magnitude of rotation and translation errors for both cameras. This is reasonable since the magnitude of the rotation term residuals in eq. (3.35) is approximately linear proportional to the input rotation error in the general case. Although the translation term residuals depend in fact on both the input rotation and translation error, their magnitude can be approximated by the translationial error in the general case.[13] The factor κ results from the actual rotation distance measure d_{rot} used in nonlinear optimization and gauges the rotation term residuals towards the angle distance. Hence, we have $\kappa = 1$ for $d_{\angle}$, $\kappa = 4$ for d_{quat} and $\kappa = \frac{1}{2}$ for d_{chord} (see Sec. 2.4).

For arbitrary covariance matrices $\boldsymbol{\Sigma}_{\boldsymbol{\mu}_k}$, $\boldsymbol{\Sigma}_{\boldsymbol{\mu}'_k}$ describing the uncertainty of the input pose parameters $\boldsymbol{\mu}_k$, $\boldsymbol{\mu}'_k$, we can use propagation of uncertainty as described in A.5 to approximate σ_{α_k} and $\sigma_{\boldsymbol{t}_k}$ for each motion k individually, resulting in weights $\xi_{\text{rot}k} = \kappa/(\sigma^2_{\alpha_k} + \sigma^2_{\alpha'_k})$ and $\xi_{\text{pos}_k} = 1/(\sigma^2_{\boldsymbol{t}_k} + \sigma^2_{\boldsymbol{t}'_k})$ for each summand in eq. (3.35). In practical applications it is reasonable to use lower bounds for σ to prevent singularities. We use 0.01° and 0.001 m which provides realistic lower bounds for the accuracy of pose estimation.

Weighting linear combined estimation Although weighting the residuals of the rotation and translation term has been considered as crucial for nonlinear optimization, no attempts have been made so far to incorporate weights into linear methods for combined estimation, i. e., methods `Tmat` and `dualquat` from Sec. 3.4.3, where the same problem is immanent.

On account of this, we propose weighted versions of both methods. In `Tmat`, the linear equations relating to the rotation matrix part are weighted with $\sqrt{\xi_{\text{rot}}} = \frac{1}{\sqrt{2}}\,\sigma_\alpha$, paying respect to the chordal metric. The linear equations relating to the translation part are weighted with $\sqrt{\xi_{\text{pos}}}$. In `dualquat`, the linear equations relating to the real quaternion part are weighted with $\sqrt{\xi_{\text{rot}}} = 2\,\sigma_\alpha$. Since the residuals of the dual quaternion part are within the same order of magnitude as translational errors, they are weighted with $\sqrt{\xi_{\text{pos}}}$ as well. The modified methods are denoted as `W-Tmat` and `W-dualquat`.

[13] These statements originally made in [TL89] are experimentally verified in Sec. 3.8.1.

We will illustrate the importance of weighting rotation and translation terms in the experiments in Sec. 3.8.

3.4.6 Nonlinear Weighted Total Least Squares Solution

All methods described so far minimize an error function based on pose observations and predictions where either the poses of the first camera are considered as observations and poses of the second camera are used as model parameters or vice versa. Hence, it is not possible to deduce proper uncertainty measures for the resulting parameters based on *both* cameras' pose measurement uncertainties. In the classical hand-eye calibration approach, pose uncertainties of either the hand or the eye are assumed as insignificant, depending on the context.

A variant of eq. (3.35) that has been rarely examined in the literature on hand-eye calibration so far aims at simultaneous optimization of rigidly coupled camera motion and eye-to-eye transformation parameters with respect to the observed camera poses.[14] This problem is formalized by the following nonlinear weighted total least squares problem that takes errors for pose measurements of both cameras into account and weights the residuals resulting from observed and predicted poses properly with respect to the Mahalanobis distance:[15]

$$\min_{\Delta\boldsymbol{\mu},\hat{\boldsymbol{\mu}}_{1,\ldots,m}} \sum_{k=1}^{m} \|\hat{\boldsymbol{\mu}}_k - \boldsymbol{\mu}_k\|^2_{\boldsymbol{\Sigma}_{\boldsymbol{\mu}_k}} + \|\hat{\boldsymbol{\mu}}'_k - \boldsymbol{\mu}'_k\|^2_{\boldsymbol{\Sigma}_{\boldsymbol{\mu}'_k}} = \|\boldsymbol{f}(\Delta\boldsymbol{\mu}, \hat{\boldsymbol{\mu}}_1, \ldots, \hat{\boldsymbol{\mu}}_m)\|^2_{\boldsymbol{\Sigma}_{\boldsymbol{\mu}_k}} \quad (3.36)$$

where $\boldsymbol{\mu}_k, \boldsymbol{\mu}'_k \in \mathbb{R}^\mu$ are the observed pose parameters, $\boldsymbol{\Sigma}_{\boldsymbol{\mu}_k}, \boldsymbol{\Sigma}_{\boldsymbol{\mu}'_k} \in \mathbb{R}^{\mu\times\mu}$ are the covariance matrices of pose measurement errors, and $\hat{\boldsymbol{\mu}}'_k$ are the parameters of the predicted poses $\mathbf{T}^{-1}_{\Delta\boldsymbol{\mu}}\mathbf{T}_{\hat{\boldsymbol{\mu}}_k}\mathbf{T}_{\Delta\boldsymbol{\mu}}$. The solution of eq. (3.36) is considered as *optimal* with respect to the Gaussian error model.

As a result, this approach provides a proper approximation to the covariance matrix of estimated parameter errors based on the covariances of

[14] To our knowledge, the total least squares formulation of the hand-eye calibration problem has only been considered by Ikits [Iki00].

[15] Note that $\boldsymbol{\Sigma}_{\boldsymbol{\mu}}^{-1}$ is replaced by the pseudoinverse $\boldsymbol{\Sigma}_{\boldsymbol{\mu}}^{\dagger}$ when $\boldsymbol{\Sigma}_{\boldsymbol{\mu}}$ is singular.

measurement errors:

$$\mathbf{J}_{\Delta\mu_\varepsilon} = (\mathbf{J}_f^\mathsf{T}\boldsymbol{\Sigma}_{\mu_\varepsilon}^{-1}\mathbf{J}_f)^{-1} \tag{3.37}$$

where $\boldsymbol{\Sigma}_{\mu_\varepsilon}$ is the block-diagonal matrix consisting of $\boldsymbol{\Sigma}_{\mu_1}, \ldots, \boldsymbol{\Sigma}_{\mu_m}, \boldsymbol{\Sigma}_{\mu'_1}, \ldots, \boldsymbol{\Sigma}_{\mu'_m}$ and $\mathbf{J}_f$ is the Jacobian matrix of the error function evaluated at the solution $(\Delta\boldsymbol{\mu}, \hat{\boldsymbol{\mu}}_1, \ldots, \hat{\boldsymbol{\mu}}_m)$ (see C.3.3 for details).

As a drawback, the number of parameters in eq. (3.36) is increased by $m\mu$ as compared to eq. (3.36). For medium-sized problems with 10 to 20 images, the resulting NLLS problem is still tractable with standard methods.

We implemented this approach using the error function from $\texttt{quat}_{\texttt{NL}}$. To avoid parameter constraints, the reduced quaternion parametrization defined in eq. (2.35) is applied. Propagation of uncertainty is used to transform the covariance matrices $\boldsymbol{\Sigma}_{\mu_k}, \boldsymbol{\Sigma}_{\mu'_k}$ of observed poses into the reduced quaternion parameter form. This method is briefly denoted as $\texttt{W-TLS}_{\texttt{NL}}$ in the following.

3.5 Extended Eye-to-Eye Calibration

In the previous sections we assumed that the absolute scale of all translation vectors is known. Although this applies to the classical hand-eye calibration problem where poses of the hand are computed from kinematic chains and poses of the camera are computed from images of a calibration pattern with known dimensions, the same assumption cannot be made for the eye-to-eye calibration problem. Given that camera poses are computed from multi-view constraints between images or from the epipolar geometry of image pairs (see Chapter 4), all translations vectors are measured only up to a global scale or even up to an individual scale for each pose transformation.

3.5.1 Translation Estimation with Unknown Scale

Assume that the absolute scales of translations for both cameras are not known, i. e., we have measured translations $\hat{\boldsymbol{t}}_k, \hat{\boldsymbol{t}}'_k$ that are related to the absolute translations $\boldsymbol{t}_k, \boldsymbol{t}'_k$ by $\lambda_k \hat{\boldsymbol{t}}_k = \boldsymbol{t}_k$ and $\lambda'_k \hat{\boldsymbol{t}}'_k = \boldsymbol{t}'_k$ with unknown scale factors $\lambda_k, \lambda'_k \in \mathbb{R}_{>0}$.

Translation constraint with unknown scales Given translations with unknown absolute scales, λ_k and λ'_k are introduced as additional parameters in eq. (3.14):

$$(\mathbf{R}_k - \mathbf{I})\Delta\boldsymbol{t} = \lambda'_k \Delta\mathbf{R}\hat{\boldsymbol{t}}'_k - \lambda_k \hat{\boldsymbol{t}}_k \tag{3.38}$$

Fixing the inherent absolute scale ambiguity, we obtain:

$$(\mathbf{R}_k - \mathbf{I})\Delta\boldsymbol{t} \sim \Delta\lambda_k \Delta\mathbf{R}\hat{\boldsymbol{t}}'_k - \hat{\boldsymbol{t}}_k \tag{3.39}$$

or equivalently

$$(\mathbf{R}_k - \mathbf{I})\Delta\boldsymbol{t} \times (\Delta\lambda_k \Delta\mathbf{R}\hat{\boldsymbol{t}}'_k - \hat{\boldsymbol{t}}_k) = \mathbf{0}$$

with $\Delta\lambda_k = \lambda'_k / \lambda_k$. Obviously, the eye-to-eye translation $\Delta\boldsymbol{t}$ can only be recovered up to scale from the resulting equation system:

$$\min_{\Delta\hat{\boldsymbol{t}}, \Delta\lambda_1, \ldots, \Delta\lambda_m} \sum_{k=1}^{m} \| [\Delta\lambda_k \Delta\mathbf{R}\hat{\boldsymbol{t}}'_k - \hat{\boldsymbol{t}}_k]_\times (\mathbf{R}_k - \mathbf{I})\Delta\hat{\boldsymbol{t}} \|^2 \quad \text{s.t. } \|\Delta\hat{\boldsymbol{t}}\| = 1 \tag{3.40}$$

This case occurs when all relative camera poses are computed from pairwise epipolar constraints, providing only the direction vector for each relative translation. However, since this problem formulation is rather hard to cope with, we will consider relaxations of this problem first.

Eye-to-eye calibration with absolute reference scale Assuming that the translations of the first camera are known with absolute scale, eq. (3.38) becomes:

$$(\mathbf{I} - \mathbf{R}_k)\Delta\boldsymbol{t} + \lambda'_k \Delta\mathbf{R}\hat{\boldsymbol{t}}'_k = \boldsymbol{t}_k \tag{3.41}$$

Solving this linear equation system w.r.t. $\Delta\boldsymbol{t}$ and $\lambda'_1, \ldots, \lambda'_m$ yields the eye-to-eye translation $\Delta\boldsymbol{t}$ with absolute scale.

The previous case can be reduced to this case if some calibration pattern is used to measure absolute poses of the first camera.

Eye-to-eye calibration with unknown constant scale Assume that all relative motions are known up to an unknown scale which is constant for all motions, i. e., we have translations $\lambda\hat{\boldsymbol{t}}_k = \boldsymbol{t}_k$ and $\lambda'\hat{\boldsymbol{t}}'_k = \boldsymbol{t}'_k$ with constant scale factors λ and λ'. In this case we can solve eq. (3.38) up to scale by including one additional unknown $\Delta\lambda = \lambda'/\lambda$:

$$(\mathbf{R}_k - \mathbf{I})\Delta\boldsymbol{t} = \lambda'\Delta\mathbf{R}\hat{\boldsymbol{t}}'_k - \lambda\hat{\boldsymbol{t}}_k \tag{3.42}$$

or equivalently

$$(\mathbf{I} - \mathbf{R}_k)\Delta\hat{\boldsymbol{t}} + \Delta\lambda\Delta\mathbf{R}\hat{\boldsymbol{t}}'_k = \hat{\boldsymbol{t}}_k$$

for all $k = 1, \ldots, m$ where $\lambda\Delta\hat{\boldsymbol{t}} = \Delta\boldsymbol{t}$. Hence, if the absolute constant scale λ of the first camera is known, the absolute scale of the eye-to-eye translation can be recovered as well. Otherwise, its scale is recovered w.r.t. the reference coordinate frame $\mathcal{C}$ of the first camera.

This case occurs when multi-view methods such as Structure from Motion are used to compute the relative poses for both cameras, resulting in 3d reconstructions with unknown but fixed scale. Hence, if absolute pose estimation methods are used for at least one of the cameras (e. g., by mean of calibration patterns or markers), the absolute scale of $\Delta\boldsymbol{t}$ can be recovered. For practical applications we can assume that the relative scale $\Delta\lambda$ between the first and second camera is constant for all considered motions.

3.5.2 Extending Eye-to-Eye Calibration Methods

The parameter $\Delta\lambda$ can be interpreted as an isometric scaling factor between the reference coordinate frames $\mathcal{C}$ and $\mathcal{C}'$ of both cameras. There are two distinct methods to incorporate scale estimation into existing eye-to-eye

calibration methods. For all decoupled methods, the translation equation (3.30) is simply replaced by eq. (3.42), introducing $\Delta\lambda$ explicitly as an additional parameter:

$$\min_{\Delta\hat{\boldsymbol{t}},\Delta\lambda} \sum_{k=1}^{m} \|(\mathbf{R}_k - \mathbf{I})\Delta\hat{\boldsymbol{t}} - (\Delta\lambda\Delta\mathbf{R}\boldsymbol{t}'_k - \boldsymbol{t}_k)\|^2 \tag{3.43}$$

Since the eye-to-eye transformation is defined by a similarity transformation $\Delta\mathbf{S} = [\Delta\lambda\Delta\mathbf{R} \mid \Delta\boldsymbol{t}]$ instead of a Euclidean transformation $\Delta\mathbf{T}$, the eye-to-eye rotation $\Delta\mathbf{R}$ is replaced by a scaled rotation $\Delta\lambda\Delta\mathbf{R}$. Hence, an alternative approach for combined estimation methods where rotation is parametrized by a unit quaternion is to omit the unit length constraint in the translation term. The scaled quaternion $\Delta\mathbf{q}_\lambda = \sqrt{\Delta\lambda}\Delta\mathbf{q}$ represents the scaled rotation matrix $\Delta\lambda\Delta\mathbf{R}$. For example, extending the nonlinear method $\texttt{quat}_{\texttt{NL}}$ provides the error function:

$$\min_{\Delta\mathbf{q}_\lambda,\Delta\boldsymbol{t}} \sum_{k=1}^{m} \| \tfrac{\mathbf{q}_k\cdot\Delta\mathbf{q}_\lambda - \Delta\mathbf{q}_\lambda\cdot\mathbf{q}'_k}{\|\Delta\mathbf{q}_\lambda\|} \|^2 + \|(\mathbf{R}_k - \mathbf{I})\Delta\boldsymbol{t} - \Delta\mathbf{q}_\lambda \cdot \boldsymbol{t}'_k \cdot \Delta\bar{\mathbf{q}}_\lambda + \boldsymbol{t}_k\|^2 \tag{3.44}$$

leading to an unconstrained nonlinear least squares problem, denoted as $\texttt{quat}^{+}_{\texttt{NL}}$.[16]

Schmidt et al. [SVN05] use a similar approach to extend the dual quaternion based approach `dualquat`, encoding the relative scale by the norm of a dual quaternion $\Delta\check{\mathbf{q}}_\lambda = \Delta\lambda\Delta\check{\mathbf{q}}$. The resulting error function becomes nonlinear:

$$\min_{\Delta\mathbf{q}_\lambda,\Delta\mathbf{p}} \sum_{k=1}^{m} \| \tfrac{\mathbf{q}_k\cdot\Delta\mathbf{q}_\lambda - \Delta\mathbf{q}_\lambda\cdot\mathbf{q}'_k}{\|\Delta\mathbf{q}_\lambda\|} \|^2 + \|\mathbf{q}\cdot\Delta\mathbf{p} + \tfrac{\mathbf{p}\cdot\Delta\mathbf{q}_\lambda}{\|\Delta\mathbf{q}_\lambda\|} - \Delta\mathbf{q}_\lambda\cdot\mathbf{p}' - \Delta\mathbf{p}\cdot\mathbf{q}'_k\|^2 \tag{3.45}$$

subject to $\Delta\mathbf{q}_\lambda^\intercal\Delta\mathbf{p} = 0$. This method is denoted as $\texttt{dualquat}^{+}_{\texttt{NL}}$.

Andreff et al. extend their combined approach in [AHE01] in a similar way. They consider the inverse scale $\Delta\lambda' = \lambda/\lambda'$ instead and solve for $(\mathbf{R}_k - \mathbf{I})\Delta\hat{\boldsymbol{t}} - (\Delta\mathbf{R}\boldsymbol{t}'_k - \Delta\lambda'\boldsymbol{t}_k)$.[17] However, the authors fail to notice that

[16] The superscript + indicates extended methods.

[17] This is due to the fact that Andreff et al. associate $\mathcal{C}$ with a camera providing poses with

their original method is already capable of estimating a relative scale implicitly since eq. (3.32) is solved without enforcing scale constraints on the estimated rotation matrix $\Delta\mathbf{R}$, yielding a scaled matrix $\Delta\mathbf{R}_\lambda = \Delta\lambda\mathbf{R}$ depending on the input data. The scale is implicitly removed from $\Delta\mathbf{R}_\lambda$ in the final orthonormalization step. Hence, we will denote this method by $\texttt{Tmat}^+$ in the context of extended eye-to-eye calibration although $\texttt{Tmat}^+$ is in fact identical to `Tmat`.

For weighting the rotation and translation error terms, the same factors as described in Sec. 3.4.5 can be used, providing methods $\texttt{W-Tmat}^+$, $\texttt{W-quat}^+_{\texttt{NL}}$, $\texttt{W-dualquat}^+_{\texttt{NL}}$ etc.

3.5.3 Relative Scale from Motion Pitch

In extended eye-to-eye calibration, the unknown scale factors are estimated from the translation constraint, either by modeling them explicitly by additional parameters or implicitly by estimating a scaled eye-to-eye rotation. Next, we will show that the relative scale factors $\Delta\lambda_k$ can also be observed in single motions using the pitch constraint.

Lemma 3.9 (Scale Constraint)**.** *Given rigidly coupled motions* $\mathbf{T}$ *and* $\mathbf{T}'$ *where translations are only known up to unknown scales, i. e.,* $\lambda\hat{\boldsymbol{t}} = \boldsymbol{t}$ *and* $\lambda'\hat{\boldsymbol{t}}' = \boldsymbol{t}'$*, the following constraint holds:*

$$\lambda\,\boldsymbol{r}^\mathsf{T}\hat{\boldsymbol{t}} = \lambda'\,\boldsymbol{r}'^\mathsf{T}\hat{\boldsymbol{t}}' \quad \textit{i.e.,} \quad \frac{\lambda'}{\lambda} = \frac{\boldsymbol{r}^\mathsf{T}\hat{\boldsymbol{t}}}{\boldsymbol{r}'^\mathsf{T}\hat{\boldsymbol{t}}'} = \frac{\hat{p}}{\hat{p}'} \tag{3.46}$$

where $p = \boldsymbol{r}^\mathsf{T}\boldsymbol{t}$ *is the pitch of a motion with rotation axis* $\boldsymbol{r}$ *and translation* $\boldsymbol{t}$*.*

This constraint is motivated by the intuition that translation along the rotation axis is caused only by translation of the rigidly coupled cameras. For absolute motion, the magnitude of this translation (i. e., the pitch) is equal, for scaled motion the relative scale of translations equals the relative scale of the pitch values likewise.

unknown scale via Structure from Motion, and $\mathcal{C}'$ with a robotic gripper providing poses with known absolute scale.

Hence, given that either the absolute translation length of the first or second camera is known, we are able to recover the absolute scale of the other camera's translation. Otherwise, at least the relative scale $\Delta\lambda_k = \lambda'_k/\lambda_k$ of the second camera's translation with respect to the first can be retrieved for each motion $k = 1, \dots, m$ unless $p_k = 0$ or $p'_k = 0$, i. e., there is either no translation or only planar motion. Since the value $\Delta\lambda_k$ depends only on the measured poses $\mathbf{T}, \mathbf{T}'$, it can also be interpreted as a measurement quantity.

Using scale measurements in eye-to-eye calibration Using eq. (3.46), the relative scale parameters $\Delta\lambda_k$ can be replaced in eq. (3.40) for each motion where translation is not purely orthogonal to the rotation axis, leading to the following constrained linear least squares problem:

$$\min_{\Delta\hat{\boldsymbol{t}}} \sum_{k=1}^{m} \| \left[\hat{p}_k \Delta\mathbf{R}\hat{\boldsymbol{t}}'_k - \hat{p}'_k \hat{\boldsymbol{t}}_k \right]_\times (\mathbf{R}_k - \mathbf{I}) \Delta\hat{\boldsymbol{t}} \|^2 \quad \text{s.t. } \|\Delta\hat{\boldsymbol{t}}\| = 1 \tag{3.47}$$

Estimation of unknown constant scale Considering the situation of eq. (3.42) i. e., there is an unknown constant scale factor $\Delta\lambda$ between coordinate frames $\mathcal{C}$ and $\mathcal{C}'$. Applying eq. (3.46), $\Delta\lambda$ can be estimated from scale measurements only via the following linear least squares problem:

$$\min_{\Delta\lambda} \sum_{k=1}^{m} (\Delta\lambda \hat{p}'_k - \hat{p}_k)^2 \quad \Rightarrow \quad \Delta\lambda = \frac{\boldsymbol{p}^\intercal \boldsymbol{p}'}{\boldsymbol{p}'^\intercal \boldsymbol{p}'} \tag{3.48}$$

where $\boldsymbol{p} = (\hat{p}_1, \dots, \hat{p}_m)$, $\boldsymbol{p}' = (\hat{p}'_1, \dots, \hat{p}'_m)$.

Note that eq. (3.48) is singular for planar motion. However, we will derive a similar equation system for planar motion pairs in Sec. 3.6.2.

Estimation of scale uncertainty Consider that a covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{r},\boldsymbol{t}}$ for the rotation axis $\boldsymbol{r}$ and translation vector $\boldsymbol{t}$ of some pose is given. The variance σ_p^2 of the motion pitch $p = \boldsymbol{r}^\intercal \boldsymbol{t}$ can be approximated via error

propagation (see also eq. (A.20) in A.5):

$$p = \boldsymbol{r}^\mathsf{T}\boldsymbol{t} \quad \Rightarrow \quad \mathbf{J}_p = (\boldsymbol{t}^\mathsf{T} \;\; \boldsymbol{r}^\mathsf{T}) \quad \text{i.e.,} \quad \sigma_p^2 = \mathbf{J}_p \boldsymbol{\Sigma}_{\boldsymbol{r},\boldsymbol{t}} \mathbf{J}_p^\mathsf{T} \tag{3.49}$$

where $\mathbf{J}_p$ is the Jacobian matrix of $p = \boldsymbol{r}^\mathsf{T}\boldsymbol{t}$ evaluated at $(\boldsymbol{r}, \boldsymbol{t})$.

Given variance approximations $\sigma^2_{\hat{p}_k}, \sigma^2_{\hat{p}'_k}$ for each motion pitch $\hat{p}_k$, $\hat{p}'_k$ of both cameras, the uncertainty of the resulting relative scale factor $\Delta\lambda$ is approximated via error propagation from eq. (3.48) as:

$$\sigma^2_{\Delta\lambda} = \mathbf{J}_{\Delta\lambda} \; \mathrm{diag}(\sigma^2_{\hat{p}_1}, \ldots, \sigma^2_{\hat{p}_m}, \sigma^2_{\hat{p}'_1}, \ldots, \sigma^2_{\hat{p}'_m}) \; \mathbf{J}_{\Delta\lambda}^\mathsf{T} \tag{3.50}$$

where

$$\mathbf{J}_{\Delta\lambda} = \left(\frac{\boldsymbol{p}'^\mathsf{T}}{\boldsymbol{p}'^\mathsf{T}\boldsymbol{p}'} \qquad \frac{\boldsymbol{p}^\mathsf{T}}{\boldsymbol{p}'^\mathsf{T}\boldsymbol{p}'} - \frac{(\boldsymbol{p}^\mathsf{T}\boldsymbol{p}')\boldsymbol{p}'^\mathsf{T}}{(\boldsymbol{p}'^\mathsf{T}\boldsymbol{p}')^2} \right)$$

leading to the closed form solution:

$$\sigma^2_{\Delta\lambda} = \tfrac{1}{(\boldsymbol{p}'^\mathsf{T}\boldsymbol{p}')^2} \sum_{k=1}^{m} \hat{p}'^2_k \sigma^2_{\hat{p}_k} + (\hat{p}_k - \tfrac{\boldsymbol{p}^\mathsf{T}\boldsymbol{p}'}{\boldsymbol{p}'^\mathsf{T}\boldsymbol{p}'} \hat{p}'_k)^2 \sigma^2_{\hat{p}'_k}$$

This measure can be evaluated to access the expected accuracy of the estimated scale factor $\Delta\lambda$. For motion close to the singular case, the approximated uncertainty $\sigma^2_{\Delta\lambda}$ will approach $+\infty$.

3.6 Partial Solution from Critical Motions

In the following we will provide specialized solutions for critical motion configurations auch as pure translation or planar motion.[18] The general approach described above cannot be applied here since the rigid coupling equations degenerate for these cases and provide no unique solution. Hence, additional constraints are introduced according to the specific motion model.

[18] This is also called "partial calibration" in [AHE01] since only a subset of eye-to-eye parameters can be obtained from the resulting singular equation systems.

3.6.1 Pure Translation

In the case of pure translations, i. e., $\mathbf{R}_k = \mathbf{R}'_k = \mathbf{I}$ for all $k = 1, \ldots, m$, eq. (3.14) degenerates to:

$$\boldsymbol{t}_k = \Delta\mathbf{R}\boldsymbol{t}'_k \quad \text{for } k = 1, \ldots, m \tag{3.51}$$

The eye-to-eye translation $\Delta\boldsymbol{t}$ cannot be recovered from relative poses only in this case. However, the eye-to-eye rotation $\Delta\mathbf{R}$ can be recovered from eq. (3.51) via estimation of the relative rotation between two sets of 3d vectors (see A.3.2), given that not all translations are parallel.

Although pure translations occur in the classical hand-eye calibration scenario due to the controlled motion of the robotic arm, they are rather unlikely in eye-to-eye calibration, at least for the case of freely moving camera systems. However, we will consider pure translations in the planar motion case since they are most likely to appear in the context of camera rigs mounted onto vehicles subject to Ackerman steering.

Combining eq. (3.16) and eq. (3.51) yields a unique solution for the eye-to-eye rotation $\Delta\mathbf{R}$ if one non-zero rotation pair $(\mathbf{R}_1, \mathbf{R}'_1)$ and one pure translation pair $(\boldsymbol{t}_2, \boldsymbol{t}'_2)$ is given, provided that the translation of the latter is not parallel to the rotation axis of the former ($\boldsymbol{t}_2 \nparallel \boldsymbol{r}_1$ resp. $\boldsymbol{t}'_2 \nparallel \boldsymbol{r}'_1$):

$$\boldsymbol{r}_1 = \Delta\mathbf{R}\boldsymbol{r}'_1 \quad \text{and} \quad \boldsymbol{t}_2 = \Delta\mathbf{R}\boldsymbol{t}'_2 \tag{3.52}$$

Equation (3.52) is again solved via relative rotation estimation (see A.3.2).

For extended eye-to-eye calibration, pure translations can be used additional to the scale constraint (3.46) to estimate the relative scale $\Delta\lambda_k$ between cameras due to:

$$\lambda_k\,\hat{\boldsymbol{t}}_k = \lambda'_k\Delta\mathbf{R}\hat{\boldsymbol{t}}'_k \quad \text{i. e.,} \quad \Delta\lambda_k = \frac{\lambda'_k}{\lambda_k} = \frac{\|\hat{\boldsymbol{t}}_k\|}{\|\hat{\boldsymbol{t}}'_k\|} \tag{3.53}$$

3.6.2 Planar Motion

Given are two planar motion pairs $\mathbf{T}_k$, $\mathbf{T}'_k$, $k \in \{1,2\}$ where the common plane of motion has the normal vector $\boldsymbol{n} = \boldsymbol{r}_1 = \boldsymbol{r}_2$ in the coordinate frame $\mathcal{C}$. As described above, eye-to-eye calibration is underconstrained for planar motion since all rotation axes are parallel. However, the relative rotation $\Delta\mathbf{R}$ can be determined uniquely and the relative translation $\Delta\boldsymbol{t}$ can be determined up to an unknown translation along the plane normal $\boldsymbol{n}$ if one of the following sets of conditions is fulfilled [AHE99]:

- ▷ One motion is a pure translation, the other contains non-zero rotation. The pure translation is assumed to be non-parallel to the rotation axis.
- ▷ Both motions contain non-zero rotation and the motions satisfy the constraint $(\mathbf{I} - \mathbf{R}_1)\boldsymbol{t}_2 \nparallel (\mathbf{I} - \mathbf{R}_2)\boldsymbol{t}_1$.

Decoupled solution First, we will describe direct methods to recover the relative rotation uniquely for both cases assuming ideal planar motion. Consider w.l.o.g. that the first motion contains non-zero rotation and the second motion is a pure translation. Since both motions occur within the same plane, $\boldsymbol{t}_2 \perp \boldsymbol{r}_1$ resp. $\boldsymbol{t}'_2 \perp \boldsymbol{r}'_1$ holds. Provided that rotation axes and translation directions are non-parallel (which is the case for planar motion), the relative rotation can be estimated from eq. (3.52). Relative scale $\Delta\lambda$ can be determined from the translation length as in eq. (3.53).

For the second, case assume that both motions have non-zero rotation and define vectors $\boldsymbol{d} = (\mathbf{I} - \mathbf{R}_1)\boldsymbol{t}_2 - (\mathbf{I} - \mathbf{R}_2)\boldsymbol{t}_1$ and $\boldsymbol{d}'$ likewise. We will show in the following that $\boldsymbol{d} = \Delta\mathbf{R}\boldsymbol{d}'$ holds for planar motion.

$$
\begin{aligned}
\Delta\mathbf{R}\boldsymbol{d}' &= \Delta\mathbf{R}((\mathbf{I} - \mathbf{R}'_1)\boldsymbol{t}'_2 - (\mathbf{I} - \mathbf{R}'_2)\boldsymbol{t}'_1) \\
&= \Delta\mathbf{R}\boldsymbol{t}'_2 - \Delta\mathbf{R}\mathbf{R}'_1\boldsymbol{t}'_2 - \Delta\mathbf{R}\boldsymbol{t}'_1 + \Delta\mathbf{R}\mathbf{R}'_2\boldsymbol{t}'_1 \\
&= \Delta\mathbf{R}\boldsymbol{t}'_2 - \mathbf{R}_1\Delta\mathbf{R}\boldsymbol{t}'_2 - \Delta\mathbf{R}\boldsymbol{t}'_1 + \mathbf{R}_2\Delta\mathbf{R}\boldsymbol{t}'_1 \\
&= (\mathbf{I} - \mathbf{R}_1)\Delta\mathbf{R}\boldsymbol{t}'_2 - (\mathbf{I} - \mathbf{R}_2)\Delta\mathbf{R}\boldsymbol{t}'_1 \\
&= (\mathbf{I} - \mathbf{R}_1)(\boldsymbol{t}_2 - (\mathbf{I} - \mathbf{R}_2)\Delta\boldsymbol{t}) - (\mathbf{I} - \mathbf{R}_2)(\boldsymbol{t}_1 - (\mathbf{I} - \mathbf{R}_1)\Delta\boldsymbol{t}) \\
&= \underbrace{(\mathbf{I} - \mathbf{R}_1)\boldsymbol{t}_2 - (\mathbf{I} - \mathbf{R}_2)\boldsymbol{t}_1}_{=\boldsymbol{d}} +
\end{aligned}
$$

$$\underbrace{(\mathbf{I}-\mathbf{R}_1)(\mathbf{I}-\mathbf{R}_2)\Delta\boldsymbol{t}-(\mathbf{I}-\mathbf{R}_2)(\mathbf{I}-\mathbf{R}_1)\Delta\boldsymbol{t}}_{=\mathbf{0}} \quad = \quad \boldsymbol{d}$$

The last part holds since $\mathbf{R}_1$, $\mathbf{R}_2$ are rotations around the same rotation axis and can hence be commuted. So the second case can be reduced to the first case using $\boldsymbol{d}$, $\boldsymbol{d}'$ as virtual pure translations and either the first or second rotation to determine $\Delta\mathbf{R}$ from eq. (3.52), provided that $\boldsymbol{d}, \boldsymbol{d}' \neq \mathbf{0}$.

Note that this constraint can also be used for relative scale estimation in extended eye-to-eye calibration from planar motion from eq. (3.53).

Once the relative rotation $\Delta\mathbf{R}$ is known, the relative translation can be estimated up to an unknown translation along $\boldsymbol{n}$ from eq. (3.30) as mentioned before. To solve the underconstrained linear equation system eq. (3.30) with respect to $\Delta\boldsymbol{t}$, add the constraint $\Delta\boldsymbol{t}^\top\boldsymbol{n} = 0$ in order to restrict the relative translation vector to the plane of motion (resp. $\Delta\boldsymbol{t}^\top\boldsymbol{n} = \Delta h$ when the translational offset perpendicular to the plane of motion is known, e. g., from localization of the ground plane or according to the construction of the camera rig).

Combined solution In the decoupled method described above, parallel rotation axes are assumed. In case of close to planar motion, this approximation is still valid to provide an initial solution that can be refined using the combined nonlinear methods. However, constraints of the form $\Delta\mathbf{R}\boldsymbol{d}' = \boldsymbol{d}$ should be omitted in combined nonlinear optimization for non-ideal planar motion. Instead, only the constraint $\Delta\boldsymbol{t}^\top\boldsymbol{n} = \Delta h$ is added to the translation error terms as in decoupled translation estimation to prevent singularities. We also recommend to weight this residual with respect to the assumed accuracy of common motion plane localization.

3.6.3 Transformation into Common Reference Plane

In [EG10], we described how to facilitate eye-to-eye calibration from planar motion by reducing the parameter space explicitly to the common plane of motion, assuming that the location of the ground plane within each

camera's reference coordinate frame is known. Pagel & Willersinn use a similar technique in [PW11]. This approach is based on Lemma 3.6 in Sec. 3.3.

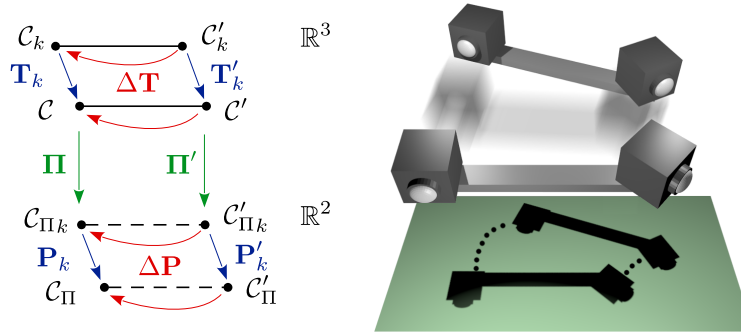

Figure 3.7. Local coordinate frames and planar motion of two rigidly coupled cameras projected into the common plane of motion.

Assume that the ground plane is described by its normal $\boldsymbol{n} \in \mathbb{S}^2$ and height $h \in \mathbb{R}$ in the first camera coordinate frame $\mathcal{C}$ resp. by $\boldsymbol{n}'$, h' in the second camera coordinate frame $\mathcal{C}'$. For the first camera, a projection $\boldsymbol{\Pi} : \mathbb{R}^3 \to \mathbb{R}^2$ from 3d space to the 2d space of the ground plane is defined by:

$$\boldsymbol{\Pi}(\boldsymbol{X}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{R}_\Pi \boldsymbol{X} \tag{3.54}$$

where $\mathbf{R}_\Pi$ is any rotation matrix mapping $\boldsymbol{n}$ to $\boldsymbol{e}_\mathrm{y}$.

The inverse mapping, lifting a 2d point $\boldsymbol{x} = (x, y)$ in the ground plane back into 3d space, is given by:

$$\boldsymbol{\Pi}^{-1}(\boldsymbol{x}) = \mathbf{R}_\Pi^\mathsf{T} \begin{pmatrix} x \\ h \\ y \end{pmatrix} \tag{3.55}$$

$\boldsymbol{\Pi}'$, $\boldsymbol{\Pi}'^{-1}$, and $\mathbf{R}'_\Pi$ are defined for the second camera likewise.

Rigidly coupled motions $\mathbf{T}_k$, $\mathbf{T}'_k$ of the cameras are transformed into the

respective 2d Euclidean transformations $\mathbf{P}_k$, $\mathbf{P}'_k$ within the ground plane, i. e., $\mathbf{P}_k$ consists of the 2d translation vector $\boldsymbol{\tau}_k = \boldsymbol{\Pi}(\boldsymbol{t}_k)$ and 2d rotation $\mathbf{R}_{\alpha_k}$ with rotation angle α_k assuming that $\mathbf{R}_k = \mathbf{R}_{\boldsymbol{n},\alpha_k}$.

The planar eye-to-eye transformation is described by a 2d translation vector $\Delta\boldsymbol{\tau}$ and 2d rotation by angle $\Delta\alpha$. For the case of scaled coordinate frames, an additional isometric scaling factor $\Delta\lambda$ is needed to define the eye-to-eye similarity transformation. The translational eye-to-eye constraint eq. (3.14) is reduced to:

$$\mathbf{R}_{\alpha_k}\Delta\boldsymbol{\tau} + \boldsymbol{\tau}_k = \Delta\lambda\mathbf{R}_{\Delta\alpha}\boldsymbol{\tau}'_k + \Delta\boldsymbol{\tau} \tag{3.56}$$

A 2d rotation matrix can be parametrized by a 2d vector $\boldsymbol{\varrho} \in \mathbb{R}^2$ with unit length, i. e., $\|\boldsymbol{\varrho}\| = 1$, representing $\cos(\alpha)$ and $\sin(\alpha)$:

$$\mathbf{R}_\alpha = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} = \begin{pmatrix} \varrho_x & -\varrho_y \\ \varrho_y & \varrho_x \end{pmatrix} \tag{3.57}$$

Note that a rotation with isometric scaling $\lambda\mathbf{R}_\alpha$ can be parametrized the same way by dropping the unit length constraint, yielding a linear minimal parametrization $\boldsymbol{\mu} = (\boldsymbol{\varrho}, \boldsymbol{\tau}) \in \mathbb{R}^4$ for 2d similarity transformations.

Inserting eq. (3.57) into eq. (3.56) yields a linear equation system that can be solved for the planar eye-to-eye transformation $(\Delta\boldsymbol{\varrho}, \Delta\boldsymbol{\tau})$:

$$\begin{pmatrix} (\boldsymbol{\tau}'_k)_x & -(\boldsymbol{\tau}'_k)_y & 1-\cos(\alpha_k) & \sin(\alpha_k) \\ (\boldsymbol{\tau}'_k)_y & (\boldsymbol{\tau}'_k)_x & -\sin(\alpha_k) & 1-\cos(\alpha_k)) \end{pmatrix} \begin{pmatrix} \Delta\boldsymbol{\varrho} \\ \Delta\boldsymbol{\tau} \end{pmatrix} = \begin{pmatrix} (\boldsymbol{\tau}_k)_x \\ (\boldsymbol{\tau}_k)_y \end{pmatrix} \tag{3.58}$$

resp.

$$\min_{\Delta\boldsymbol{\varrho},\Delta\boldsymbol{\tau}\in\mathbb{R}^2} \sum_{k=1}^{m} \|\mathbf{R}_{\boldsymbol{\tau}'_k}\Delta\boldsymbol{\varrho} + (\mathbf{I}_2 - \mathbf{R}_{\alpha_k})\Delta\boldsymbol{\tau} - \boldsymbol{\tau}_k\|^2 \tag{3.59}$$

Note that this approach can also be applied for non-planar motion to incorporate partial knowledge about the camera setup. In this case, $\boldsymbol{\Pi}$ is replaced by $[\mathbf{R}_\Pi| - h\boldsymbol{e}_y]$ in eq. (3.54) and (3.55), and eq. (3.56) is formulated with 3d vectors $\boldsymbol{t}_k$, $\boldsymbol{t}'_k$, 3 dof rotations $\mathbf{R}_k$, $\Delta\tilde{\mathbf{R}} = \mathbf{R}_{\boldsymbol{e}_y,\Delta\alpha}$, and $\Delta\tilde{\boldsymbol{t}} = (\Delta\tau_x, 0, \Delta\tau_y)$.

3.7 Robust Eye-to-Eye Calibration

3.7.1 Pose Selection

In the following we will describe guidelines for the selection of two rigidly coupled motions $(\mathbf{T}_{k_1}, \mathbf{T}'_{k_1})$ and $(\mathbf{T}_{k_2}, \mathbf{T}'_{k_2})$ that are appropriate for estimating the eye-to-eye transformations between two cameras, i. e., pose configurations avoiding close to singular conditions of the rigid motion equations. Critical factors and criteria to improve the accuracy of hand-eye calibration have been identified and described already by Tsai & Lenz in their seminal publication [TL89]. Their main points are:

- The hand-eye rotation error is inversely proportional to the sine of the angle between the rotation axes of both motions, i. e., $\sin(d_\angle(\boldsymbol{r}_{k_1}, \boldsymbol{r}_{k_2}))$ resp. $\sin(d_\angle(\boldsymbol{r}'_{k_1}, \boldsymbol{r}'_{k_2}))$. Hence, one criterion for pose selection is the non-parallelism of the rotation axes.
- The hand-eye rotation and translation errors are both inversely proportional to the rotation angles α_{k_i} resp. α'_{k_i}. Hence, rotation of the camera system should be as large (i. e., close to $+\pm 90^\circ$) as possible.
- The hand-eye translation error is proportional to the magnitude of translation $\|\boldsymbol{t}_k\|$ of the the first camera, but does not depend on the magnitude of translation $\|\boldsymbol{t}'_k\|$ of the second camera. This is due to error propagation from decoupled rotation estimation (see Sec. 3.4.2).

Shi, Wang & Liu [SWL05] suggest the following rules for the selection of motion pairs for hand-eye calibration based on these observations:

- Select only motion pairs (k_1, k_2) with sufficiently large angle between first and second rotation axis, i. e., $\beta_{k_1,k_2} := d_\angle(\boldsymbol{r}_{k_i}, \boldsymbol{r}_{k_2}) > \beta_{\min}$ and $\beta'_{k_1,k_2} := d_\angle(\boldsymbol{r}'_{k_1}, \boldsymbol{r}'_{k_2}) > \beta_{\min}$ for a chosen threshold $\beta_{\min} > 0$.
- Select only motions k_i with sufficiently large rotation angle, i. e., $|\alpha_{k_i}| > \alpha_{\min}$ and $|\alpha'_{k_i}| > \alpha_{\min}$ for a chosen threshold $\alpha_{\min} > 0$.
- Select only motions k_i with sufficiently small translation of the first camera, i. e., $\|\boldsymbol{t}_{k_i}\| \leqslant \lambda_{\max}$ for a chosen threshold $\lambda_{\max} > 0$.

Zhang, Shi & Liu describe a method for adaptive selection of thresholds $\beta_{\min}$, $\alpha_{\min}$ and $\lambda_{\max}$ using iterative polynomial regression in [ZSL05].

Schmidt, Vogt & Niemann propose two similar approaches to select motions with distinct rotation axes for hand-eye calibration in [SVN03] and [SVN04]. Both methods reject motions with small rotation angle in a preprocessing step as in the approach above. The first method is considered as an exhaustive search. All motion pairs (k_1, k_2) are rated by the score $\max(\beta_{k_1,k_2}, \beta'_{k_1,k_2})$, favoring motion pairs with large angle between their rotation axes. The highest rated pairs are then used for hand-eye calibration from eq. (3.12). The second method is based on vector quantization. Motions with sufficiently distinct rotation axes are selected by clustering motions with respect to rotation axis direction and selecting only one representative per cluster for hand-eye calibration.

The methods described here can all be applied to eye-to-eye calibration in a straightforward way. However, we will ignore the third rule preferring small motion of the first camera, since this poses only problems for the decoupled solution of the rigid motion equation.

3.7.2 Motion Model Selection

In Sec. 3.6 we provided partial solutions for critical motion configurations, i. e., planar motion and pure translation. These cases can be detected using the converse criteria for pose selection from Sec. 3.7.1:

- For motion pairs (k_1, k_2) with sufficiently small angle between first and second rotation axis, i. e., $\beta_{k_1,k_2} := d_{\angle}(\boldsymbol{r}_{k_i}, \boldsymbol{r}_{k_2}) < \beta_{\max}$ and $\beta'_{k_1,k_2} := d_{\angle}(\boldsymbol{r}'_{k_1}, \boldsymbol{r}'_{k_2}) < \beta_{\max}$ for a chosen threshold $\beta_{\max} > 0$, apply the second method described in Sec. 3.6.2.
- For motions k_i with sufficiently small rotation angle, i. e., $|\alpha_{k_i}| < \alpha_{\max}$ and $|\alpha'_{k_i}| < \alpha_{\max}$ for a chosen threshold $\alpha_{\max} > 0$, apply the method described in Sec. 3.6.1.

Thresholds for motion model selection can be derived from the experiments in Sec. 3.8.

3.7.3 Outlier Handling

The methods described in the previous section aim at rejecting input poses that lead to instable or ambiguous solutions, especially for the minimal case of two pose correspondences. However, they do not identify erroneous data resulting from failures in the subsequent pose acquisition process ("outliers") that might corrupt the calibration process significantly.

Robust eye-to-eye calibration in the presence of outliers can be achieved via a RANSAC approach [FB81], i. e., computing solutions from a number of random samples consisting of two pose correspondences and evaluating the number of pose correspondences that are consistent with this solution (see C.4 for further details):

- ▻ Draw a pair of random pose correspondences $(\mathbf{T}_{k_1}, \mathbf{T}'_{k_1})$ and $(\mathbf{T}_{k_2}, \mathbf{T}'_{k_2})$ (subject to the pose selection strategy described in Sec. 3.7.1).
- ▻ Compute an eye-to-eye transformation hypothesis $\Delta\mathbf{T}$ from the sample.
- ▻ Evaluate the set of "inliers" with respect to $\Delta\mathbf{T}$:

  $$\mathcal{I}_{\Delta\mathbf{T}} = \{(\mathbf{T}_k, \mathbf{T}'_k) \mid d(\mathbf{T}_k, \Delta\mathbf{T}\mathbf{T}'_k\Delta\mathbf{T}^{-1}) < \varepsilon\}$$

  for a given error threshold ε and distance measure d.[19]
- ▻ Repeat $N_{\max}$ times and return the solution with the largest inlier set.
- ▻ Recompute $\Delta\mathbf{T}$ from all pose correspondences in the final inlier set.

Furthermore, we can eliminate outliers already during subsequent pose acquisition by taking rigid motion constraints between observed pose transformations into account, in particular the equal angle constraint (3.17), equal pitch constraint (3.18) and equal inter-axis angle constraint (3.21). A straightforward approach is to reject all pose correspondences resp. pairs of pose correspondences where the constraints are not satisified given heuristic thresholds:

- ▻ Reject $(\mathbf{T}_k, \mathbf{T}'_k)$ if $\min(|\alpha_k - \alpha'_k|, |\alpha_k + \alpha'_k|) > \delta_\alpha$ for a threshold δ_α.

[19] More specifically, rotation errors are evaluated with $d_\angle$ w.r.t. threshold ε_α and translation errors are evaluated with d_{geom} w.r.t. threshold ε_t.

- ▷ Reject $(\mathbf{T}_k, \mathbf{T}'_k)$ if $||\boldsymbol{r}_k^\top \boldsymbol{t}_k| - |\boldsymbol{r}_k'^\top \boldsymbol{t}'_k|| > \delta_p$ for a threshold δ_p (only for equal scale of $\mathcal{C}$ and $\mathcal{C}'$).
- ▷ Reject $(\mathbf{T}_{k_1}, \mathbf{T}'_{k_1})$, $(\mathbf{T}_{k_2}, \mathbf{T}'_{k_2})$ if $|d_\angle(\boldsymbol{r}_{k_1}, \boldsymbol{r}_{k_2}) - d_\angle(\boldsymbol{r}'_{k_1}, \boldsymbol{r}'_{k_2})| > \delta_\beta$ for a threshold δ_β.

Given covariance matrices $\boldsymbol{\Sigma}_{\boldsymbol{\mu}_k}$, $\boldsymbol{\Sigma}_{\boldsymbol{\mu}'_k}$ for input poses, we can compute the thresholds with respect to a given significance value $\theta \in [0, 1]$ instead, describing the probability that the observed measurements agree w.r.t. the probability distributions of the pose measurement errors. The thresholds describe the boundaries of the *confidence interval* for θ.

Suppose that the errors of the rotation angles for the k-th motion are described by zero-mean Gaussian distributions with standard deviations σ_{α_k}, $\sigma_{\alpha'_k}$. Thus, the rotation angle difference $\alpha_k - \alpha'_k$ is described by a zero-mean Gaussian distribution with standard deviation $\sigma_k = \sqrt{\sigma_{\alpha_k}^2 + \sigma_{\alpha'_k}^2}$. Select the threshold δ_α subject to $P[|\alpha_k - \alpha'_k| < \delta_\alpha] = 2\Phi(\frac{\delta_\alpha}{\sigma_k}) - 1 = \theta$, i. e., $\delta_\alpha = \Phi^{-1}(\frac{\theta+1}{2})$ where $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{1}{2}t^2} \, \mathrm{d}t$ is the probability distribution function of the normal distribution. Thresholds δ_p and δ_β can be determined analogously.

3.8 Evaluation

3.8.1 Method Comparison

We implemented the following eye-to-eye calibration methods as described in the previous sections in MATLAB and C/C++ (see also Table 3.1):

- ▷ Direct decoupled methods of Tsai & Lenz, Shiu & Ahmad, Chou & Kamel, Park & Martin, Horaud & Dornaika, and Andreff et al. as described in Sec. 3.4.1 and 3.4.2 (denoted jointly as `decoupled`)
- ▷ Direct combined methods of Andreff et al. (`Tmat`) and Daniilidis et al. (`dualquat`) as described in Sec. 3.4.3

- ▻ Nonlinear combined methods of Strobl & Hirzinger ($\texttt{angle}_{\texttt{NL}}$), Horaud & Dornaika ($\texttt{qvec}_{\texttt{NL}}$) and the combined version of Chou & Kamel ($\texttt{quat}_{\texttt{NL}}$) as described in Sec. 3.4.4
- ▻ Joint optimization of camera poses and eye-to-eye transformation ($\texttt{W-TLS}_{\texttt{NL}}$) as described in Sec. 3.4.6

Extended versions of all methods have been implemented as described in Sec. 3.5, denoted by a superscript + in the method name.
Statistically weighted versions of the combined methods as described in Sec. 3.4.5 are denoted by a suffix `W-`.

Generation of test data In order to evaluate the sensitivity of all methods with respect to pose measurement errors, properties of input motions and system configurations, we generate the following input data:

- ▻ Create $n-1$ random eye-to-eye configurations $\Delta\mathbf{T}_{1,2}, \ldots, \Delta\mathbf{T}_{1,n}$ consisting of rotations $\Delta\mathbf{R}_{1,i}$ and translations $\Delta\boldsymbol{t}_{1,i}$. The rotation angles are set to a fixed value $\Delta\alpha$ and the absolute translation lengths are set to a fixed value Δd.
- ▻ Create m random poses $\mathbf{T}_k^{(1)}$, $k = 1, \ldots, m$ for the first camera. The absolute rotation angle for each rotation is set to a fixed value α and the absolute translation length is set to a fixed value d. The rotation axes are chosen so that $d_{\angle}(\boldsymbol{r}_{k-1}^{(1)}, \boldsymbol{r}_k^{(1)}) = \beta$ for $k = 2, \ldots, m$ for a fixed value β.
- ▻ Compute local poses of the i-th camera $\mathbf{T}_k^{(i)} = \Delta\mathbf{T}_{1,i}^{-1}\mathbf{T}_k^{(1)}\Delta\mathbf{T}_{1,i}$ for each $i = 2, \ldots, n$, $k = 1, \ldots, m$.
- ▻ Add random rotation and translation errors to all poses, resulting in translations $\hat{\boldsymbol{t}}_k^{(i)} = \boldsymbol{t}_k^{(i)} + \boldsymbol{t}_\varepsilon$ and rotations $\hat{\mathbf{R}}_k^{(i)} = \mathbf{R}_\varepsilon\mathbf{R}_k^{(i)}$. Translation errors are drawn from a multivariate Gaussian distribution $\boldsymbol{t}_\varepsilon \sim \mathcal{N}_3(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{t}_\varepsilon})$. Rotation errors are created from uniformly distributed rotation axes $\boldsymbol{r}_\varepsilon \in \mathbb{S}^2$ and Gaussian distributed rotation angles $\alpha_\varepsilon \sim \mathcal{N}(0, \sigma_{\alpha_\varepsilon}^2)$, i. e., $\mathbf{R}_\varepsilon = \mathbf{R}_{\boldsymbol{r}_\varepsilon, \alpha_\varepsilon}$.[20]

[20] Note that picking random vectors on the unit sphere is not trivial. In particular, it is not

In the following we will evaluate the accuracy of the eye-to-eye calibration methods w.r.t. ground truth poses under variation of different parameters such as input error magnitude $\sigma_{\alpha_\varepsilon}$ and $\sigma_{\boldsymbol{t}_\varepsilon}$, number m of input poses, max. input rotation angle α and translation length d, max. angle between rotation axes β, and finally angle $\Delta\alpha$ and distance Δd between rigidly coupled cameras.

For each set of parameters, a large number of 1000 random samples is created. For each instance, eye-to-eye transformations $\Delta\hat{\mathbf{T}}_{1,i}$ are estimated for $i = 2, \ldots, n$ and the error w.r.t. to the ground truth transformation $\Delta\mathbf{T}_{1,i}$ is stored. The rotational error is measured by the angle of the residual rotation $d_\angle(\Delta\mathbf{R}_{1,i}, \Delta\hat{\mathbf{R}}_{1,i})$, for the translational part the absolute difference $\|\Delta\boldsymbol{t}_{1,i} - \Delta\hat{\boldsymbol{t}}_{1,i}\|$ is used. Each plot shows the average estimation error and its standard deviation, displayed with a different color for each method.

In all simulations, we use translation lengths $\Delta d = 1\,\text{m}$, $d = 1\,\text{m}$, rotation angles $\Delta\alpha = 60^\circ$, $\alpha = 30^\circ$, $\beta = 90^\circ$, $n = 2$ cameras, and $m = 4$ motions if not stated otherwise. The default standard deviation for input rotation errors is chosen as $\alpha_\varepsilon = 1^\circ$ for the rotation estimation experiments and $\alpha_\varepsilon = 0.5^\circ$ for the combined estimation experiments. Translations errors are by default drawn from an equally distributed uncorrelated error distribution, i. e., $\boldsymbol{\Sigma}_{\boldsymbol{t}_\varepsilon} = \sigma_{\boldsymbol{t}_\varepsilon}\mathbf{I}$, with $\sigma_{\boldsymbol{t}_\varepsilon} = 0.01\,\text{m}$ (i. e., 1% of translation distances).

Rotation estimation First, we evaluate rotation estimation from the decoupled methods. The results are shown in Fig. 3.8.

- ▷ In the first test case (see Fig. 3.8, top left), the standard deviation of input rotation errors is increased from $\sigma_{\alpha_\varepsilon} = 0$ to 2°. The resulting rotation error is linearly proportional to the magnitude of $\sigma_{\alpha_\varepsilon}$. Note that all method provide virtually the same results apart from Shiu & Ahmad which is slightly inferior to the other methods.
- ▷ In the second test case (see Fig. 3.8, top right), the number of motions is increased from the minimal number $m = 2$ to 16. The estimation

correct to select spherical coordinates from uniform distributions $\phi \in [0, 2\pi]$ and $\theta \in [0, \pi]$, since the resulting points will cluster at the poles. We use the *trig method* from [Rus96] instead and select $z \in [-1, 1]$, obtaining vectors $v = (\cos(\phi)r, \sin(\phi)r, z)^\intercal$ with $r = \sqrt{1 - z^2}$ that are uniformly distributed over $\mathbb{S}^2$.

error is reduced with rising number of motions (approx. proportional to $\sqrt{m}$). For $m = 2$, all methods provide the same results. With increasing number of motions, Shiu & Ahmad's method performs worse than the rest due to the fact that additional parameters are introduced for every motion (cf. discussion in Sec. 3.4.1).

- In the third test case (see Fig. 3.8, center left), the input rotation angle α is decreased from 90° to 5°. The estimation error increases inversely proportional to α and begins converges around $\alpha > 30^\circ$. This illustrates observation #2 in Sec. 3.7.1.
- In the fourth test case (see Fig. 3.8, center right), the maximal angle β between rotation axes is decreased from 90° to 5°. The estimation error increases inversely proportional to β, rising drastically for $\beta < 10^\circ$, approaching the singular case of rotation around a single axis, and converging for $\beta > 45^\circ$. This illustrates observation #1 in Sec. 3.7.1.
- In the fifth test case (see Fig. 3.8, bottom left), the eye-to-eye rotation angle $\Delta\alpha$ is increased from 0° to 180°. Apparently, the estimation results are independent from the system configuration with the exception of Tsai & Lenz' method which becomes corrupt when $\Delta\alpha$ is close to 180°. The latter is due to the fact that the Cayley-Gibbs-Rodrigues vector used there cannot represent rotations of $\pm 180^\circ$ (cf. Sec. 2.3.2).

These test cases confirm the observations made in Sec. 3.7.1 and illustrate the accuracy of decoupled rotation estimation w.r.t. critical parameters. Additional simulatations for several combinations of parameters further confirm that the eye-to-eye rotation estimation error and its standard deviation increase linearly with the magnitude of the input rotation error in the general case.
Apart from Tsai & Lenz and Shiu & Ahmad, all methods exhibit essentially the same accuracy. Hence, we will consider w.l.o.g. only Chou & Kamel's method minimizing the quaternion distance in the following due to its computational efficiency. The associated decoupled method is referred to as `decoupled`.

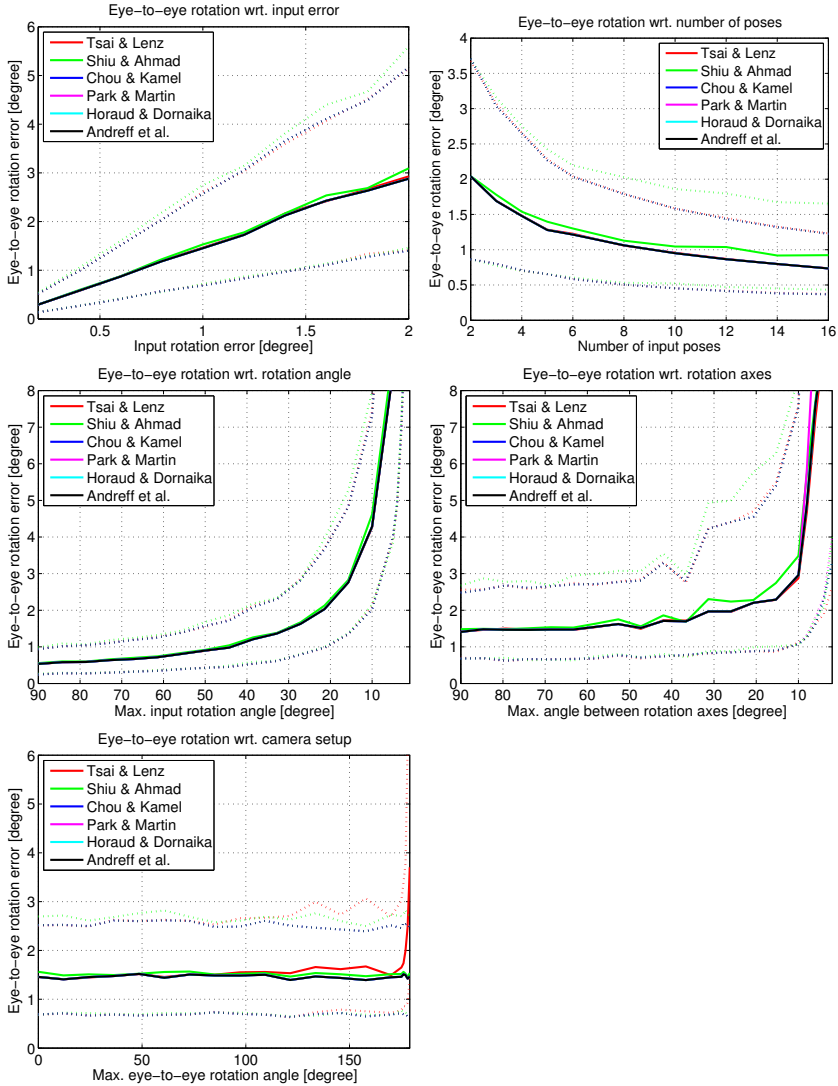

Figure 3.8. Comparison of eye-to-eye rotation estimation methods with respect to different parameters (from top left to bottom right: standard deviation $\sigma_{\alpha_\varepsilon}$ of input rotation error, number m of rigidly coupled motions, input rotation angle α, max. angle β between rotation axes, eye-to-eye rotation angle $\Delta\alpha$).

Combined estimation Next, we evaluate combined rotation and translation estimation and compare the results with the decoupled approach. Nonlinear methods use the result of the decoupled method as starting point. Additional to the simulations for rotation estimation, we evaluate the estimation error w.r.t. translation distance d of the first camera resp. distance between the rigidly coupled cameras Δd. The results are shown in Fig. 3.9–3.11.

- In the first test case, the standard deviation of input rotation errors is increased from $\sigma_{\alpha_\varepsilon} = 0.5^\circ$ to 5° while $\sigma_{\boldsymbol{t}_\varepsilon} = 0.01\,\mathrm{m}$ is fixed (see Fig. 3.9, top) resp. the standard deviation of input translation errors is increased from $\sigma_{\boldsymbol{t}_\varepsilon} = 0.01\,\mathrm{m}$ to $0.1\,\mathrm{m}$ (i. e., 10% of translation distances) while $\sigma_\alpha = 0.5^\circ$ is fixed (see Fig. 3.9, center). The results of both combined and decoupled estimation are linearly proportional to the magnitude of input rotation and translation error respectively. However, the actual accurcay of the individual methods depends heaviliy on the ratio between input rotation and translation errors, especially notable in eye-to-eye rotation estimation (see below).
- The following three test cases – increasing the number of motions m (see Fig. 3.9, bottom), input rotation angle α (see Fig. 3.10, top) and maximal angle β between rotation axes (see Fig. 3.10, center) – provide basically the same conclusions the respective test cases for decoupled rotation estimation. All methods provide very similar results here.
- In the fifth test case (see Fig. 3.10, bottom), the maximal translation d of the first camera is increased from $d = 0.01\,\mathrm{m}$ to 10m. Note that the absolute translation error is still fixed as $\sigma_{\boldsymbol{t}_\varepsilon} = 0.01\,\mathrm{m}$. As expected, the decoupled rotation estimation is not affected by this. However, decoupled estimation translation degrades in accuracy with growing translation length d. This is due to the fact that errors from decoupled rotation estimation are enhanced by $\|\boldsymbol{t}_k^{(1)}\|$ in the translation equation (3.30) as observed in Sec. 3.7.1. The direct combined methods are significantly less susceptible to large translations, `Tmat` degenerates when motion approaches pure rotation of the first camera ($d \approx \sigma_{\boldsymbol{t}_\varepsilon}$). This is due to the fact that the unconstrained matrix equation (3.32) becomes singular for this case.

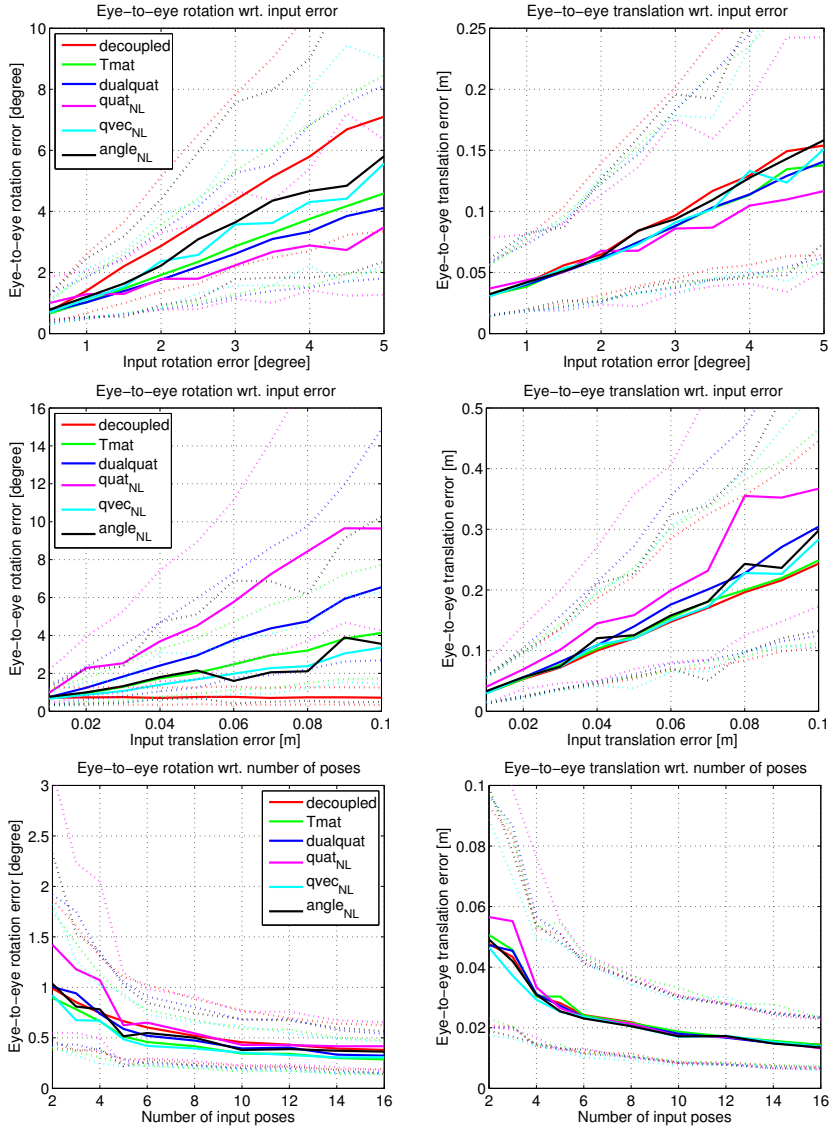

Figure 3.9. Comparison of combined eye-to-eye calibration methods with respect to different parameters (from top to bottom: magnitude $\sigma_{\alpha_\varepsilon}$ of input rotation error, magnitude σ_{t_ε} of input translation error, number m of rigidly coupled motions).

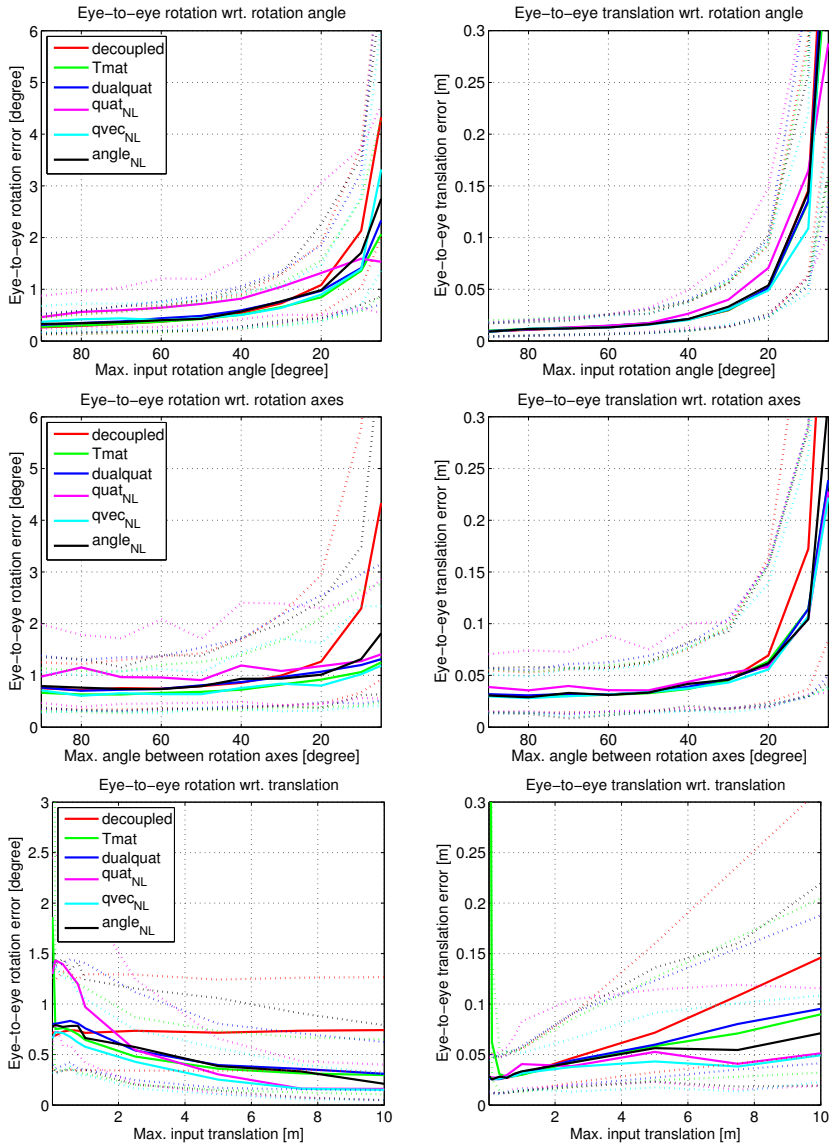

Figure 3.10. Comparison of combined eye-to-eye calibration methods with respect to input pose parameters (from top to bottom: input rotation angle α, max. angle β between rotation axes, max. translation distance d).

▷ In the last test case, the eye-to-eye rotation angle $\Delta\alpha$ is increased from 0° to 180° (see Fig. 3.11, upper row) resp. the eye-to-eye translation distance Δd is increased from 0.1 r to 2 m (see Fig. 3.11, lower row). As for decoupled rotation estimation, the results of the combined methods are also independent from both parameters, suggesting that the actual camera setup is not significant for the absolute accuracy of eye-to-eye calibration.

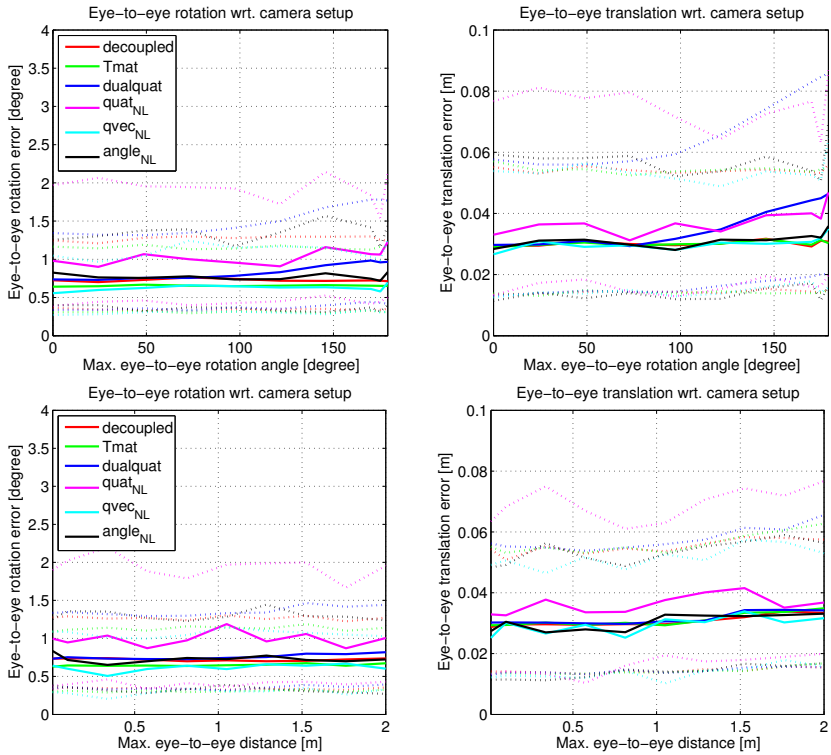

Figure 3.11. Comparison of combined eye-to-eye calibration methods with respect to eye-to-eye parameters (top: eye-to-eye rotation angle $\Delta\alpha$, bottom: eye-to-eye translation distance $\Delta\boldsymbol{t}$).

Weighted estimation In the first test case of combined estimation evaluation, we observed that the combined methods provide rather different results in terms of accuracy of the estimated eye-to-eye rotations and translations (compareFig. 3.9). This can be explained by the inequality of the residuals resulting from the rotational and translational error terms which are assumed to be linearly proportional to σ_α and σ_{t_ε}.[21]

For small translation errors w.r.t. rotation errors, the rotational error dominates the translational error leading to more accurate eye-to-eye rotation estimation at the expense of translation estimation. As a consequence, for large translation errors w.r.t. to the magnitude of rotation errors the decoupled method provides significantly better results than the combined methods (as in Fig. 3.9, center). As illustrated in Fig. 3.12, this behaviour is successfully remedied by weighting the rotation and translation error terms in the combined methods according to the approximative variance of the residuals. The direct method `W-dualquat` performs slightly better than `W-Tmat`. The results of the nonlinear methods are very close to `W-dualquat`.

Extended estimation Following estimation with known scale, extended eye-to-eye transformation estimation is evaluated for $m = 4$ motions. The results are plotted in Fig. 3.13.

In the first test case using the default methods (see Fig. 3.13, top), the relative scale $\Delta\lambda$ is ranging from 0.5 to 2. As expected, eye-to-eye rotation estimation using the decoupled method is not affected. However, decoupled translation estimation and all combined methods degenerate rapidly as soon as $\Delta\lambda$ deviates from 1. It is noteworthy that the methods based on quaternion distance, i. e., `dualquat` and $\texttt{quat}_{\texttt{NL}}$ are most notably affected by the relative scaling. As described in Sec. 3.5, the `Tmat` method is not affected by scaling, since it implicitly estimated a relative scale factor.

Using the extended eye-to-eye calibration methods, neither the decoupled method nor the combined methods are notably affected by relative scale estimation (see Fig. 3.13, center). Furthermore, weighting the residuals

[21] E. g., for $\sigma_\alpha = 0.5^\circ \approx 0.01\,\text{rad}$, the residuals of rotation and translation terms are within the same order of magnitude w.r.t. the method $\texttt{ang}_{\texttt{NL}}$ that minimizes the angle metric $d_\angle$.

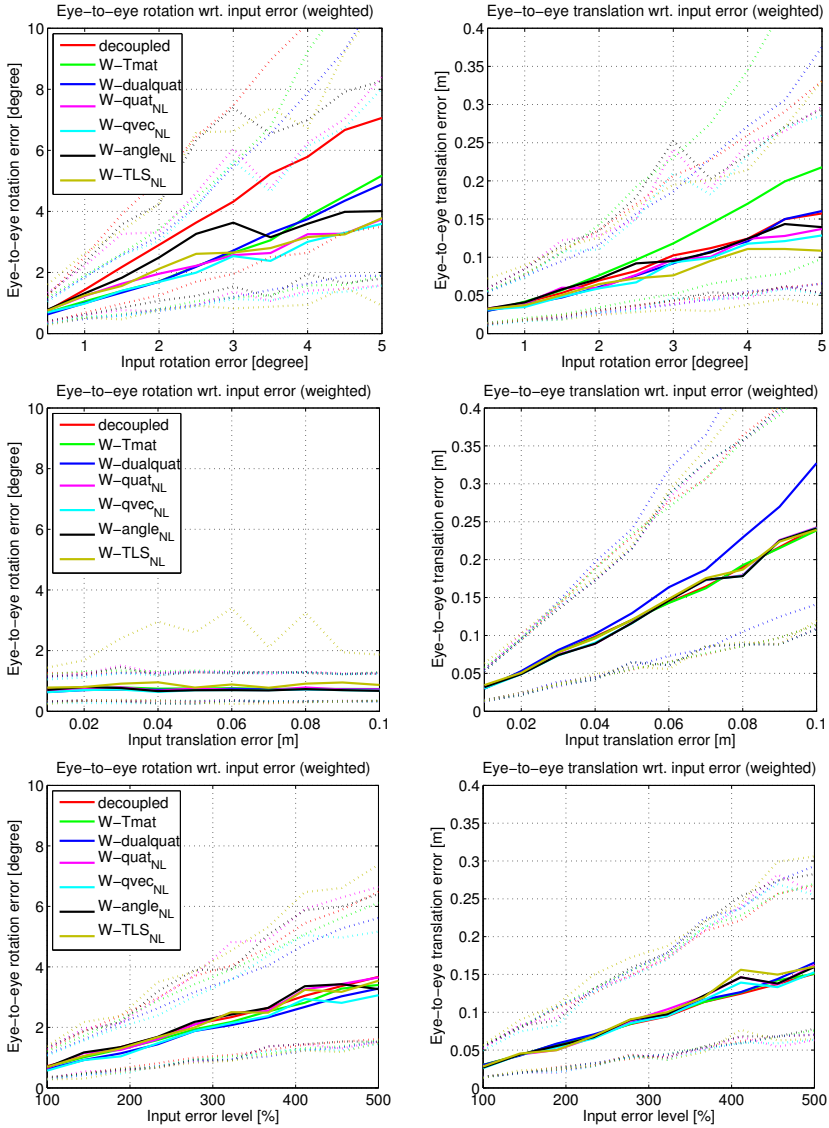

Figure 3.12. Comparison of weighted combined eye-to-eye calibration methods with respect to input error (from top to bottom: magnitude $\sigma_{\alpha_\varepsilon}$ of input rotation error, magnitude σ_{t_ε} of input translation error, both magnitudes simultaneously w.r.t. $\sigma_{\alpha_\varepsilon} = \rho \cdot 0.5^\circ$, $\sigma_{t_\varepsilon} = \rho \cdot 0.01\,\mathrm{m}$).

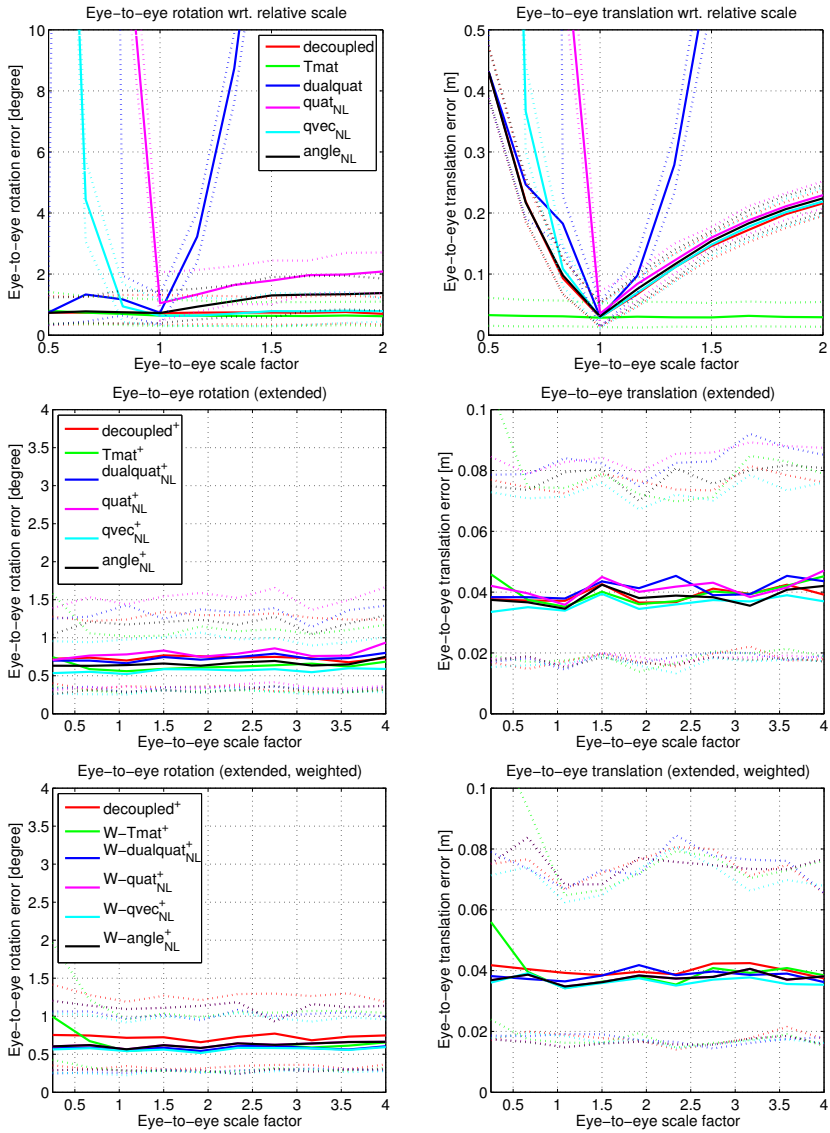

Figure 3.13. Comparison of extended eye-to-eye calibration methods with respect to relative scale factor $\Delta\lambda$ (from top to bottom: w/o extended methods, using extended methods, using weighted extended methods).

with respect to expected input errors slightly improves the accuracy of the extended methods in this case (see Fig. 3.13, bottom).

Expanding the range of analyzed relative scales to $[0.25, 4]$ reveals that the `Tmat` method decreases in accuracy for smaller $\Delta\lambda$. This is likely caused by the suboptimal renormalization of the resulting parameter vector as already mentioned by the authors in [AHE01].

Estimation with error accumulation Finally, the impact of error accumulation on the resulting accuracy is evaluated (see Fig. 3.14). Given $m = 4$ input poses with initial error magnitude of $\sigma_\alpha = 0.5^\circ$ and $\sigma_{t_\varepsilon} = 0.01\,\text{m}$, the error is increased by $\rho \cdot \sigma_\alpha$, $\rho \cdot \sigma_{t_\varepsilon}$ for each motion where ρ ranges from 0% to 200%, resulting in a final error of $\sigma_\alpha = 3.5^\circ$ and $\sigma_{t_\varepsilon} = 0.07\,\text{m}$ for $\rho = 2$.

Eye-to-eye calibration was tested with default methods, weighted methods with fixed weight, and W-TLS_{NL} using adaptive weights. As expected, the results deteriorate with rising error gain, approaching the results for fixed error level of approx. 400% of the initial error (see Fig. 3.12). The methods with fixed weights show equal results. Choosing adaptive weights alleviates the impact of error accumulation, reducing rotation and translation estimation errors by approx. 25% for $\rho \geqslant 1$.

3.8.2 Partial Solution From Planar Motion

In the following experiment, the modified partial calibration methods from Sec. 3.6.2 for planar motion are evaluated. The test case is similar to restricting the angle between rotation axes (cf. Fig. 3.10, center): Input motions gradually degenerate to planar motion by reducing the maximal angle β between rotation axes from 30° to 0° while also restricting the maximal angle between translation vectors and the rotation plane to β. This implicitly defines a virtual plane of motion Π with normal $\boldsymbol{n}$ from which the input motions deviate by less than β. The ground truth eye-to-eye translation $\Delta\hat{\boldsymbol{t}}$ is restricted to Π.

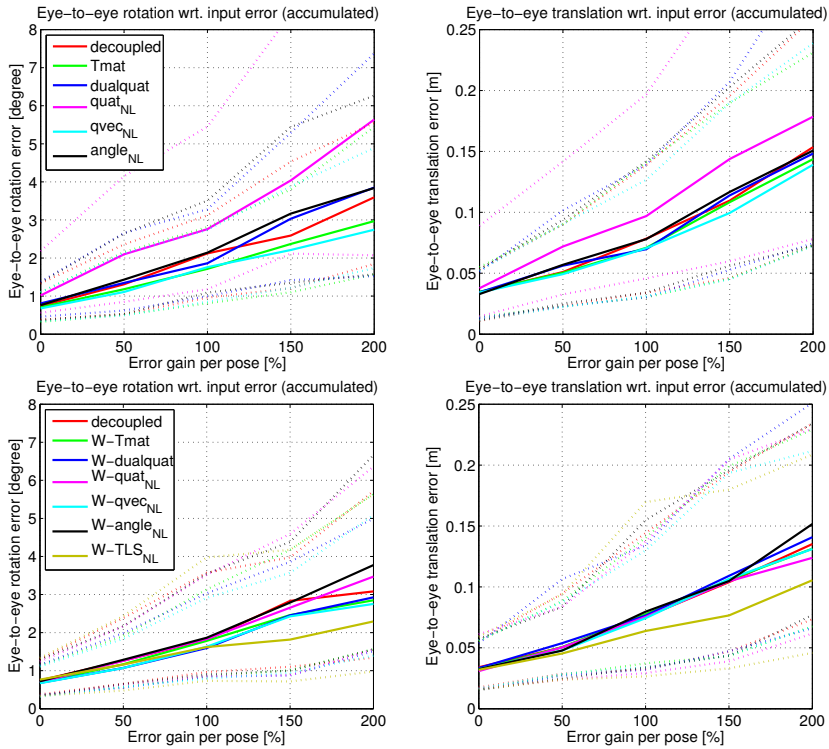

Figure 3.14. Comparison of combined eye-to-eye calibration methods with respect to input error accumulation (top: w/o weights, bottom: weighted methods). Input error magnitude is increased by $\rho \cdot 0.5^\circ$ resp. $\rho \cdot 0.01\,\mathrm{m}$ per motion.

The direct methods use additional constraints between input pose pairs assuming ideal planar motion as described in Sec. 3.6.2. The ambiguity of the resulting eye-to-eye translation is fixed by adding the linear constraint $\hat{\boldsymbol{n}}^\intercal \Delta \boldsymbol{t} = 0$ where $\hat{\boldsymbol{n}}$ approximates the normal of the assumed plane of motion Π by averaging over all rotation axes $\hat{\boldsymbol{r}}^{(1)}_{1,\dots,m}$. Estimation errors parallel to $\boldsymbol{n}$ are not taken into account in the error evaluation, since this quantity cannot be recovered from planar motions without previous knowledge.

The error plots in Fig. 3.15 demonstrate the behaviour of unconstrained and constrained methods for motions approaching the planar case. Decoupled estimation becomes inferior to the constrained method for $\beta < 12^\circ$. The combined methods `Tmat` and `dualquat` show higher robustness and provide accurate results up to $\beta \approx 2.5^\circ$. By adding planar motion constraints, all methods converge for $\beta \approx 0$. For the given error level, thresholds for model selection $\beta_{\text{max}} \approx 5^\circ$ and pose selection $\beta_{\text{min}} \approx 15^\circ$ seem reasonable, given by the locations where constrained and unconstrained methods reach approx. 200% of their converged accuracy.

It is noteworthy that the combined nonlinear methods exhibit the same accuracy for translation estimation with and w/o planar motion constraint, contrary to the results in Fig. 3.10 (center). This is due to the fact that with decreasing β, the translation error function becomes invariant w.r.t. shifting $\Delta \boldsymbol{t}$ orthogonal to the motion plane Π, so iterative nonlinear optimization methods such as the Levenberg-Marquardt algorithm will most probably refrain from modifying the translation parameters along $\boldsymbol{n}$. Hence, explicit motion model selection to handle planar motion is mainly important to provide initial solutions from direct methods but not essential for nonlinear optimization.

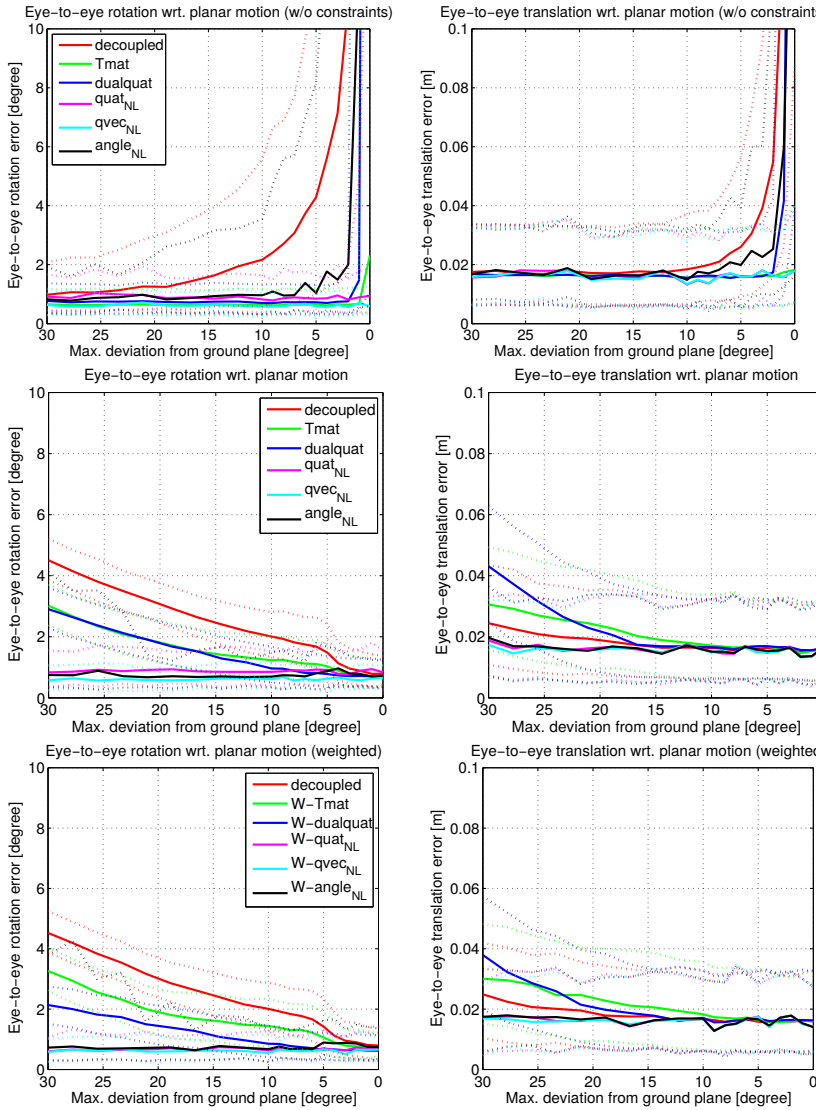

Figure 3.15. Comparison of eye-to-eye calibration methods adapted for planar motion with respect to input motions deviating from planar motion by angle β (from top to bottom: w/o planar constraints, using planar constraints, weighted).

3.9 Summary

In this chapter, we discussed different direct and iterative methods for eye-to-eye calibration from rigidly coupled motions for the general motion case and provided distinct solutions for critical motion configurations, i. e., planar motion and pure translation. For the latter, only partial solutions can be provided without further knowledge about the system configuration.

Existing methods were extended by relative scale estimation which is necessary since poses acquired by Structure from Motion techniques relate in general to differently scaled reference coordinate frames.

We also described a novel method $\texttt{W-TLS}_{\texttt{NL}}$ for simultaneous refinement of camera motion and eye-to-eye transformation w.r.t. pose measurement uncertainties of both cameras.

From theoretical considerations that have been proved by the experimental evaluations with simulated data, the following rules for motion planning and selection of the particular eye-to-eye calibration method can be derived:

- ▻ Apart from Shiu & Ahmad's method, all decoupled methods provide very similar results. We use preferably the quaternion based method by Chou & Kamel (`quat`) due to its computational efficiency.
- ▻ For known scale, the dual quaternion based method by Daniilidis et al. (`dualquat`) provides in general the most stable solution among the direct methods. For the extended case, the combined method of Andreff et al. ($\texttt{Tmat}^{+}$) can be used to provide a direct solution.
- ▻ Weighting the rotational and translational error terms according to the magnitude of the expected residual errors has a crucial effect on the accuracy of combined methods. Therefore, using statistical weights based on the magnitude of input pose measurement errors is strongly recommended.
- ▻ Motions of the rig should be planned with respect to the indications of the experiments, i. e., rotation axes should differ notably and input

rotation angles should be large in the optimal case. However, small range motion provides in general more accurate input poses depending on the actual pose estimation method used. Nonlinear combined methods have proven as capable of handling even close to singular motions given that the input poses are accurate enough.

- ▻ In order to provide accurate relative scale observations to support extended eye-to-eye calibration, distinct translation along the rotation axis should be observed. It is beneficial to keep the number of relative scales as low as possible. In a default calibration scenario one constant scale per camera should be presumed.
- ▻ For critical motions, i. e., pure translation or planar motion, either partial knowledge about the extrinsic calibration should be acquired or specific solutions for the respective motion model should be used. The latter technique requires to apply automatic motion model selection based on empirical thresholds that can be derived from the experiments in Sec. 3.8.2.

Additional to these guidelines, we provided diagnostic measures for the accuracy of rigidly coupled poses based on the rigid motion constraints, i. e., equal rotation angle, pitch, and pairwise inter-axis angle, that can be used to detect gross errors and drift in pose estimation, and described a RANSAC algorithm for robust eye-to-eye calibration.

Part II

Visual Eye-to-Eye Calibration

Chapter 4

Structure from Motion

4.1 Introduction

Structure from Motion (SfM) – i. e., simultaneous estimation of 3d structure of the scene and camera motion from visual features such as points or lines detected in multiple images of the camera – is a well known problem in computer vision. Since its origins that can be dated back to the seminal works of Ullman [Ull79] and Longuet-Higgins [Lon81], Structure from Motion has been the topic of extensive research. For a comprehensive description of Structure from Motion and its basic concepts we refer the reader to the excellent textbooks by Hartley & Zisserman [HZ04] and Szeliski [Sze10].

The Structure from Motion problem is similar to another popular problem in computer vision and robotics, *Simultaneous Localization and Mapping (SLAM)*, as the names already imply. However, classical SLAM approaches typically also incorporate measurements from additional sensors such as inertial sensors, laser range scanners, vehicle odometry, or ultrasound sensors while SfM is based on vision only. Recently, variants of the SLAM problem have evolved such as *MonoSLAM* [Dav+07] that rely on visual data only and consider only a single moving camera (monocular or "bearing-only" SLAM). These methods – often referred to as *Visual SLAM* – are in fact very similar in nature – and sometimes even considered as synonymous – with the Structure from Motion approach.

The basic concept of SLAM is defined by concurrently updating the pose of the camera system and the positions of landmarks in the map from

observations of the landmarks given a mathematical observation model and an observation error distribution, based on an initial system pose and a description of the initial uncertainty of the pose parameters. Additional sensor data is often used to predict the pose update from velocity and acceleration measurements. Common techniques to solve this task are *Extended Kalman Filters*, *Expectation Maximization* approximation, or more recently *Rao-Blackwellized Particle Filters* [BGK06]. Since small, smooth inter-frame movements without high accelerations are expected in order to provide precise pose predictions, the SLAM approach is in general useful for real-time applications where data is captured at high frame rates.

There are in general two different approaches to Structure from Motion[1]: "sparse" feature-based methods and "dense" methods based on optical flow. The first class of SfM methods is rooted in the early work of Longuet-Higgins [Lon81] describing how to estimate the relative pose between two cameras from 2d point correspondences by means of the epipolar geometry (see Chapter 7 in [Sze10]). These methods have the drawback that they are only able to recover sparse 3d information about the scene. On the other hand, they are computationally efficient due to the restricted number of features involved. Therefore, state-of-the-art feature-based SfM methods can be applied to very large unordered datasets consisting of thousands of images in reasonable time [SSS06; Aga+11]. In contrast to SLAM, SfM from sparse feature correspondences is in general not restricted to sequential image feeds, allowing for a wider field of applications. To emphasize this, the literature further distinguishes between *sequential/incremental SfM* (also denoted as "visual odometry") and *non-sequential/hierarchical SfM* applications.

In contrast to feature-based SfM, methods based on optical flow analyze the flow field between two images, i. e., the apparent 2d motion vector for every pixel in the image, which is related to the relative camera motion via the epipolar constraint (see Chapter 8 in [Sze10]). These methods can reproduce dense representations of the scene but rely on similar appear-

[1] Although 3d scene reconstruction and pose tracking using range imaging cameras such as active stereo cameras, time-of-flight cameras, or coded aperture cameras is sometimes also denoted as Structure from Motion (e. g., *RGB-D SfM*), we will refer to the term in the classical notion in this work, i. e., with respect to monocular color cameras.

ance of subsequent images, assuming small interframe displacement and constant brightness, and are computationally expensive. However, in the recent years there has been significant progress on optical flow algorithms that can deal with more complex situations and approach real-time processing by utilizing the GPU [Pau+12]. Nonetheless, approaches based on dense optical flow estimation are restricted to sequential SfM.

For the case of eye-to-eye calibration, accurate camera ego-motion is more important than a detailed 3d model of the scene. There is also no reason to restrict image acquisition to sequential video streams. Therefore, sparse feature-based SfM is considered as the more appropriate choice. In the following we will describe the theoretical background of Structure from Motion and define the notations used in this work. Afterwards, we will describe the building blocks of SfM applications and outline existing solutions.

4.2 Camera Model

4.2.1 Pinhole Camera Model

Most commonly the imaging process of a standard digital CCD or CMOS camera is described by the *pinhole camera model*. This model is derived from an ideal description of the "camera obscura", a classical optical device used to project an image of the environment onto a screen.

Consider the illustration of a pinhole camera in Fig. 4.1. Light passes through the *focal point* $\boldsymbol{C}$ and is projected onto a plane at the back side of the camera. The size of the image depends on the distance between the focal point and the image plane, the *focal length*. Note that the image is flipped with respect to the scene in front of the camera. Since in general the image is flipped again to meet the actual scene geometry, this can be modeled mathematically by placing the image plane in front of the focal point instead of behind it (see Fig. 4.1, right). Because images are projected onto a plane via perspective projection, this model is also referred to as *planar camera model* in this work.

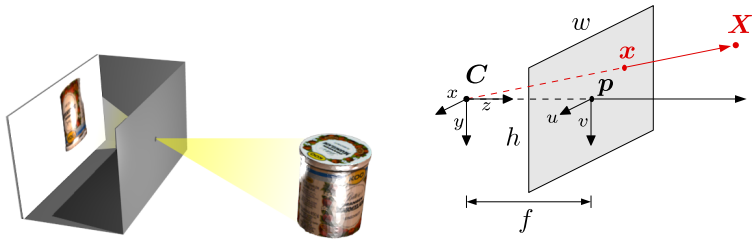

Figure 4.1. Illustration of a pinhole camera (left) and planar camera model describing ideal perspective projection with central point $\boldsymbol{C}$ and focal length f (right).

Since all light rays forming the image intersect in a single point, this kind of projection is also denoted as a central projection. This also implies that all 3d points lying on the same line through the central point are projected onto the same image point. Depth information of 3d points, i. e., their distance to the camera center, is lost during the imaging process.

In the following we will describe the camera model mathematically: Perspective projection onto the plane $z = 1$ is performed by dividing the 3d coordinate vector by the z-component. The resulting coordinate frame is denoted as the *normalized image plane* in this work. Assuming that the focal length of the camera is 1, the normalized image plane coincides with the actual image coordinate frame. This case is referred to as the *calibrated camera*.

In general this is not the case. The image coordinate frame is measured in pixels and its origin is typically on the upper left rather than in the image center. For an ideal pinhole camera, the coordinate transformation can be performed by scaling the coordinates with focal length f (measured in pixel units rather than in meters) and shifting the origin to the pixel coordinates (p_u, p_v), called the *principal point*. Cameras with non-square pixels can be modeled by including an aspect ratio ρ between horizontal and vertical pixel size and a skew factor s, but in general we can assume $\rho = 1$ and $s = 0$ for square pixels.

Putting all together, a 3d point $\boldsymbol{X} = (X, Y, Z)$ within the camera coordinate frame is transformed into 2d image coordinates $\boldsymbol{u}$ by a perspective projection – resulting in the normalized image point $\boldsymbol{x}$ – followed by a linear transformation described by the *camera matrix* $\mathbf{K}$:

$$\boldsymbol{u} = \mathbf{K}\boldsymbol{x} \quad \text{with} \quad \boldsymbol{x} = \boldsymbol{X}/Z \quad \text{and} \quad \mathbf{K} = \begin{pmatrix} f & s & p_u \\ 0 & \rho f & p_v \\ 0 & 0 & 1 \end{pmatrix} \tag{4.1}$$

resp. with the simplified representation of the camera matrix:

$$\mathbf{K} = \begin{pmatrix} f & 0 & p_u \\ 0 & f & p_v \\ 0 & 0 & 1 \end{pmatrix} \tag{4.2}$$

Note that pixel coordinates and normalized points are described using *homogeneous coordinates* $\boldsymbol{u} = (u, v, 1)$ and $\boldsymbol{x} = (x, y, 1)$ here.

The inverse camera matrix converting image pixels into correponding points on the normalized image plane is explicitly given by:

$$\mathbf{K}^{-1} = \begin{pmatrix} \frac{1}{f} & -\frac{s}{\rho f^2} & -\frac{p_u}{f} + \frac{s p_v}{\rho f^2} \\ 0 & \frac{1}{\rho f} & -\frac{p_v}{\rho f} \\ 0 & 0 & 1 \end{pmatrix} \tag{4.3}$$

resp. with reduced parameters:

$$\mathbf{K}^{-1} = \begin{pmatrix} \frac{1}{f} & 0 & -\frac{p_u}{f} \\ 0 & \frac{1}{f} & -\frac{p_v}{f} \\ 0 & 0 & 1 \end{pmatrix} \tag{4.4}$$

The set of parameters (f, p_u, p_v, ρ, s) resp. (f, p_u, p_v) is denoted as *intrinsic parameters* or *intrinsics* of the pinhole camera model.

4.2.2 Distortion Model

Real cameras deviate from the ideal pinhole camera model to a certain degree since the photographic lens used to focus the environmental light introduces radial distortion, depending on the angle of an observed point to the focal axis, and other types of nonlinear displacement such as tangential distortion. There are different approaches to model these distortions mathematically. However, a detailed discussion of lens distortion is beyond the scope of this work. In general, image distortion is described by a nonlinear function $\mathcal{D} : \mathbb{P}^2 \to \mathbb{P}^2, \boldsymbol{x} \mapsto \boldsymbol{x}_d$ mapping a point $\boldsymbol{x}$ in the normalized image plane to a distorted point $\boldsymbol{x}_d$. Distortion is ideally modeled as an invertible function[2], so image coordinates within the normalized image plane can be undistorted by applying $\mathcal{D}^{-1}(\boldsymbol{x}_d)$. The parameters of the distortion function are added to the intrinsic parameters of the camera model. In this work, the common 5-parameter instance of Brown's distortion model [Bro66] is used for perspective cameras:

$$\mathcal{D} : \mathbb{P}^3 \to \mathbb{P}^3, (x, y, 1)^\intercal \mapsto (d_x(x, y), d_y(x, y), 1)^\intercal \tag{4.5}$$

where

$$\begin{aligned} d_x(x, y) &= (1 + k_1 r^2 + k_2 r^4 + k_3 r^6)x + 2p_1 xy + p_2(r^2 + 2x^2) \,, \\ d_y(x, y) &= (1 + k_1 r^2 + k_2 r^4 + k_3 r^6)y + 2p_2 xy + p_1(r^2 + 2y^2) \,, \\ r^2 &= x^2 + y^2 \end{aligned}$$

depending on radial distortion parameters k_1, k_2, k_3 and tangential distortion parameters p_1, p_2.

4.2.3 Calibrated Camera Model

In order to abstract from the actual camera model, we will describe the transformation of 3d points into 2d pixel coordinates by a general – possibly nonlinear – *camera function* $\mathcal{K} : \mathbb{R}^3 \to \mathbb{P}^2, \boldsymbol{X} \mapsto \boldsymbol{u}$ in the remainder

[2] Although in practice the inverse distortion function is in general not given algebraically but approximated numerically which is not a trivial task.

of this work. For the calibrated planar case, we also use the notation $\mathcal{P} : \mathbb{R}^3 \to \mathbb{P}^2, \boldsymbol{X} \mapsto \boldsymbol{x}$ instead where $\mathcal{P}(\boldsymbol{X}) = \boldsymbol{X}/Z$ is the perspective foreshortening as defined in eq. (4.1). Following this notation, the camera function for the pinhole camera model with known distortion mapping $\mathcal{D}$ is given by $\mathcal{K}(\boldsymbol{X}) = \mathbf{K}\mathcal{D}(\mathcal{P}(\boldsymbol{X}))$.

Note that the camera function of any central camera is not injective since $\mathcal{K}(\boldsymbol{X}) = \mathcal{K}(\lambda\boldsymbol{X})$ for all $\lambda \in \mathbb{R} \setminus \{0\}$, i. e., all 3d points on the same ray emerging from the camera center are projected to the same image point. Without knowledge about the 3d scene, the camera function cannot be inverted. However, a "reverse" mapping $\mathcal{U} : \mathbb{P}^2 \to \mathbb{R}^3$ between image pixels and the corresponding projected 3d point up to scale can be defined that satifies $\boldsymbol{u} = \mathcal{K}(\mathcal{U}(\boldsymbol{u}))$. This process is called "unprojection" in the following. The unprojected coordinates coincide with the ray within the camera coordinate frame on which the projected 3d point $\boldsymbol{X}$ is located, i. e., $\mathcal{U}(\boldsymbol{u}) \sim \boldsymbol{X}$. Note that this also implies $\mathcal{U}(\boldsymbol{u}) \sim \boldsymbol{x}$.

Reversing the camera function with respect to the normalized image plane $z = 1$ yields the *planar unprojection function* $\mathcal{U}_\mathcal{P} : \mathbb{P}^2 \to \mathbb{P}^2$ mapping from image pixels $\boldsymbol{u}$ to normalized image coordinates $\boldsymbol{x}$. For a pinhole camera with known distortion mapping $\mathcal{D}$, the unprojection function is given by $\mathcal{U}_\mathcal{P}(\boldsymbol{u}) = \mathcal{D}^{-1}(\mathbf{K}^{-1}\boldsymbol{u})$.

Note that the potentially visible scene is restricted to the field of view:

$$\mathcal{F} = \{\lambda\mathcal{U}(\boldsymbol{u}) \mid \lambda > 0, \boldsymbol{u} \in [0, w] \times [0, h]\} \subset \mathbb{R}^3 \tag{4.6}$$

where (w, h) specifies the image width and height in pixels.

4.2.4 Camera Calibration

In order to facilitate tasks like 3d scene reconstruction, pose estimation and extrinsic multi-camera calibration, we assume in the following that the intrinsic parameters of each camera are known – i. e., that intrinsic calibration techniques such as [Tsa87; Zha00] for the planar camera model or [KB06; SMS06] for the spherical camera model (see B.1) have been used to determine the camera function $\mathcal{K}_i$ for each camera individually – and

that these parameters do not change over time. We also assume that each unprojection function $\mathcal{U}_i$ can either be derived from the respective camera function $\mathcal{K}_i$ algebraically or approximated numerically with subpixel accuracy. If not declared otherwise, we will always work with normalized image coordinates instead of actual pixel positions in the following.

Intrinsic calibration is often realized by capturing multiple images of a calibration pattern, e. g., a planar checkerboard with known size as depicted in Fig. 4.2, from different view points. The calibration pattern is detected in each camera image and the resulting 2d/3d point correspondences are used to estimate the camera poses (extrinsic parameters) and intrinsic parameters minimizing the reprojection error with respect to the chosen camera model. In this thesis, the calibration software from [SBK08] is used which contains the intrinsic calibration method from OpenCV [Bra00] for planar cameras.

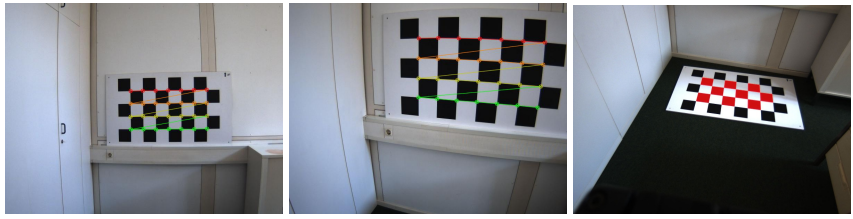

Figure 4.2. Images of a checkerboard pattern for intrinsic camera calibration captured from different view points and at different positions with detected corners drawn into. The right image is overlaid with a rendering of the checkerboard.

4.3 Scene Model

Since we use camera images to reconstruct the 3d structure of the world and the poses of the camera within it, we have to consider an appropriate model of the world. In the following we will refer to the part of the world that we are interested in as the *(global) scene*. The potentially non-overlapping parts of the world that are explored by the cameras are referred to as *local scenes* or *scene parts*.

Obviously there is a multitude of possibilities how to represent a scene mathematically. In the context of this work we will follow the classical Structure from Motion approach and consider only geometric quantities of the scene. We assume the scene to be static while the cameras are moving through it. Apart from this we do not explicitly pose any assumptions about the scene geometry such as planarity or smoothness of surfaces. A scene is simply described by a finite set of 3d point features $\boldsymbol{X} \in \mathbb{R}^3$ located on visible surfaces. However, the following considerations can be extended to other geometric primitives such as lines, planes, or quadric surfaces as well. In order to abstract from the actual parametrization of 3d point features, we will sometimes use a general parameter vector $\boldsymbol{\chi} \in \mathbb{R}^\chi$ instead where the 3d point $\boldsymbol{X}_\chi$ parametrized by $\boldsymbol{\chi}$ is given by some transformation $\mathcal{X} : \mathbb{R}^\chi \to \mathbb{R}^3$, although we have in general $\boldsymbol{\chi} = \boldsymbol{X}$ throughout this work.

The process of estimating the set of 3d points representing the scene from their respective projections in the camera images is referred to as *3d scene reconstruction* or *Structure from Motion* – the latter emphasizing the fact that camera motion and 3d scene structure are in general estimated in conjunction with each other. Statistical considerations on the precision of 3d point estimates is commonly modeled as a covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{X}} \in \mathbb{R}^{3\times3}$ with respect to $\boldsymbol{X}$ assuming that the measurement error of point coordinates is subject to a Gaussian distribution with zero mean.

4.4 Feature Detection and Matching

Classical Structure from Motion methods are based on 2d/2d point correspondences between camera images of the same camera. In computer vision, a multitude of different approaches for feature detection and matching exists. Comprehensive surveys on both topics can be found in [TM08] and [BKD09]. The recommended method depends on aspects like similarity between images, scene texture, availability of color information etc. For moderately textured scenes, corner detectors are commonly used such as KLT corners [ST94] or Harris corners [HS88]. Feature detection is followed by feature extraction, i. e., computation of a feature descriptor from statistics based on the local appearance of the point. Feature descriptors

are designed to be invariant w.r.t. to certain image transformations that are subject to the camera pose such as rotation, translation, scaling, global change in brightness, or image noise. Feature similarity is then evaluated from distance measures between descriptor vectors.

In this thesis, we use the SIFT descriptor [Low04] for wide baseline matching, i. e., for images captured from significantly distinct poses, and KLT tracking [LK81; TK91] for small baseline matching, e. g., for subsequent images from dense video streams. In particular, we use the implementations of SIFT and KLT provided by OpenCV [Bra00]. An example for feature point correspondences is shown in Fig. 4.3.

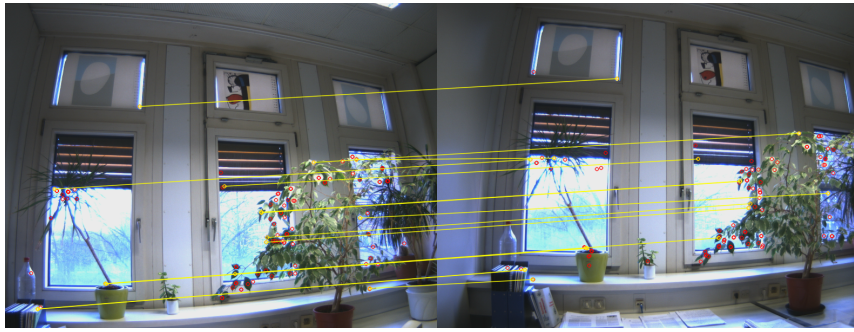

Figure 4.3. Point matches detected between two images using SIFT features.

4.5 Structure from Motion

In the following, we will describe basic Structure from Motion applications providing a sparse 3d reconstruction of the scene and ego-motion of a moving monocular camera with known intrinsics from 2d point correspondences between images captured by the camera. We will consider both incremental SfM for image sequences (e. g., acquired by a video stream during motion of the rig) and hierarchical SfM for unordered image sets. An overview of basic pipelines for both methods is shown in Fig. 4.4 and Fig. 4.5. All modules make use of point correspondences between images

that are created in a preprocessing step via feature detection and matching.

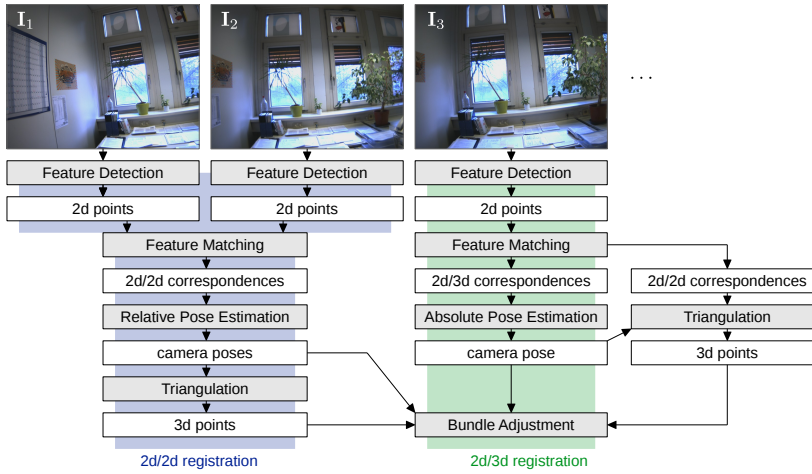

Figure 4.4. Basic pipeline for sequential Structure from Motion, consisting of an initialization phase (left) and a tracking phase (right).

Sequential SfM is usually initialized by solving the relative pose problem for the first two images based on 2d point correspondences that are created using image-based feature detection and matching methods as described in Sec. 4.4. This step is followed by triangulation of 3d points, outlier removal, and initial refinement via bundle adjustment. Subsequent poses are estimated by tracking 2d/3d point correspondences and solving the absolute pose problem with respect to the scene reconstructed so far for each additional image. During pose tracking, the scene is further extended via triangulation from 2d/2d correspondences. Optimization of all parameters via bundle adjustment is used at intermediate steps and as a final step to cope with error accumulation due to the sequential processing ("drift") and to improve the overall quality of the reconstructed 3d scene and camera motion. To prevent drift, feature retrieval (i. e., matching with earlier images) can be added to the SfM pipeline when parts of the scene are visited a second time, potentially leading to "loop closures".

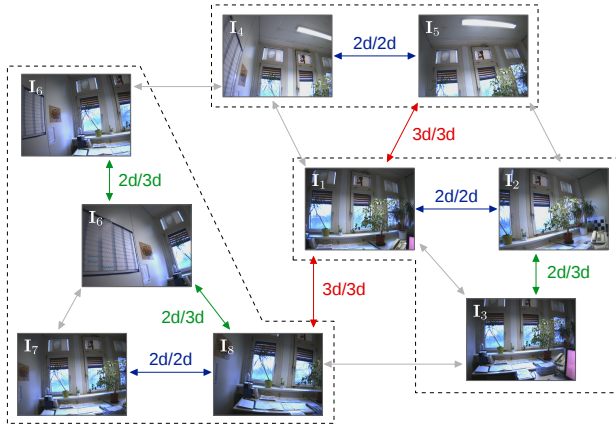

Figure 4.5. Basic pipeline for hierarchical Structure from Motion. Primary clusters are created via 2d/2d registration (blue) and extended via 2d/3d registration (green), clusters are merged via 3d/3d registration (red).

In the hierarchical SfM approach, 2d point correspondences and relative poses are computed first for *all* image pairs in the dataset. Instead of extending the scene sequentially, it is constructed in a bottom-up manner from smaller local reconstructions. Image pairs are processed ordered by a rating that is based on the number, quality, or distribution of 2d correspondences between them. In each step, either a new image cluster is created from an image pair and local 3d points are triangulated, an image is added to an existing cluster from 2d/3d correspondences, or clusters are merged from 3d/3d correspondences until all images have been integrated into the scene. Local and global bundle adjustment is used to refine 3d points and camera poses within clusters and as a final step.

Note that both methods provide a reconstruction that is defined only up to an arbitrary similarity transformation. These ambiguities, referred to as *gauge freedoms* in the literature, are in general solved by fixing the first image as reference point and setting the distance between the first and second camera location to a fixed value.[3]

[3] The reference images are either the first two images in the sequential approach or the

Although quite different with respect to the processing order of images, the basic building blocks of both sequential and hierarchichal SfM are essentially the same: relative pose estimation from 2d/2d correspondences, triangulation of 3d points, absolute pose estimation from 2d/3d correspondences, registration of 3d points, and global refinement of structure and motion. In this section we will introduce these subproblems formally. Practical solutions are described in Appendix B. The following notation is employed: Indices $k = 1, \ldots, m$ or a prime are used to distinguish between different images. Indices $j = 1, \ldots, n$ denote different 3d points. The camera projection function $\mathcal{P}$ is defined as in Sec. 4.2.3. The function d denotes a problem specific error metric, e. g., the Euclidean distance or Mahalanobis distance, if not stated otherwise.

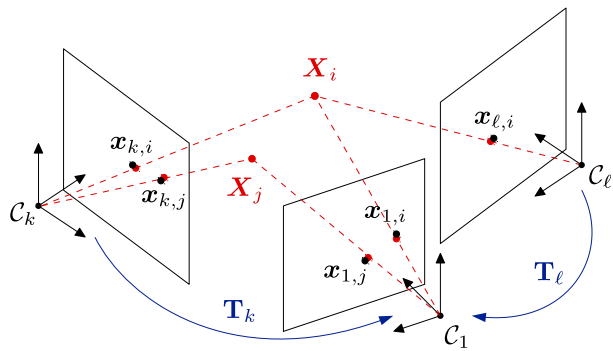

Figure 4.6. Illustration of Structure from Motion from 2d point correspondences for three camera images.

Relative pose problem Given N corresponding normalized 2d points $(\boldsymbol{x}_j, \boldsymbol{x}'_j)$ in two camera images that are projections of unknown 3d points, we seek the relative pose $[\mathbf{R} \mid \boldsymbol{t}]$ and 3d points $\boldsymbol{X}_j$ within the coordinate frame of the first camera minimizing the reprojection error of $\boldsymbol{X}_j$ w.r.t. $\boldsymbol{x}_j$

image pair with highest rating in the hierarchical approach.

and $\boldsymbol{x}'_j$:

$$\min_{\mathbf{R},\boldsymbol{t},\boldsymbol{X}_{1,\dots,N}} \sum_{j=1}^{N} d(\mathcal{P}(\boldsymbol{X}_j), \boldsymbol{x}_j)^2 + d(\mathcal{P}(\mathbf{R}^\intercal(\boldsymbol{X}_j - \boldsymbol{t})), \boldsymbol{x}'_j)^2 \tag{4.7}$$

Note that the relative pose problem has an inherent scale ambiguity. The absolute length of $\boldsymbol{t}$ resp. the absolute mean distance of $\boldsymbol{X}_j$ cannot be recovered unless additional knowledge about either the scene or the relative pose is given. This ambiguity is often resolved by setting $\|\boldsymbol{t}\| = 1$.

To faciliate the relative pose problem, the explicit estimation of 3d points $\boldsymbol{X}_{1,\dots,N}$ is often eliminated from the minimization by considering implicit error metrics for 2d/2d correspondences based on epipolar constraints:

$$\min_{\mathbf{R},\boldsymbol{t}} \sum_{j=1}^{N} d(\boldsymbol{x}_j, \boldsymbol{x}'_j; \mathbf{E})^2 \tag{4.8}$$

where $\mathbf{E} \sim [\boldsymbol{t}]_\times \mathbf{R}$ is the *essential matrix* defined by the epipolar constraints $\boldsymbol{x}_j^\intercal \mathbf{E} \boldsymbol{x}'_j = 0$ [Lon81]. Once estimated, the essential matrix can be decomposed into rotation and translation via SVD. However, the resulting relative translation is in general only defined up to an unknown scale factor.

Equation (4.7) is approached stepwise then: First, eq. (4.8) is solved for the relative pose, afterwards 3d points are estimated via triangulation. The final solution can be refined as a two-view instance of the Structure from Motion problem via bundle adjustment. Further details on relative pose estimation based on the essential matrix can be found in B.2.

Absolute pose problem Given N normalized 2d points $\boldsymbol{x}_j$ corresponding to known 3d points $\boldsymbol{X}_j$, we seek the absolute pose $[\mathbf{R} \mid \boldsymbol{t}]$ minimizing the reprojection error of $\boldsymbol{X}_j$ w.r.t. $\boldsymbol{x}_j$:

$$\min_{\mathbf{R},\boldsymbol{t}} \sum_{j=1}^{N} d(\mathcal{P}(\mathbf{R}^\intercal(\boldsymbol{X}_j - \boldsymbol{t})), \boldsymbol{x}_j)^2 \tag{4.9}$$

Direct and iterative solutions for the absolute problem can be found in B.3.

Triangulation problem Given normalized 2d points $\boldsymbol{x}_k$ corresponding to the same unknown 3d point in m camera images with known absolute poses $[\mathbf{R}_k \mid \boldsymbol{t}_k]$, we seek the 3d point $\boldsymbol{X}$ minimizing the reprojection error w.r.t. $\boldsymbol{x}_k$:

$$\min_{\boldsymbol{X}} \sum_{k=1}^{m} d(\mathcal{P}(\mathbf{R}_k^\intercal(\boldsymbol{X} - \boldsymbol{t}_k)), \boldsymbol{x}_k)^2 \tag{4.10}$$

Numerical solutions for the triangulation problem such as the *mid-point method* are described in B.4.

3d/3d registration problem Given N 3d points $\boldsymbol{X}_j$ in the first camera coordinate frame $\mathcal{C}$ and corresponding 3d points $\boldsymbol{X}'_j$ in the second camera coordinate frame $\mathcal{C}'$, we seek the Euclidean transformation $[\mathbf{R} \mid \boldsymbol{t}]$ aligning both sets of 3d points:[4]

$$\min_{\mathbf{R},\boldsymbol{t}} \sum_{j=1}^{N} d(\boldsymbol{X}_j, \mathbf{R}\boldsymbol{X}'_j + \boldsymbol{t})^2 \tag{4.11}$$

Numerical solutions for 3d point alignment can be found in A.3.

Structure from Motion problem Given normalized 2d points $\boldsymbol{x}_{k,j}$ for $k = 1, \ldots, m$ and $j = 1, \ldots, N$ where $\boldsymbol{x}_{k,j}$ is the projection of the j-th 3d point into the k-th camera image, we seek camera poses $[\mathbf{R}_k \mid \boldsymbol{t}_k]$ and 3d points $\boldsymbol{X}_j$ minimizing the reprojection error for all points in all images:

$$\min_{\substack{\mathbf{R}_{1,\ldots,m}, \boldsymbol{t}_{1,\ldots,m}, \\ \boldsymbol{X}_{1,\ldots,N}}} \sum_{k=1}^{m} \sum_{j=1}^{N} d(\mathcal{P}(\mathbf{R}_k(\boldsymbol{X}_j - \boldsymbol{t}_k)), \boldsymbol{x}_{k,j})^2 \tag{4.12}$$

In general, 3d points are not visible in all camera images. In this case, we consider only indices $(k, j) \in \mathcal{V}$ for a subset $\mathcal{V} \subset \{1, \ldots, m\} \times \{1, \ldots, N\}$

[4] This problem is often referred to as *absolute orientation problem*. However, we avoid this term here to prevent confusion with the 2d/3d absolute pose problem.

denoted as the *visibility set*, replacing the sum $\sum_{k=1}^{m}\sum_{j=1}^{N}$ in eq. (4.12) by $\sum_{(k,j)\in\mathcal{V}}$. The number of observations is $M = |\mathcal{V}| \leqslant mN$.

Similar to the relative pose problem, the solution to eq. (4.12) exhibits an inherent absolute pose and scale ambiguity that must be fixed in an appropriate way, e. g., by fixing the first camera pose and the distance between the first and second camera.

The numerical solution to the full Structure from Motion problem is in general computed with *bundle adjustment* (see B.5). Initial solutions for absolute poses and 3d points are estimated in a multi-step approach consisting of alternating relative pose estimation, triangulation, and absolute pose estimation.

4.6 Summary

In this chapter we described the basic building blocks of monocular Structure from Motion using calibrated cameras. Existing direct methods to solve the relative and absolute pose problems were described that are used to provide initial parameters for a large-scale nonlinear optimization problem refining both the estimated 3d structure and camera poses w.r.t. general error measures. In the following chapter we will describe how to bring SfM-based ego-motion estimation and eye-to-eye calibration together.

Chapter 5

Extrinsic Calibration From Non-Overlapping Images

In this chapter we will extend monocular Structure from Motion as described in the previous chapter by simultaneous pose estimation of rigidly coupled cameras, eye-to-eye calibration, and global refinement. First, we will briefly formalize the eye-to-eye calibration problem based on visual observations instead of rigidly coupled local poses. Then we will present the major novel contributions of our work: First, a general multi-camera Structure from Motion approach with eye-to-eye calibration is presented, including a rigidly coupled bundle adjustment for joint optimization of camera motion, 3d structure, and eye-to-eye transformation parameters.[1] Afterwards, we describe how to stabilize monocular SfM of rigidly coupled camera prior to eye-to-eye calibration by enforcing partial rigid motion constraints.[2] As a final contribution, we will consider global consistency of extrinsic calibration for rigs consisting of many cameras, described under the term *complete eye-to-eye calibration*.

5.1 Problem Statement

We use in general the same notations as in Sec. 4.5 here to define the *rigidly coupled Structure from Motion* problem. Taking on the notations

[1] [EG10] S. Esquivel, S. Gehrig: "Entwicklung eines Kalibrierverfahrens für fahrzeugmontierte Mehrkamerasysteme", final project report for AKTIV-AS/KAS, Daimler AG, 07/2010

[2] [EK13] S. Esquivel, R. Koch: "Structure from Motion using rigidly coupled cameras without overlapping views", Proc. of GCPR'13, 09/2013

from Part I, a superscript (i) is used to distinguish between poses, 3d points, and 2d points for different cameras resp. properties of the second camera are denoted by a prime in the case of two rigidly coupled cameras. W.l.o.g. we define the first camera as the reference camera for eye-to-eye calibration. Hence, we will write $\Delta\mathbf{T}_i$ instead of $\Delta\mathbf{T}_{1,i}$ for sake of simplicity. $\Delta\mathbf{T}_1$ is implicitly defined as $[\mathbf{I} \mid \mathbf{0}]$ and $\Delta\mathbf{T}_{i,j} = \Delta\mathbf{T}_i^{-1}\Delta\mathbf{T}_j$ describes the eye-to-eye transformation between the i-th and j-th camera. Eye-to-eye transformations $\Delta\mathbf{T}_i$ are in general similarity transformations in this context although the relative scale parameters $\Delta\lambda_i$ might be fixed.

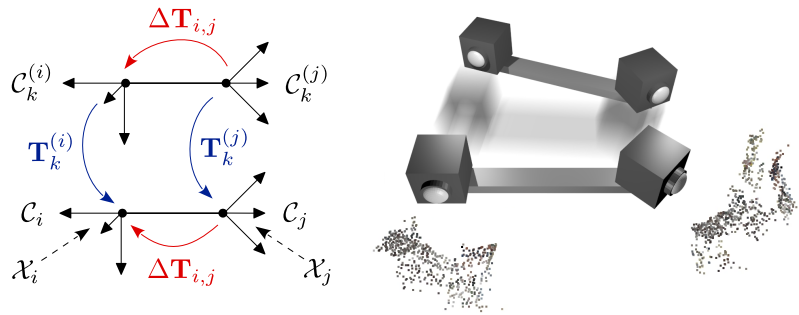

Figure 5.1. Local coordinate frames and poses of two rigidly coupled cameras moving through separate scenes reconstructed via Structure from Motion.

Visual eye-to-eye calibration problem Consider two rigidly coupled cameras that capture m images at different positions simultaneously. The initial pose for each camera is considered w.l.o.g. as the reference pose, defining the local coordinate frames $\mathcal{C}$, $\mathcal{C}'$. Given are 2d points $\boldsymbol{u}_{k,j}, (k,j) \in \mathcal{V}$ where $\boldsymbol{u}_{k,j}$ is the projection of the j-th 3d point $\boldsymbol{X}_j$ into the k-th image of the first camera for the visibility set $\mathcal{V} \subset \{1,\ldots,m\} \times \{1,\ldots,N\}$. Independent 2d/3d points for the second camera are given likewise. In contrast to geometric eye-to-eye calibration where distance measures between predicted poses $\hat{\mathbf{T}}'_k = \Delta\mathbf{T}^{-1}\mathbf{T}_k\Delta\mathbf{T}$ and measured poses $\mathbf{T}'_k$ are evaluated, *visual eye-to-eye calibration* is based on the reprojection errors resulting from

these:

$$\min_{\Delta\mathbf{T}} \sum_{(k,j)\in\mathcal{V}} d(\boldsymbol{u}_{k,j}, {\mathbf{T}_k}^{-1}\boldsymbol{X}_j)^2 + \sum_{(k,j)\in\mathcal{V}'} d'(\boldsymbol{u}'_{k,j}, (\underbrace{\Delta\mathbf{T}^{-1}\mathbf{T}_k\Delta\mathbf{T}}_{\hat{\mathbf{T}}'_k})^{-1}\boldsymbol{X}'_j)^2 \tag{5.1}$$

where d and d' are camera-specific error metrics between 2d/3d points. Since the reconstructions of rigidly coupled cameras are a priori disjunct, eq. (5.1) is in general only solved for two cameras of the rig at a time, providing pair-wise eye-to-eye transformations. Solutions of eq. (5.1) will be described in Sec. 5.3.1.

Eye-to-Eye Structure from Motion problem Consider a multi-camera system consisting of n rigidly coupled cameras. Given are 2d points $\boldsymbol{u}^{(i)}_{k,j}, (k,j) \in \mathcal{V}_i$ and 3d point sets $\mathcal{X}_i = \{\boldsymbol{X}^{(i)}_j \mid j = 1,\ldots,N_i\}$ where N_i is the number of 3d points in the i-th scene part. The set $\mathcal{X} = \bigcup_{i=1}^{n} \mathcal{X}_i$ contains the parameters of 3d points for all scene parts.

In the most general formulation of the *Eye-to-Eye Structure from Motion problem*, we seek rigidly coupled poses $\mathbf{T}^{(i)}_k$ in $\mathcal{C}_i$, the respective eye-to-eye transformations $\Delta\mathbf{T}_i$ between the i-th and first camera, and local 3d points for each scene part minimizing the joint reprojection error:

$$\min_{\substack{\Delta\mathbf{T}_{2,\ldots,n} \\ \mathbf{T}^{(1,\ldots,n)}_{1,\ldots,m}, \mathcal{X}}} \sum_{i=1}^{n} \sum_{(k,j)\in\mathcal{V}_i} d_i(\boldsymbol{u}^{(i)}_{k,j}, {\mathbf{T}^{(i)}_k}^{-1}\boldsymbol{X}^{(i)}_j)^2 \tag{5.2}$$

subject to the constraints

$$\mathbf{T}^{(i)}_k \Delta\mathbf{T}_{i,j} = \Delta\mathbf{T}_{i,j}\mathbf{T}^{(j)}_k \quad \text{for all } 1 \leqslant i < j \leqslant n$$

As before, $d_{1,\ldots,n}$ denote 2d/3d error metrics depending on the actual camera model of the i-th camera. We assume that the intrinsic camera parameters defining these measures are known.

To make this problem feasible, rigid motion constraints are enforced explicitly by estimating poses $\mathbf{T}_k = \mathbf{T}_k^{(1)}$ of the first camera only and predicting poses of the coupled cameras from $\mathbf{T}_k$ and $\Delta\mathbf{T}_i$:

$$\min_{\substack{\Delta\mathbf{T}_{2,\dots,n} \\ \mathbf{T}_{1,\dots,m},\mathcal{X}}} \sum_{i=1}^{n} \sum_{(k,j)\in\mathcal{V}_i} d_i(\boldsymbol{u}_{k,j}^{(i)}, \underbrace{(\Delta\mathbf{T}_i^{-1}\mathbf{T}_k\Delta\mathbf{T}_i)}_{\hat{\mathbf{T}}_k^{(i)}})^{-1}\boldsymbol{X}_j^{(i)})^2 \tag{5.3}$$

where $\hat{\mathbf{T}}_k^{(i)}$ is the prediction of the k-th local pose of the i-th camera from the k-th pose of the first camera and the i-th eye-to-eye transformation.[3] This strategy requires that the initial solutions for eye-to-eye transformations $\Delta\mathbf{T}_{2,\dots,n}$ already fulfill a certain degree of global consistency.

We can facilitate eq. (5.3) further by transforming the poses and 3d points for all cameras from $\mathcal{C}_i$ into the first camera coordinate frame $\mathcal{C}_1$ via $\Delta\mathbf{T}_i$:

$$\min_{\substack{\Delta\mathbf{T}_{2,\dots,n} \\ \mathbf{T}_{1,\dots,m},\mathcal{X}*}} \sum_{i=1}^{n} \sum_{(k,j)\in\mathcal{V}_i} d_i(\boldsymbol{u}_{k,j}^{(i)}, \underbrace{(\mathbf{T}_k\Delta\mathbf{T}_i)}_{\hat{\mathbf{T}}_k^{*(i)}})^{-1}\boldsymbol{X}_j^{*(i)})^2 \tag{5.4}$$

where the transformed 3d points $\boldsymbol{X}_j^{*(i)} = \Delta\mathbf{T}_i\boldsymbol{X}_j^{(i)}$ and transformed predicted poses $\hat{\mathbf{T}}_k^{*(i)}$ are relative to $\mathcal{C}_1$, thus effectively combining the individual reconstructions.

In this chapter, we will propose a modified bundle adjustment approach to solve the Eye-to-Eye Structure from Motion problem. In practical applications, the number of parameters involved grows quite large. The complexity of the problem can be reduced by considering only two cameras in the rig at a time, e. g., when this approach is used as a refinement step for pair-wise visual eye-to-eye calibration.

Eye-and-World Structure from Motion problem The *Eye-and-World Structure from Motion* problem can be formulated similar to eq. (5.3) by considering an unknown world-to-world transformation $\Delta\mathbf{W}_i$ between the i-th

[3] Note that $\hat{\mathbf{T}}_k^{(1)}$ is simply given by $\mathbf{T}_k$.

and first world coordinate frames $\mathcal{W}_i$ and $\mathcal{W}_1$. Here, all camera poses $\mathbf{W}_k^{(i)}$ and 3d points $\boldsymbol{X}_j^{(i)}$ are assumed to be measured within $\mathcal{W}_i$ respectively:

$$\min_{\substack{\Delta\mathbf{T}_{2,\dots,n} \\ \Delta\mathbf{W}_{2,\dots,n} \\ \mathbf{W}_{1,\dots,m},\mathcal{X}}} \sum_{i=1}^{n} \sum_{(k,j)\in\mathcal{V}_i} d_i(\boldsymbol{u}_{k,j}^{(i)}, (\underbrace{\Delta\mathbf{W}_i^{-1}\mathbf{W}_k\Delta\mathbf{T}_i}_{\hat{\mathbf{W}}_k^{(i)}})^{-1}\boldsymbol{X}_j^{(i)})^2 \tag{5.5}$$

Similar to eq. (5.4), the world-to-world transformations are absorbed by the 3d points. Hence, the problem can be alleviated analogously by considering 3d points and camera poses within the global world coordinate frame instead:

$$\min_{\substack{\Delta\mathbf{T}_{2,\dots,n} \\ \mathbf{W}_{1,\dots,m},\mathcal{X}^*}} \sum_{i=1}^{n} \sum_{(k,j)\in\mathcal{V}_i} d_i(\boldsymbol{u}_{k,j}^{(i)}, (\underbrace{\mathbf{W}_k\Delta\mathbf{T}_i}_{\hat{\mathbf{W}}_k^{*(i)}})^{-1}\boldsymbol{X}_j^{*(i)})^2 \tag{5.6}$$

where the transformed 3d points $\boldsymbol{X}_j^{*(i)} = \Delta\mathbf{W}_i\boldsymbol{X}_j^{(i)}$ and transformed predicted poses $\hat{\mathbf{W}}_k^{*(i)}$ are measured within $\mathcal{W}_1$.

SfM with known eye-to-eye transformation Given that the eye-to-eye transformations are known, the accuracy and robustness of Structure from Motion can be significantly improved by using a virtual camera as a representation of the multi-camera rig with compound camera function [Ple03; FKK04]:

$$\mathcal{K}_i^* : \mathbb{R}^3 \to \mathbb{P}^2, \boldsymbol{X} \mapsto \mathcal{K}_i(\Delta\mathbf{R}_i(\boldsymbol{X} - \Delta\boldsymbol{t}_i)) \tag{5.7}$$

due to the large combined field of view, alleviating the impact of critical motions, preventing drift, and providing a consistent scale over time. This has been proved by the works of Kim et al. [KC06; KLH10] and Clipp et al. [Cli+08] amongst others.

5.2 Related Work

In classical hand-eye calibration, computation of the poses of the "eye" from images of a calibration object and estimation of the relative pose between camera and gripper based on relative pose measurements is often decoupled as in our *geometric eye-to-eye calibration* approach. However, there are some publications on estimating the hand-eye transformation with respect to the reprojection error of the calibration object in the camera images, denoted as "visual" hand-eye calibration here.

Among the first, Horaud & Dornaika [HD95] stated that the inevitable separation of intrinsic and extrinsic camera parameters to obtain $\mathbf{A}$ introduces errors in $\mathbf{X}$. The authors advised to solve the equation $\mathbf{M_2Y} = \mathbf{M_1YB}$ instead where $\mathbf{M}_1$ and $\mathbf{M}_2$ are the 3×4 projection matrices of the eye at the initial and second location. $\mathbf{Y}$ denotes the unknown homogeneous transformation matrix from the hand's coordinate frame to the camera calibration frame. For fixed and known intrinsic camera parameters and $\mathbf{M}_1 = \mathbf{KI}$, $\mathbf{M}_2 = \mathbf{KA}$, this formulation leads to $\mathbf{KAY} = \mathbf{KYB}$ where $\mathbf{Y}$ is identical with $\mathbf{X}$ from the classical formulation in [SA89]. However, Zhuang notes in [Zhu97] that the equation in [HD95] is only superior to the classical formulation when a simplistic pinhole model is used for the camera. More complex intrinsic calibration methods yield far better results for hand-eye calibration according to [SA89] as shown in [HD95], hence the classical formulation evaluating pose observations rather than image measurements is still prevalent in the literature.

Zhuang, Wang & Roth extend Horaud & Dornaika's approach by a more complex camera function taking also image distortion into account and estimate the intrinsic parameters of the camera simultaneously with the extrinsic parameters. Wei, Arbter & Hirzinger [WAH98] adapt self-calibration methods to get rid of known calibration objects and estimate the positions of tracked 3d points simultaneously with intrinsic and extrinsic camera parameters. In contrast to the previous methods, they explicitly decouple intrinsic and extrinsic parameters in the estimation process, parametrizing camera rotations with Euler angles. Zhao & Liu [ZL08] describe a method for simultaneous hand-eye calibration and intrinsic camera calibration

using images of a known plane.

Stewénius & Åström [SÅ04] describe a feature-point based approach using a calibrated camera and eliminate the explicit estimation of 3d points by deriving an error function from multilinear constraints based on the epipolar geometry. However, their approach requires pure translations.

Malm & Heyden [MH00] describe a visual hand-eye calibration approach based on optical flow instead of 2d/2d point correspondences that is also dependent on images of calibration objects with known geometry.

Visual hand-eye calibration minimizing the L_∞ norm of reprojection errors approaching globally optimal solutions have been proposed by Heller et al. [Hel+11] and Ruland et al. [RPK11] for the case of known hand-eye rotation, Seo et al. [SCL09] for the case of known hand-eye translations, and Ruland et al. [RPK12] estimating both hand-eye rotation and translation using branch-and-bound algorithms.

However, all of the methods described above predict poses of the camera given accurate pose measurements of the "hand" (either via kinematic chains of a robot arm or odometry data in the case of a vehicle-mounted camera) and minimize error functions based on the reprojection error w.r.t. hand-eye transformation parameters and optionally camera intrinsics and 3d points in the observed scene. For eye-to-eye calibration, in contrast, poses of both rigidly coupled sensors have to be estimated from visual measurements concurrently.

In [EG10], we described nonlinear optimization of extended eye-to-eye transformation parameters from 2d/2d correspondences using joint local bundle adjustment for rigidly coupled cameras, minimizing the error function (5.3). Local camera motion and 3d scenes used as input are generated via sequential Structure from Motion from synchronously captured video streams. Based on our work [EWK07], Lébraly et al. [Léb+10; Léb+11] published a similar approach to rigidly coupled bundle adjustment w.r.t. eq. (5.3), providing an efficient implementation by taking the sparsity of the Jacobian matrix of the joint error function into account. In their evaluation, however, they use calibration objects to determine camera poses with absolute scale instead of Structure from Motion techniques.

5.3 Eye-to-Eye Calibration Using SfM

In Chapter 3 we described eye-to-eye calibration from rigidly coupled local motions that have been acquired in a preprocessing step. Given that the camera poses are measured with different scales, extended eye-to-eye calibration methods are used to recover the relative scale and provide the proper distance between the coupled cameras w.r.t. the first camera's reference coordinate frame.

This approach can be adapted in a straightforward way to use poses computed via individual Structure from Motion for multiple cameras as input:

- ▻ First, capture images with all cameras simultaneously during motion of the camera rig.
- ▻ Compute local poses $\mathbf{T}_k^{(i)}$ for each camera using separate Structure from Motion pipelines as described in Sec. 4.5. To facilitate this task we assume that the intrinsic camera parameters are known. The resulting reconstructions have potentially different scaling w.r.t. each other.
- ▻ Afterwards, compute pairwise eye-to-eye transformations between the i-th and the reference camera (i. e., the first camera) from corresponding relative poses $\mathbf{T}_{k,\ell}^{(1)}$, $\mathbf{T}_{k,\ell}^{(i)}$ for suitable image pairs (k, ℓ) selected according to the strategies described in Sec. 3.7.1. Note that the resulting eye-to-eye translations are scaled w.r.t. the reference coordinate frame $\mathcal{C}_1$.
- ▻ Refine the results from geometric eye-to-eye calibration via *visual eye-to-eye calibration* minimizing the joint reprojection error for the i-th and the reference camera (see Sec. 5.3.1).
- ▻ Refine the estimated camera poses, 3d points, and eye-to-eye transformation for each camera pair $(1, i)$ using rigidly coupled bundle adjustment. This step is denoted as *eye-to-eye bundle adjustment* (see Sec. 5.5).
- ▻ For the case of $n > 2$ cameras, simultaneous eye-to-eye bundle adjustment of all reconstructions can be used as a postprocessing step to

improve all results and achieve global consistency. This step is denoted as *global eye-to-eye bundle adjustment*.

This establishes the basic pipeline for *Eye-to-Eye Structure from Motion* as depicted in Fig. 5.5 for sequential Structure from Motion with two rigidly coupled cameras.

Figure 5.2 gives an overview of the main stages for two rigidly coupled cameras. As an optional postprocessing step, the 3d points of all cameras transformed into C_1 can be globally aligned using rigid 3d point set alignment methods without correspondences[4] (e. g., via *RPM* [Gol+98]), given that there is significant overlap between the reconstructed scenes, and repeat eye-to-eye bundle adjustment after merging corresponding 3d points across different scene parts.

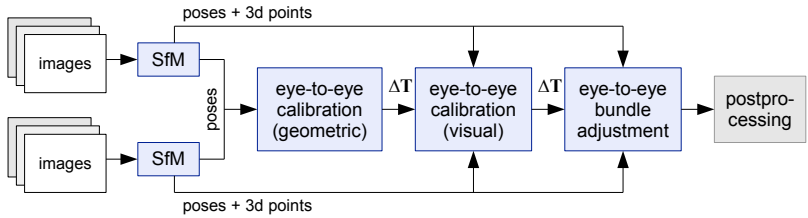

Figure 5.2. Basic overview of decoupled Eye-to-Eye Structure from Motion algorithm for two rigidly coupled cameras.

Integrated approach In the basic pipeline, Structure from Motion and eye-to-eye calibration are performed in two different stages: SfM is used as a preprocessing step to create input for geometric and visual eye-to-eye calibration. Afterwards, the results of the previous stages are refined jointly using eye-to-eye bundle adjustment. Therefore, this approach is referred to as *decoupled Eye-to-Eye SfM*. As an alternative, eye-to-eye calibration can be directly integrated into the Structure from Motion pipeline, estimating the eye-to-eye transformation incrementally during parallel

[4] Note that this is inherently different from the *3d/3d registration problem* in Sec. 4.5.

scene reconstruction of both cameras (see Fig. 5.3). This approach, denoted as *integrated Eye-to-Eye SfM*, poses some restrictions on the individual scene reconstruction threads, since relative poses must be processed in the same sequential order by both. Therefore, this approach is especially convenient for sequential Strucure from Motion.

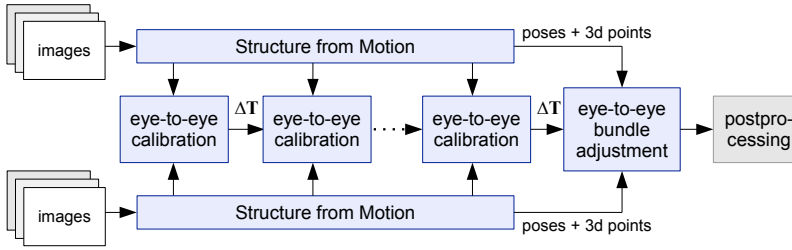

Figure 5.3. Basic overview of integrated Eye-to-Eye Structure from Motion algorithm for two rigidly coupled cameras.

Global consistency The basic pipeline assumes that a specific camera is selected as reference camera for eye-to-eye calibration, per se leading to globally consistent eye-to-eye transformations that can be used in a final global eye-to-eye bundle adjustment step. However, the choice of the reference camera might not be self-evident. For camera rigs with $m > 2$, an alternative is to compute pairwise eye-to-eye transformations $\Delta\mathbf{T}_{i,j}$ between all cameras and enforce global consistency of the estimated parameters based on geometric constraints prior to global bundle adjustment, or instead of global bundle adjustment for applications where only pair-wise eye-to-eye calibration is feasible. This problem is denoted as *complete eye-to-eye calibration* in this work. Details and solutions can be found in Sec. 5.6. Figure 5.4 gives an overview of the different stages for three rigidly coupled cameras, yielding up to 6 different pair-wise eye-to-eye transformations $\Delta\mathbf{T}_{i,j}$.

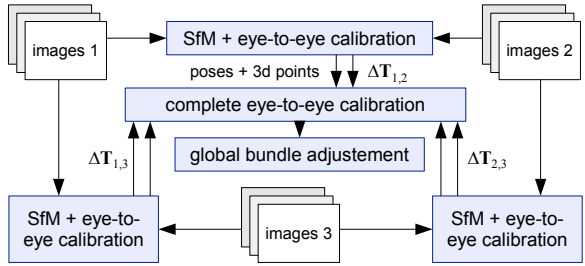

Figure 5.4. Basic overview of complete Eye-to-Eye Structure from Motion algorithm for three rigidly coupled cameras.

5.3.1 Eye-to-Eye Calibration from 2D/3D Correspondences

In this section we describe a numerical solution to eq. (5.1). The approach is similar to the method described by Wei, Arbter & Hirzinger [WAH98] for hand-eye calibration where poses of the camera are predicted from the hand-eye transformation and the poses of the gripper.

The general idea is to minimize the joint reprojection error resulting from 2d/3d correspondences of rigidly coupled cameras at the same time w.r.t. the eye-to-eye transformation parameters. Hence, in contrast to Wei, Arbter & Hirzinger, we use reprojection error terms for both cameras, predicting the pose of the first camera from the second and vice versa, i. e., eq. (5.1) is minimized w.r.t. the symmetric reprojection error. While in [WAH98] Euler angles are used to parametrize the hand-eye rotation, we describe the eye-to-eye transformation $\Delta\mathbf{T}$ by a quaternion $\Delta\mathbf{q}$ and translation vector $\Delta\boldsymbol{t}$ instead, following the considerations on motion parametrization in Part I. This results in the following nonlinear error function:

$$\begin{aligned}\min_{\Delta\mathbf{q},\Delta\boldsymbol{t}} \sum_{(k,j)\in\mathcal{V}} d(\boldsymbol{u}_{k,j}, \hat{\mathbf{R}}_k^\mathsf{T}(\boldsymbol{X}_j - \hat{\boldsymbol{t}}_k))^2 + \\ \sum_{(k,j)\in\mathcal{V}'} d'(\boldsymbol{u}'_{k,j}, \hat{\mathbf{R}}'^\mathsf{T}_k(\boldsymbol{X}'_j - \hat{\boldsymbol{t}}'_k))^2\end{aligned} \tag{5.8}$$

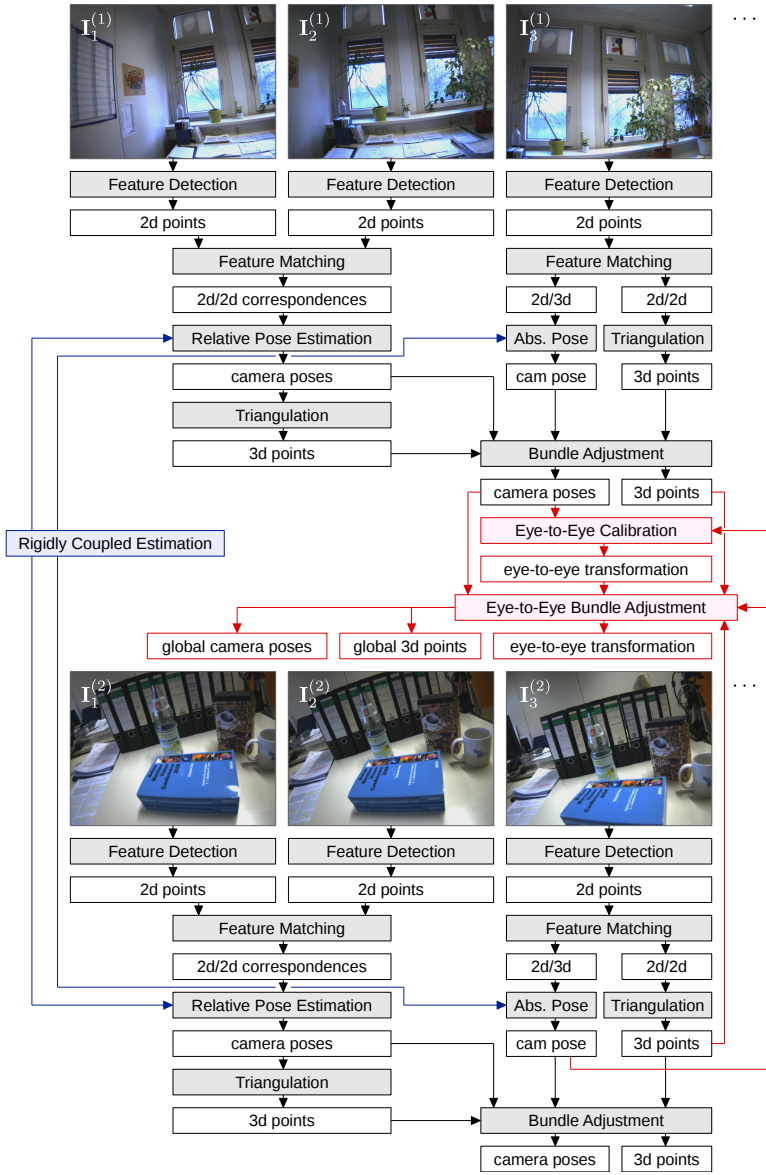

Figure 5.5. Basic pipeline for sequential Eye-to-Eye Structure from Motion for two rigidly coupled cameras.

with poses predicted from measured poses and eye-to-eye transformation:

$$\begin{aligned}\hat{\mathbf{R}}_k &= \Delta\lambda^{-2}\Delta\mathbf{R}\mathbf{R}'_k\Delta\mathbf{R}^\mathsf{T} \\ \hat{\boldsymbol{t}}_k &= (\mathbf{I} - \Delta\lambda^{-2}\Delta\mathbf{R}\mathbf{R}'_k\Delta\mathbf{R}^\mathsf{T})\Delta\boldsymbol{t} + \Delta\mathbf{R}\boldsymbol{t}'_k \\ \hat{\mathbf{R}}'_k &= \Delta\lambda^{-2}\Delta\mathbf{R}^\mathsf{T}\mathbf{R}_k\Delta\mathbf{R} \\ \hat{\boldsymbol{t}}'_k &= \Delta\lambda^{-2}\Delta\mathbf{R}^\mathsf{T}((\mathbf{R}_k - \mathbf{I})\Delta\boldsymbol{t} + \boldsymbol{t}_k)\end{aligned}$$

where $\Delta\mathbf{R} = \mathbf{R}_{\Delta\mathbf{q}}$ and $\Delta\lambda = \Delta\mathbf{q}^\mathsf{T}\Delta\mathbf{q}$.

The residuals of the error functions are most commonly described by the *image reprojection error*:

$$d(\boldsymbol{u}, \boldsymbol{X}) = \|\mathcal{K}(\boldsymbol{X}) - \boldsymbol{u}\|$$

given that the camera functions $\mathcal{K}$ and $\mathcal{K}'$ are known.[5]

Note that $\Delta\mathbf{T}$ represents a similarity transformation due to the potentially different scales of $\mathcal{C}$ and $\mathcal{C}'$. The scaling factor $\Delta\lambda$ from $\mathcal{C}'$ to $\mathcal{C}$ is encoded by the length of the quaternion $\Delta\mathbf{q}$ as in Sec. 3.5.2.[6]

Given an initial solution for $\Delta\mathbf{T}$ resulting from geometric eye-to-eye calibration, eq. (5.8) is solved using nonlinear optimization methods such as the Levenberg-Marquardt algorithm as in [WAH98]. This methods is referred to as *visual eye-to-eye calibration* or briefly as `vE2E` in the remainder of this work.

Combining different camera models Since the camera functions are assumed to be known, the computational complexity of eq. (5.8) can be reduced by considering normalized 2d points $\boldsymbol{x} = \mathcal{U}(\boldsymbol{u})$ instead of actual pixel coordinates. However, the residuals for both cameras should be weighted with respect to the expected error magnitude in this case to account for different image resolutions and camera models.

[5] We assume in general that undistorted pixel coordinates $\mathcal{D}^{-1}(\boldsymbol{u})$ are used for Structure from Motion to reduce the complexity of the camera function.

[6] If the coordinate frames are known to have equal scale, the factor $\Delta\lambda$ can be omitted and eq. (5.8) has to be minimized subject to $\Delta\mathbf{q}^\mathsf{T}\Delta\mathbf{q} = 1$ instead.

For a planar camera satisfying the simplified pinhole camera model eq. (4.2), a reasonable error function is given by the Euclidean distance between image points in the normalized image plane, i. e., the *normalized reprojection error*, weighted by the focal length f:

$$d(\boldsymbol{x}, \boldsymbol{X}) = f \left\| \mathcal{P}(\boldsymbol{X}) - \boldsymbol{x} \right\| \quad \text{where } \boldsymbol{x} = \mathcal{U}_{\mathcal{P}}(\boldsymbol{u})$$

For a spherical camera described by eq. (B.1), a comparable error measure is given by the scaled *spherical reprojection error*:

$$d(\boldsymbol{v}, \boldsymbol{X}) = f \left\| \mathcal{S}(\boldsymbol{X}) - \boldsymbol{v} \right\| \quad \text{where } \boldsymbol{v} = \mathcal{U}_{\mathcal{S}}(\boldsymbol{u})$$

approximating the actual image reprojection error. An overview of error measures used for pose estimation from 2d/3d correspondences can be found in A.4.2. In the remainder of this work, we will restrict the problem formulation to planar cameras for sake of conciseness.

5.4 SfM with Partial Rigid Motion Constraints

Using sequential Structure from Motion techniques to compute camera ego-motion separately typically suffers from drift, i. e., pose errors accumulating over time. Estimating the poses for each camera individually in the initial phase of our SfM-based eye-to-eye calibration pipeline does not pay respect to the rigid coupling constraints described in Sec. 3.3. Although the geometric eye-to-eye calibration methods described in Chapter 3 are able to handle non-rigidly constrained motion up to a certain degree, the solution deteriorates significantly for systematic errors of the input poses as we have seen in the last test case in Sec. 3.8.1 (see Fig. 3.14). It should be helpful to incorporate the rigid motion constraints already into the Structure from Motion step to obtain more stable results suitable for eye-to-eye calibration. We also assume that the effect of drift can be counteracted by this approach. However, this task seems like a chicken-and-egg problem at first sight, since the rigid coupling parameters are not known yet in this stage.

In the following section we extend the relative and absolute pose estimation methods described in B.2 and B.3 to enforce partial rigid motion constraints prior to eye-to-eye calibration. This technique was originally published by us in [EK13].

Enforcing partial rigid motion constraints Consider rigidly coupled poses $\mathbf{T}_k^{(i)}$ for $i = 1, \ldots, n$ cameras with respect to the respective reference coordinate frames $\mathcal{C}_i$ that have been estimated for the k-th image during sequential Structure from Motion, either from 2d/2d correspondences $(\boldsymbol{u}_{1,j}^{(i)}, \boldsymbol{u}_{k,j}^{(i)})$, $j = 1, \ldots, N_i$ in the initialization phase or from 2d/3d correspondences $(\boldsymbol{u}_{k,j}^{(i)}, \boldsymbol{X}_{k,j}^{(i)})$, $(k, j) \in \mathcal{V}_i$ in the tracking phase.

As stated in the Screw Congruence Theorem (Lemma 3.3), all cameras undergo relative rotation around different local rotation axes $\boldsymbol{r}_k^{(i)}$ (related to each other by the unknown eye-to-eye rotations $\Delta\mathbf{R}_{i,j}$) by the same absolute rotation angle $\alpha_k^{(i)}$ and have equal absolute pitch $d_k^{(i)}$, i. e., magnitude of translation along the rotation axis. Hence, in simultaneous nonlinear refinement of all pose parameters shared parameters α_k^*, d_k^* can be used for the rotation angle and pitch, assuming that motion is parametrized in a manner that exhibits these values. Note that the equal pitch constraint can only be applied in absolute pose estimation and cannot be used for planar motion or pure translation of the cameras. In the latter case, the equal translation length constraint (3.15) could be used instead.

This approach can be implemented by parametrizing rotations by unit quaternions $\Delta\mathbf{q}_k^{(i)}$ in the relative pose problem resp. motions by dual quaternions $\Delta\check{\mathbf{q}}_k^{(i)}$ in the absolute pose problem with shared parameters q_k^*, p_k^* for the scalar part, encoding rotation angle and pitch as $q_k^* = \cos(\frac{\alpha*_k}{2})$, $p_k^* = -\frac{d_k^*}{2}\sin(\frac{\alpha_k^*}{2})$. Initial solutions can be obtained from averaging the rotation angles and pitch values of input poses or fixing the values of the reference camera.

Relative pose problem This leads to the following joint relative pose estimation from 2d/2d correspondences:

$$\min_{q,\boldsymbol{q}_{1,\ldots,n},\boldsymbol{t}_{1,\ldots,n}} \sum_{i=1}^{n} \sum_{j=1}^{N_i} d_i(\boldsymbol{x}_{1,j}^{(i)}, \boldsymbol{x}_{k,j}^{(i)}; \mathbf{E}_{(\boldsymbol{q}_i,q),\boldsymbol{t}_i})^2 \tag{5.9}$$

subject to $\boldsymbol{q}_i^\top \boldsymbol{q}_i + q^2 = 1$ and $\|\boldsymbol{t}_i\| = 1$ for all $i = 1, \ldots, n$, where d_i defines an error measure for corresponding normalized 2d points $\boldsymbol{x} = \mathcal{U}(\boldsymbol{u})$ of the i-th camera with respect to an essential matrix $\mathbf{E}$ (see A.4.1). We follow the advice in [HZ04] and use the *geometric epipolar distance*:

$$d(\boldsymbol{x}, \boldsymbol{x}'; \mathbf{E}) = f \frac{\boldsymbol{x}^\top \mathbf{E} \boldsymbol{x}'}{\|\mathbf{E}_{[1\ldots2]} \boldsymbol{x}'\|}$$

scaled by the focal length f to provide comparable measures for different image resolutions.

The essential matrix parametrized by a quaternion $\mathbf{q}$ and translation vector $\boldsymbol{t}$ is given by $\mathbf{E}_{\mathbf{q},t} = [\boldsymbol{t}]_\times \mathbf{R}_\mathbf{q}$ according to eq. (B.10). Equal pitch (resp. equal translation length for pure translations) is enforced afterwards providing the same scale for all following motions (see Sec. 3.5.3), given that the rigidly coupled motions are non-planar.

Absolute pose problem Absolute pose estimation from 2d/3d correspondences is approached analogously by:

$$\min_{\substack{q,\boldsymbol{q}_{1,\ldots,n} \\ p,\boldsymbol{p}_{1,\ldots,n}}} \sum_{i=1}^{n} \sum_{(k,j)\in\mathcal{V}_i} d_i(\boldsymbol{u}_{k,j}^{(i)}, \mathbf{R}_{(\boldsymbol{q}_i,q)}^\top (\boldsymbol{X}_j^{(i)} - \boldsymbol{t}_{(\boldsymbol{q}_i,q),(\boldsymbol{p}_i,p)}))^2 \tag{5.10}$$

subject to $\boldsymbol{q}_i^\top \boldsymbol{q}_i + q^2 = 1$ and $\boldsymbol{q}_i^\top \boldsymbol{p}_i + qp = 0$ for all $i = 1, \ldots, n$. In order to alleviate the impact of the initial scaling, the relative scale between $\mathcal{C}_1$ and $\mathcal{C}_i$ should be reevaluated from all previous motions as described in Sec. 3.5.3 during sequential SfM.

Equation (5.9) and (5.9) are solved via nonlinear optimization methods given initial approximate solutions here.

Remarks In [EK13] we have shown that imposing these hard constraints is capable of increasing the accuracy of both pose estimation and eye-to-eye calibration from the estimated poses, especially for the case that only few rigidly coupled motions are given. This will also be demonstrated by the experiments in Sec. 5.7.1.

Note that the same strategy can be used to enforce partial rigid motion constraints in bundle adjustment as described in B.5. However, the number of parameters is approximately n times the number of parameters for decoupled bundle adjustment for n rigidly coupled cameras.

5.5 Rigidly Coupled Bundle Adjustment

Given that initial estimates for the eye-to-eye transformations are known, full rigid motion constraints can be enforced instead of partial constraints. In this section we describe a numerical solution to eq. (5.3) via rigidly coupled bundle adjustment of all cameras, extending the classical bundle adjustment approach described in B.5. This approach extends the method described by Lébraly et al. [Léb+10; Léb+11] by taking more than two coupled cameras into account and adding support for fixed 3d points in the local camera coordinate frames.

We use a similar notation as in Sec. 5.1 here. A brief overview of relevant symbols is listed in Table 5.1. Indices $i = 1, \dots, n$ identify different rigidly coupled cameras. For each camera we are given m camera poses $\mathbf{T}_k^{(i)}$, $k = 1, \dots, m$ parametrized by $\boldsymbol{\mu}_k^{(i)} \in \mathbb{R}^\mu$ resp. $\bar{\boldsymbol{\mu}}_k^{(i)} \in \mathbb{R}^\mu$ for the inverse pose transformations ${\mathbf{T}_k^{(i)}}^{-1}$. Additional parameter vectors $\Delta\boldsymbol{\mu}_i \in \mathbb{R}^{\mu'}$ are used to describe the eye-to-eye similarity transformations $\Delta\mathbf{T}_i$ for $i = 1, \dots, n$ where $\Delta\mathbf{T}_1 = \mathbf{I}$ is fixed since the first camera defines w.l.o.g. the reference coordinate frame of the rig.

The joint error function of all cameras is given by the sum of all individual error functions. The coupling of the individual error terms is realized by expressing each motion of the i-th camera by the respective motion of the

first camera and the i-th eye-to-eye transformation:

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{F}(\boldsymbol{\theta})\|^2 = \sum_{i=1}^{n} \sum_{(k,j)\in\mathcal{V}_i} d_i(\boldsymbol{u}_{k,j}^{(i)}, \underbrace{\mathcal{M}(\mathrm{pred}(\bar{\boldsymbol{\mu}}_k, \Delta\boldsymbol{\mu}_i), \boldsymbol{\chi}_j^{(i)})}_{\hat{\boldsymbol{X}}'(\bar{\boldsymbol{\mu}}_k, \Delta\boldsymbol{\mu}_i, \boldsymbol{\chi}_j^{(i)})})^2 \tag{5.11}$$

where

$$\boldsymbol{\theta} = (\underbrace{\Delta\boldsymbol{\mu}_2, \ldots, \Delta\boldsymbol{\mu}_n}_{\boldsymbol{\theta}_\Delta}, \underbrace{\bar{\boldsymbol{\mu}}_1, \ldots, \bar{\boldsymbol{\mu}}_m}_{\boldsymbol{\theta}_\mathrm{M}}, \underbrace{\boldsymbol{\chi}_1^{(1)}, \ldots, \boldsymbol{\chi}_{N_1}^{(1)}}_{\boldsymbol{\theta}_\mathrm{X}^{(1)}}, \ldots, \underbrace{\boldsymbol{\chi}_1^{(n)}, \ldots, \boldsymbol{\chi}_{N_n}^{(n)}}_{\boldsymbol{\theta}_\mathrm{X}^{(n)}})$$

combines camera poses, eye-to-eye transformations, and 3d point parameters. The size of the parameter space is given by $P = m\mu + (n-1)\mu' + N\chi$ where $N = (\sum_{i=1}^{n} N_i)$. Inverse poses of the first camera are parametrized by vectors $\bar{\boldsymbol{\mu}}_k \in \mathbb{R}^\mu$, denoted as *reference poses* here.

The *local pose prediction function* $\mathrm{pred} : \mathbb{R}^\mu \times \mathbb{R}^\mu \to \mathbb{R}^\mu$ estimates the inverse local pose of a camera from the inverse reference pose parameters and the eye-to-eye transformation parameters:

$$\mathrm{pred}(\bar{\boldsymbol{\mu}}_k, \Delta\boldsymbol{\mu}_i) = \mathrm{rel}(\Delta\boldsymbol{\mu}_i, \mathrm{conj}(\bar{\boldsymbol{\mu}}_k, \Delta\boldsymbol{\mu}_i)) \tag{5.12}$$

motivated from $\mathbf{T}_k^{(1)}\Delta\mathbf{T}_i = \Delta\mathbf{T}_i\mathbf{T}_k^{(i)}$, i. e., $(\mathbf{T}_k^{(i)})^{-1} = \Delta\mathbf{T}_i^{-1}(\mathbf{T}_k^{(1)})^{-1}\Delta\mathbf{T}_i$.

The *3d point prediction function* for non-reference cameras

$$\hat{\boldsymbol{X}}' : \mathbb{R}^\varrho \times \mathbb{R}^\varrho \times \mathbb{R}^\chi \to \mathbb{R}^3,\ (\bar{\boldsymbol{\mu}}, \Delta\boldsymbol{\mu}, \boldsymbol{\chi}) \mapsto \mathcal{M}(\mathrm{pred}(\bar{\boldsymbol{\mu}}, \Delta\boldsymbol{\mu}), \boldsymbol{\chi}) \tag{5.13}$$

in eq. (5.11) is defined analogously to the prediction function eq. (B.19) for the reference camera $\hat{\boldsymbol{X}}(\bar{\boldsymbol{\mu}}, \boldsymbol{\chi}) = \mathcal{M}(\bar{\boldsymbol{\mu}}, \boldsymbol{\chi})$ from default bundle adjustment.

As in the monocular case, gauge freedoms are avoided by fixing the initial pose $\mathbf{T}_1^{(1)}$ and the baseline length $\|\boldsymbol{t}_2^{(1)} - \boldsymbol{t}_1^{(1)}\|$ of the reference camera, reducing the number of parameters to estimate by $\mu + 1$.

If the locations of some 3d points are fixed w.r.t. the camera reference frames $\mathcal{C}_i$ (e. g., given by fiducial markers), the corresponding coordinates

are removed from the parameter vector, providing residuals in eq. (5.11) that depend only on reference pose parameters and eye-to-eye transformation parameters:

$$d_i(\boldsymbol{u}_{k,j}^{(i)}, \hat{\boldsymbol{X}}'_{\boldsymbol{\chi}_j^{(i)}}(\bar{\boldsymbol{\mu}}_k, \Delta\boldsymbol{\mu}_i))^2 \quad \text{with } \hat{\boldsymbol{X}}'_{\boldsymbol{\chi}}(\bar{\boldsymbol{\mu}}, \Delta\boldsymbol{\mu}) = \hat{\boldsymbol{X}}'(\bar{\boldsymbol{\mu}}, \Delta\boldsymbol{\mu}, \boldsymbol{\chi})$$

In the following we will describe bundle adjustment of the rig for planar cameras using the normalized reprojection error as introduced in Sec. 5.3.1 for sake of clarity. In this case, the error function is defined by:

$$G : \mathbb{R}^P \to \mathbb{R}, \boldsymbol{\theta} \mapsto \|\boldsymbol{F}(\boldsymbol{\theta})\|^2 \quad \text{with } \boldsymbol{F} : \mathbb{R}^P \to \mathbb{R}^{2M}$$

where $M = \sum_{i=1}^{n} M_i$, $M_i = |\mathcal{V}^{(i)}|$, is the number of 2d point observations in all images for all cameras, and $\boldsymbol{F}$ is defined by:

$$\boldsymbol{F}(\boldsymbol{\theta}) = \left(\boldsymbol{f}_1^{(1)}(\boldsymbol{\theta})^\mathsf{T}, \ldots, \boldsymbol{f}_{M_1}^{(1)}(\boldsymbol{\theta})^\mathsf{T}, \ldots, \boldsymbol{f}_1^{(n)}(\boldsymbol{\theta})^\mathsf{T}, \ldots, \boldsymbol{f}_{M_n}^{(n)}(\boldsymbol{\theta})^\mathsf{T}\right)^\mathsf{T}$$

$$\text{with } \boldsymbol{f}_\ell^{(i)}(\boldsymbol{\theta}) = \begin{cases} \left(\mathcal{P}(\hat{\boldsymbol{X}}(\bar{\boldsymbol{\mu}}_k, \boldsymbol{\chi}_j^{(1)})) - \boldsymbol{x}_{k,j}^{(1)}\right)_{[1\ldots2]} & \text{for } i = 1 \\ \left(\mathcal{P}(\hat{\boldsymbol{X}}'(\bar{\boldsymbol{\mu}}_k, \Delta\boldsymbol{\mu}_i, \boldsymbol{\chi}_j^{(i)})) - \boldsymbol{x}_{k,j}^{(i)}\right)_{[1\ldots2]} & \text{else} \end{cases} \tag{5.14}$$

where $(k, j) = (k_\ell^{(i)}, j_\ell^{(i)})$ for $i = 1, \ldots, n$, $\ell = 1, \ldots, M_i$.

The *2d point prediction functions* for reference and non-reference cameras are abbreviated as $\hat{\boldsymbol{x}} = \mathcal{P} \circ \hat{\boldsymbol{X}}$ and $\hat{\boldsymbol{x}}' = \mathcal{P} \circ \hat{\boldsymbol{X}}'$ in the following.

The normal equations for iterative solution of eq. (5.11) via the Levenberg-Marquardt algorithm given an initial solution $\boldsymbol{\theta}_0$ are defined by:

$$\mathbf{J}_{\boldsymbol{F}}{}^\mathsf{T}\mathbf{J}_{\boldsymbol{F}}(\boldsymbol{\theta} - \boldsymbol{\theta}_0) = -\mathbf{J}_{\boldsymbol{F}}{}^\mathsf{T}\boldsymbol{F}(\boldsymbol{\theta}_0) \tag{5.15}$$

where $\mathbf{J}_{\boldsymbol{F}} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}_0)$ is the Jacobian matrix of $\boldsymbol{F}$ evaluated at $\boldsymbol{\theta}_0$ and $\mathbf{H}_G = \mathbf{J}_{\boldsymbol{F}}{}^\mathsf{T}\mathbf{J}_{\boldsymbol{F}}$ is an approximation to the Hessian matrix of the error function G.

The Jacobian matrix of $\boldsymbol{F}$ is given by:

$$\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{\theta}} = \begin{pmatrix} \frac{\partial \boldsymbol{f}_1^{(1)}}{\partial \boldsymbol{\theta}_\Delta} & \frac{\partial \boldsymbol{f}_1^{(1)}}{\partial \boldsymbol{\theta}_\mathrm{M}} & \frac{\partial \boldsymbol{f}_1^{(1)}}{\partial \boldsymbol{\theta}_\mathrm{X}^{(1)}} & \cdots & \frac{\partial \boldsymbol{f}_1^{(1)}}{\partial \boldsymbol{\theta}_\mathrm{X}^{(n)}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \boldsymbol{f}_{M_1}^{(1)}}{\partial \boldsymbol{\theta}_\Delta} & \frac{\partial \boldsymbol{f}_{M_1}^{(1)}}{\partial \boldsymbol{\theta}_\mathrm{M}} & \frac{\partial \boldsymbol{f}_{M_1}^{(1)}}{\partial \boldsymbol{\theta}_\mathrm{X}^{(1)}} & \cdots & \frac{\partial \boldsymbol{f}_{M_1}^{(1)}}{\partial \boldsymbol{\theta}_\mathrm{X}^{(n)}} \\ & & \vdots & & \\ \frac{\partial \boldsymbol{f}_1^{(n)}}{\partial \boldsymbol{\theta}_\Delta} & \frac{\partial \boldsymbol{f}_1^{(n)}}{\partial \boldsymbol{\theta}_\mathrm{M}} & \frac{\partial \boldsymbol{f}_1^{(n)}}{\partial \boldsymbol{\theta}_\mathrm{X}^{(1)}} & \cdots & \frac{\partial \boldsymbol{f}_1^{(n)}}{\partial \boldsymbol{\theta}_\mathrm{X}^{(n)}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \boldsymbol{f}_{M_n}^{(n)}}{\partial \boldsymbol{\theta}_\Delta} & \frac{\partial \boldsymbol{f}_{M_n}^{(n)}}{\partial \boldsymbol{\theta}_\mathrm{M}} & \frac{\partial \boldsymbol{f}_{M_n}^{(n)}}{\partial \boldsymbol{\theta}_\mathrm{X}^{(1)}} & \cdots & \frac{\partial \boldsymbol{f}_{M_n}^{(n)}}{\partial \boldsymbol{\theta}_\mathrm{X}^{(n)}} \end{pmatrix} \tag{5.16}$$

where the partial derivatives are all zero except for:[7]

$$\begin{aligned}
\frac{\partial \boldsymbol{f}_\ell^{(1)}}{\partial \bar{\boldsymbol{\mu}}_k}(\boldsymbol{\theta}) &= \frac{\partial \hat{\boldsymbol{x}}}{\partial \bar{\boldsymbol{\mu}}}(\bar{\boldsymbol{\mu}}_k, \boldsymbol{\chi}_j^{(1)}) \quad \text{for } \ell = 1, \ldots, M_1 \\
\frac{\partial \boldsymbol{f}_\ell^{(1)}}{\partial \boldsymbol{\chi}_j^{(1)}}(\boldsymbol{\theta}) &= \frac{\partial \hat{\boldsymbol{x}}}{\partial \boldsymbol{\chi}}(\bar{\boldsymbol{\mu}}_k, \boldsymbol{\chi}_j^{(1)}) \quad \text{for } \ell = 1, \ldots, M_1 \\
\frac{\partial \boldsymbol{f}_\ell^{(i)}}{\partial \bar{\boldsymbol{\mu}}_k}(\boldsymbol{\theta}) &= \frac{\partial \hat{\boldsymbol{x}}'}{\partial \bar{\boldsymbol{\mu}}}(\bar{\boldsymbol{\mu}}_k, \Delta\boldsymbol{\mu}_i, \boldsymbol{\chi}_j^{(i)}) \quad \text{for } i = 2, \ldots, n,\ \ell = 1, \ldots, M_i \\
\frac{\partial \boldsymbol{f}_\ell^{(i)}}{\partial \boldsymbol{\chi}_j^{(i)}}(\boldsymbol{\theta}) &= \frac{\partial \hat{\boldsymbol{x}}'}{\partial \boldsymbol{\chi}}(\bar{\boldsymbol{\mu}}_k, \Delta\boldsymbol{\mu}_i, \boldsymbol{\chi}_j^{(i)}) \quad \text{for } i = 2, \ldots, n,\ \ell = 1, \ldots, M_i \\
\frac{\partial \boldsymbol{f}_\ell^{(i)}}{\partial \Delta\boldsymbol{\mu}_i}(\boldsymbol{\theta}) &= \frac{\partial \hat{\boldsymbol{x}}'}{\partial \Delta\boldsymbol{\mu}}(\bar{\boldsymbol{\mu}}_k, \Delta\boldsymbol{\mu}_i, \boldsymbol{\chi}_j^{(i)}) \quad \text{for } i = 2, \ldots, n,\ \ell = 1, \ldots, M_i
\end{aligned}$$

The sparse block structure of the approximate Hessian matrix is similar to the "arrow head" structure of the individual Hessian matrices for bundle adjustment with a single camera as illustrated in Fig. 5.6 and 5.7.

[7] For brevity, we write (k, j) instead of $(k_\ell^{(1)}, j_\ell^{(1)})$ resp. $(k_\ell^{(i)}, j_\ell^{(i)})$ here.

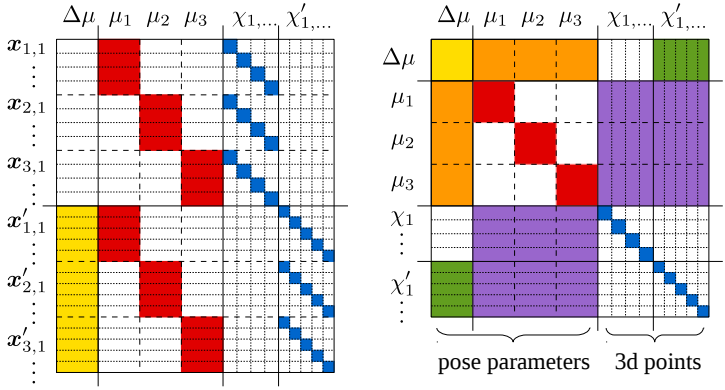

Figure 5.6. Form of the Jacobian matrix $\mathbf{J}_F$ (left) and approximate Hessian matrix $\mathbf{H}_G = \mathbf{J}_F{}^\mathsf{T}\mathbf{J}_F$ (right) for a rigidly coupled bundle adjustment example consisting of $n = 2$ rigidly coupled cameras, $m = 3$ image pairs and $N = 4$ resp. $N' = 5$ 3d points for each camera. The eye-to-eye transformation is described by $\Delta\boldsymbol{\mu}$, poses of the first camera by $\boldsymbol{\mu}_{1,\dots,3}$, and 3d points for the cameras by $\chi_{1,\dots,4}$, $\chi'_{1,\dots,5}$.

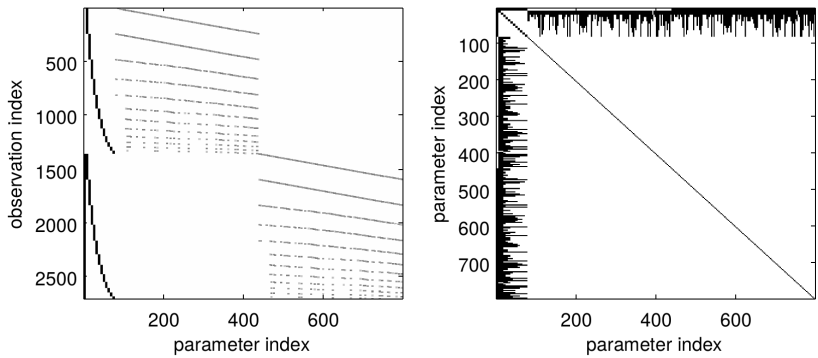

Figure 5.7. Jacobian matrix $\mathbf{J}_F$ (left) and approximate Hessian matrix $\mathbf{H}_G = \mathbf{J}_F{}^\mathsf{T}\mathbf{J}_F$ (right) for a moderately sized rigidly coupled bundle adjustment problem consisting of $m = 12$ images, $N_1 = N_2 = 120$ 3d points each, $P = 798$ parameters, and $M = 1350$ 2d points. Black areas correspond to non-zero entries. The fill degree of $\mathbf{J}_F$ and $\mathbf{H}_G$ is 1.5% and 8.86% respectively.

Bundle adjustment in common coordinate frame The complexity of the error function can be reduced by transforming the poses and 3d points for all cameras from $\mathcal{C}_i$ into the reference coordinate frame $\mathcal{C}_1$ according to eq. (5.4) as suggested in [Léb+11]:

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{F}(\boldsymbol{\theta})\|^2 = \sum_{i=1}^{n} \sum_{(k,j)\in\mathcal{V}_i} d_i(\boldsymbol{u}_{k,j}^{(i)}, \underbrace{\mathcal{M}(\mathrm{rel}(\Delta\boldsymbol{\mu}_i, \bar{\boldsymbol{\mu}}_k), \boldsymbol{\chi}_j^{*(i)})}_{\hat{\boldsymbol{X}}^*(\bar{\boldsymbol{\mu}}_k, \Delta\boldsymbol{\mu}_i, \boldsymbol{\chi}_j^{*(i)})})^2 \tag{5.17}$$

In this context, 3d point parameters $\boldsymbol{\chi}_j^{(i)}$ are replaced by parameters $\boldsymbol{\chi}_j^{*(i)}$ of $\boldsymbol{X}_j^{*(i)} = \Delta\mathbf{T}_i \boldsymbol{X}_j^{(i)}$, i. e., the j-th 3d point of the i-th camera within the reference coordinate system $\mathcal{C}_1$. Thus, the number of pose concatenations in the 3d point prediction function is effectively reduced:

$$\hat{\boldsymbol{X}}^* : \mathbb{R}^{\varrho} \times \mathbb{R}^{\varrho} \times \mathbb{R}^{\chi} \to \mathbb{R}^3,\ (\bar{\boldsymbol{\mu}}, \Delta\boldsymbol{\mu}, \boldsymbol{\chi}) \mapsto \mathcal{M}(\mathrm{rel}(\Delta\boldsymbol{\mu}, \bar{\boldsymbol{\mu}}), \boldsymbol{\chi}) \tag{5.18}$$

The general structure of the Jacobian matrix $\mathbf{J}_{\boldsymbol{F}}$ and approximate Hessian matrix $\mathbf{H}_G$ is not altered by this modification.

Note that no relative scale must be taken into account since different reconstruction scales are encoded within the 3d points here.

Remarks The same strategy can be used to solve the alleviated Eye-and-World SfM problem (5.6). If there are no fixed 3d points, the world-to-world transformation parameters $\Delta\mathbf{W}_i$ can be dropped completely from rigidly coupled bundle adjustment since they are encoded entirely within the estimated 3d points.

Note however that equations with respect to fixed 3d points cannot be transformed into a common reference coordinate frame since fixed 3d point coordinates refer to the local coordinate frames $\mathcal{C}_i$ resp. $\mathcal{W}_i$. Considering that fixed 3d points are most commonly involved in the eye-and-world calibration problem (i. e., camera poses are at least partially estimated from calibration objects with known geometry), the world-to-world transformation parameters are included in bundle adjustment.

Table 5.1. Overview of symbols used in rigidly coupled bundle adjustment.

i	index of camera within the rig
n	number of cameras within the rig
k	index of camera image
m	number of images for each camera
j	index of 3d point within a scene
N_i	number of 3d points for the i-th camera
N	number of 3d points for all cameras ($\sum_{i=1}^{n} N_i$)
M_i	number of 2d observations for the i-th camera
M	number of 2d observations for all cameras ($\sum_{i=1}^{n} M_i$)
μ	number of parameters per pose (6 to 8)
μ'	number of parameters per eye-to-eye transform (6 to 8)
χ	number of parameters per 3d point (3)
P	number of all parameters (max. $m\mu + (n-1)\mu' + N\chi$)
$\boldsymbol{\theta}$	vector of all parameters
$\mathbf{T}_k^{(i)}, \boldsymbol{\mu}_k^{(i)}$	k-th local pose for i-th camera and its parameters
$\Delta\mathbf{T}_i, \Delta\boldsymbol{\mu}_i$	i-th eye-to-eye transformation and its parameters
$\boldsymbol{X}_j^{(i)}, \boldsymbol{\chi}_j^{(i)}$	j-th 3d point for i-th camera and its parameters
$\boldsymbol{u}_{k,j}^{(i)}$	projection of j-th 3d point in k-th image of i-th camera
$\boldsymbol{x}_{k,j}^{(i)}$	normalized image coordinates of 2d point $\boldsymbol{u}_{k,j}^{(i)}$
$\mathcal{V}_i$	visibility set for i-th camera ($(k,j) \in \mathcal{V}_i$ iff j-th 3d point is visible in k-th image of i-th camera)

Implementation More general methods than for classical sparse bundle adjustment as described in B.5 must be used to implement the proposed rigidly coupled bundle adjustment since $\mathbf{H}_G$ has not the distinct block structure needed to compute the Schur complement as required by `sba` [LA09]. This is due to the fact that all reprojection errors for the i-th cameras depend on the eye-to-eye transformation parameters $\Delta\boldsymbol{\mu}_i$. We implemented this approach using a sparse C/C++ implementation of the Levenberg-Marquardt algorithm, `sparseLM` [Lou10], instead. Eye-to-eye bundle adjustment is briefly referred to as `E2EBA` in the following.

5.6 Complete Multi-Camera Calibration

In the previous chapters we have reduced the problem of extrinsic multi-camera calibration to the case of pair-wise calibration so far. Although global bundle adjustment can handle multiple cameras at the same time, it is very demanding due to the large number of parameters and observations involved. Moreover, it might not be possible to capture corresponding images for all cameras at the same time due to practical limitations.

In this section we will discuss how to derive a full calibration from pair-wise eye-to-eye transformations between cameras. Given several relative poses, a posteriori enforcement of global consistency provides an effective method of achieving improved extrinsic calibration results (see [Dai+10]).

In the literature we find predominantly approaches to calibrating static camera networks consisting of cameras with pair-wise overlapping views [BA00]. First, relative poses are computed for subsets of cameras that share a common field of view using feature-based approaches. Afterwards, the relative pose between any two cameras in the network can be computed via pose propagation (also denoted as "transfer") as long as there is a direct path between both cameras where relative poses are known. The problem of achieving a globally consistent network calibration is in general approached in two different stages: First, find an optimal minimal subset of relative poses and infer globally consistent relative and absolute poses from these. Second, refine absolute poses w.r.t. the measured relative poses. Baker & Aloimonos simplify the latter optimization problem by compensating rotation in all cameras first and afterwards finding absolute camera positions such that the reprojection errors for all overlapping pose pairs is minimized [BA00]. However they do not state explicitly how to select the relative poses to derive initial absolute poses from. Martinec & Pajdla [MP07] describe a method to evaluate image pairs and relative poses in the context of multi-camera 3d reconstruction based on robust feature matching using heuristic inliers thresholds for rejection and find a minimal spanning tree in the camera adjacency graph using this measure. Vergés-Llahí, Moldovan & Wada [VMW08] propose a similar solution but use an uncertainty measure consisting of a residual and a constraint

violation term for weighting edges in the adjacency graph. Bajramovic & Denzler [BD08] also use a probabilistic model instead of an inlier threshold. They describe three geometric uncertainty measures for relative pose estimates based on the posterior probability density function of relative poses with respect to a set of SIFT feature correspondences. This approach is strongly related to the work of Engels & Nistér [EN05] who propose a sampling-based approach to estimate the global uncertainty of a relative pose estimate. However, all these methods are based on 2d point correspondences between the images from which the relative pose was estimated. Several authors advise to remove camera pairs with insignificant common field of view from the input in order to achieve good results, using for instance view similarity measures based on correspondence probabilities of point features [BBD09a; BBD09b]. A more generic approach for joint calibration of multiple rigidly coupled sensors was proposed by Le & Ng [LN09]. However, since sensors are considered to produce 3d point measurements here (e. g., depth cameras or stereo cameras), their method cannot be applied to monocular cameras within the multi-camera system.

In the following, we will provide a purely geometric approach to complete eye-to-eye calibration that is based on minimizing an error function of absolute poses w.r.t. measured relative poses given measurement uncertainties. This approach is similar to conjugate rotation averaging [Dai+10] for the case of rigidly coupled rotations. First we will recapitulate the problem and propose a graph-based initial solution and absolute scale retrieval based on Bajramovic & Denzler's approach [BD08].

Problem statement Given is a multi-camera setup consisting of n cameras $\mathcal{C}_1, \ldots, \mathcal{C}_n$. Using eye-to-eye calibration techniques we obtain relative pose estimates $\Delta\mathbf{T}_{i,j}$ from the local coordinate frame of camera $\mathcal{C}_j$ to camera $\mathcal{C}_i$ for camera pairs $(i,j) \in \mathcal{J}$ with $1 \leqslant i,j \leqslant n$. These quantities are considered as measurements here. We assume that all cameras are connected to each other by at least one eye-to-eye transformation.

As eye-to-eye transformations have been computed for each camera pair individually, they are not guaranteed to be globally consistent, i. e., the consistency constraint $\Delta\mathbf{T}_{i,j} = \Delta\mathbf{T}_{i,k}\Delta\mathbf{T}_{k,j}$ is not exactly satisfied for trian-

gles $(i,j),(i,k),(k,j) \in \mathcal{J}$. To achieve global consensus, we seek absolute camera poses $\mathbf{T}_i$, $1 \leqslant i \leqslant n$, w.r.t. a fixed reference coordinate frame so that the error between predicted relative poses $\mathbf{T}_i^{-1}\mathbf{T}_j$ and observed relative poses $\Delta\mathbf{T}_{i,j}$ is minimized with respect to a metric d on SE(3):

$$\min_{\mathbf{T}_1,\ldots,\mathbf{T}_n} \sum_{(i,j)\in\mathcal{J}} d(\mathbf{T}_i^{-1}\mathbf{T}_j, \Delta\mathbf{T}_{i,j})^2 \tag{5.19}$$

This problem can be considered as converse to the *Absolute Pose Alignment problem* introduced in Sec. 2.6.1: Given relative pose measurements describing eye-to-eye transformations, find the absolute poses that are aligned with each other by them.

Retrieving relative scales As stated in Sec. 3.5.3, eye-to-eye transformations are only estimated up to scale when local pose transformations with unknown absolute scale are provided as input. Under the assumption that for each camera $\mathcal{C}_i$ we have a triangle $(i,j),(i,k),(k,j) \in \mathcal{J}$, globally consistent scales $\lambda_{i,j}$ can be inferred for each relative pose $\Delta\mathbf{T}_{i,j}$ when at least one distance between cameras is known or fixed arbitrarily. Assume that $\|\Delta\boldsymbol{t}_{i,j}\|$ has already been determined. The camera positions in the triangle (i,j,k) are related by:

$$\lambda_{i,k}\Delta\boldsymbol{t}_{i,k} = \lambda_{j,k}\Delta\mathbf{R}_{i,j}\Delta\boldsymbol{t}_{j,k} + \Delta\boldsymbol{t}_{i,j} \tag{5.20}$$

which actually poses a *Triangulation problem* and can be solved as described in B.4. Hence, absolute scales can be propagated through triangles in $\mathcal{J}$ by processing triangles with a common edge subsequently [BD08].

Graph-based solution A globally consistent solution can be derived from a minimal set of "good" relative pose estimates via pose propagation. For this purpose, the relationship between cameras is modeled as a directed weighted graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$ where nodes $V_i \in \mathcal{V} = \{V_1,\ldots,V_n\}$ represent cameras $\mathcal{C}_i$ and edges $E_{i,j} \in \mathcal{E} = \{(V_i,V_j) \mid (i,j) \in \mathcal{J}\}$ are related to the measured eye-to-eye transformation $\Delta\mathbf{T}_{i,j}$ from the j-th to the i-th camera. The weights $w_{i,j} \geqslant 0$ associated with each edge $E_{i,j}$ should corre-

spond to the quality of the eye-to-eye transformation $\Delta\mathbf{T}_{i,j}$ (e. g., derived from the parameter covariance passed down from eye-to-eye calibration). After computation of a minimum spanning tree $\mathcal{G}' = (\mathcal{V}, \mathcal{E}'), \mathcal{E}' \subset \mathcal{E}$ of $\mathcal{G}$ with respect to the weights $w_{i,j}$, globally consistent relative poses $\Delta\mathbf{T}^*_{i,j}$ are derived for any camera pair (i, j) via pose propagation along the unique path from V_j to V_i in $\mathcal{G}'$, described recursively by:[8]

$$\Delta\mathbf{T}^*_{i,j} = \begin{cases} \Delta\mathbf{T}_{i,j} & E_{i,j} \in \mathcal{E}' \\ \Delta\mathbf{T}^{-1}_{i,j} & E_{j,i} \in \mathcal{E}' \\ \Delta\mathbf{T}^*_{i,k}\Delta\mathbf{T}^*_{k,j} & V_k \in \mathcal{V}, k \neq i \wedge k \neq j \end{cases} \tag{5.21}$$

Note that traversing an edge $E_{i,j}$ in the spanning tree in opposite direction corresponds to the transformation $\Delta\mathbf{T}^{-1}_{i,j}$.

The graph-based approach provides globally consistent relative poses from which absolute poses according to eq. (5.19) can be derived. The overall pose ambiguity inherent to this problem is fixed by defining an arbitrary camera $\mathcal{C}_{i_{\text{ref}}}$ as reference. Hence, the i-th absolute pose is described by the transformation from the i-th camera to the reference camera $\mathbf{T}_i = \Delta\mathbf{T}^*_{i_{\text{ref}},i}$.

Nonlinear optimization Although this method yields a feasible solution, it rejects the amount of redundacy in the input data that should be exploited in order to improve the quality of the initial solution [LN09]. Therefore, further nonlinear optimization of an error function based on eq. (5.19) is recommended, e. g., via the Levenberg-Marquardt algorithm. The error function depends on the actual parametrization $\boldsymbol{\mu}$ of absolute pose parameters and relative pose measurements. According to the discussion of non-linear methods for eye-to-eye calibration in Chapter 3, we use unit quaternions for rotation parametrization and minimize the rotation term of eq. (5.19) with respect to quaternion distance:

$$\min_{\substack{\mathbf{q}_1,\ldots,\mathbf{q}_n \\ \boldsymbol{t}_1,\ldots,\boldsymbol{t}_n}} \sum_{(i,j)\in\mathcal{J}} \|\bar{\mathbf{q}}_i \cdot \mathbf{q}_j - \Delta\mathbf{q}_{i,j}\|^2 + \|(\mathbf{R}^{\intercal}_{\mathbf{q}_i}(\boldsymbol{t}_j - \boldsymbol{t}_i) - \Delta\boldsymbol{t}_{i,j}\|^2 \tag{5.22}$$

[8] The graph $\mathcal{G}'$ is in fact a spanning tree of $\mathcal{G}$ ignoring edge directions. However, $\mathcal{G}'$ is considered as orientated, since each edge corresponds either to $\Delta\mathbf{T}_{i,j}$ or to $\Delta\mathbf{T}_{j,i}$.

subject to unit length of $\mathbf{q}_1, \ldots, \mathbf{q}_n$ and fixed parameters $\mathbf{q}_{i_{\text{ref}}}$, $\boldsymbol{t}_{i_{\text{ref}}}$. Since eq. (5.22) is based on an explicit mapping from parameters to observation, measurement covariances for the eye-to-eye transformations can be taken into account, replacing the Euclidean distance by the Mahalanobis distance with respect to $\boldsymbol{\Sigma}_{\Delta\boldsymbol{\mu}_{i,j}}$.

5.7 Evaluation

5.7.1 Method Comparison

First, we evaluate the proposed methods for feature-based eye-to-eye calibration with synthetic data created as follows:

- For a rig consisting of n cameras, $n-1$ random eye-to-eye configurations $\Delta\mathbf{T}_{1,2}, \ldots, \Delta\mathbf{T}_{1,n}$ consisting of rotations $\Delta\mathbf{R}_{1,i}$ and translations $\Delta\boldsymbol{t}_{1,i}$ are created. The rotation angles are set to a fixed value $\Delta\alpha$ and the absolute translation lengths are set to a fixed value Δd.
- Random poses $\mathbf{T}_k^{(1)}$, $k = 2, \ldots, m+1$ for the first camera are created as in Sec. 3.8.1. The initial pose is fixed as $\mathbf{T}_1^{(1)} = [\mathbf{I} \mid \mathbf{0}]$ for all cases. Absolute rotation and translation is bounded by fixed values α and d.
- Compute local poses of the i-th camera $\mathbf{T}_k^{(i)} = \Delta\mathbf{T}_{1,i}^{-1}\mathbf{T}_k^{(1)}\Delta\mathbf{T}_{1,i}$ for each $i = 2, \ldots, n$, $k = 1, \ldots, m+1$.
- Create N random 3d points $\boldsymbol{X}_j^{(i)}$ uniformly distributed within a confined space in front of each camera $i = 1, \ldots, n$ constituting the scene geometry.
- Project each 3d point $\boldsymbol{X}_j^{(i)}$ in front of the i-th camera with pose $\mathbf{T}_k^{(i)}$ into the image plane, $\boldsymbol{u}_{k,j}^{(i)} = \mathcal{K}(\mathbf{R}_k^{(i)\mathsf{T}}(\boldsymbol{X}_j^{(i)} - \boldsymbol{t}_k^{(i)}))$, and add the resulting 2d/3d correspondence $\boldsymbol{u}_{k,j}^{(i)} \leftrightarrow \boldsymbol{X}_j^{(i)}$ to the visibility set $\mathcal{V}_i$ if $\boldsymbol{u}_{k,j}^{(i)}$ lies within the virtual camera image. We use virtual images with size $w \times h$ and virtual pinhole cameras with principal point $\boldsymbol{p} = (\frac{w}{2}, \frac{h}{2})$ and focal

length $f = w$ providing a field of view of $53.1^\circ \times 41.1^\circ$. The image size is chosen as 800×600 pixels for each camera.

▷ Add Gaussian distributed errors $\boldsymbol{u}_\varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma_{\boldsymbol{u}_\varepsilon} \mathbf{I}_2)$ to all 2d points. The resulting 2d points $\hat{\boldsymbol{u}}_{k,j}^{(i)}$ are considered as SfM input observations.

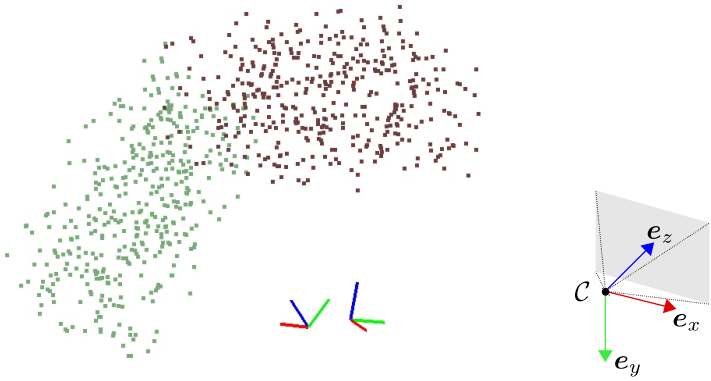

Figure 5.8. Example for synthetic data used for visual eye-to-eye calibration evaluation showing 3d points and initial poses for two rigidly coupled cameras. The camera orientation is indicated by RGB colored coordinate axes (right).

Absolute pose estimation The first simulation addresses pose estimation from 2d/3d correspondences with partial rigid motion constraint enforcement (see Sec. 5.4) and eye-to-eye calibration based on the resulting poses. The locations of the 3d points are supposed to be certain in this case, e. g., defined by a calibration pattern with accurately known geometry.

First, poses $\hat{\mathbf{T}}_k^{(i)}$ are estimated for each camera from 2d/3d correspondences within the respective reference coordinate frame $\mathcal{C}_i$ (see B.3). 2d/3d correspondences are derived from the visibility set $\mathcal{V}_i$. The maximal number of 2d/3d correspondences per image is limited to $M_{\max} = 50$ points. Optionally, the estimated poses are refined jointly for each image via partial rigid motion constraint enforcement as described in Sec. 5.4. Both methods are briefly referred to as `Abs` vs. `cAbs` in the plots.

Eye-to-eye transformations $\Delta\hat{\mathbf{T}}_{1,2}, \ldots, \Delta\hat{\mathbf{T}}_{1,n}$ are computed from the estimated poses $\hat{\mathbf{T}}_k^{(i)}$ via different eye-to-eye calibration methods: *geometric eye-to-eye calibration* (`gE2E`) using the method `dualquat` refined by $\texttt{quat}_{\texttt{NL}}$ as described in Sec. 3.8, *visual eye-to-eye calibration* (`vE2E`) refining the geometric solution as described in Sec. 5.3.1, and *eye-to-eye bundle adjustment* (`E2EBA`) refining the visual solution as described in Sec. 5.5. However, bundle adjustment is reduced to rigidly coupled pose estimation by fixing all 3d point parameters here. The resulting eye-to-eye transformations are measured with absolute scale within the reference coordinate frame $\mathcal{C}_1$.

For each set of parameters, a number of 100 random samples is created. Rotational errors are measured by the angle of the residual rotation w.r.t. ground truth rotations, i. e., $d_\angle(\mathbf{R}_k^{(i)}, \hat{\mathbf{R}}_k^{(i)})$ and $d_\angle(\Delta\mathbf{R}_{1,i}, \Delta\hat{\mathbf{R}}_{1,i})$. Translational errors are measured as the absolute difference w.r.t. ground truth, i. e., $\|\boldsymbol{t}_k^{(i)} - \hat{\boldsymbol{t}}_k^{(i)}\|$ and $\|\Delta\boldsymbol{t}_{1,i} - \Delta\hat{\boldsymbol{t}}_{1,i}\|$.

In all simulations, we use translation lengths $\Delta d = 1\,\text{m}$, $d = 1\,\text{m}$, rotation angles $\Delta\alpha = 90^\circ$, $\alpha = 30^\circ$, $n = 2$ cameras, and $m = 8$ relative poses (i. e., $m + 1 = 9$ images) if not stated otherwise. Each scene consists of $N = 250$ 3d points distributed within a cuboid of size $8 \times 8 \times 4\,\text{m}$ located 4 m in front of the initial camera location (see Fig. 5.8 for an example). The default standard deviation for 2d point measurements is chosen as $\sigma_{\boldsymbol{u}_\varepsilon} = 1\,\text{px}$.

The resulting errors are visualized as box plots in Fig. 5.9.[9] The upper row shows the average pose estimation errors for all tests while the lower row shows the respective eye-to-eye calibration errors. The first box plot refers to decoupled pose estimation followed by geometric eye-to-eye calibration. Enforcing partial rigid motion constraints (second box plot) improves the pose estimation accuracy, also providing more accurate eye-to-eye calibration results. Minimizing the reprojection error in eye-to-eye calibration (third box plot) further improves the eye-to-eye calibration results. While rigidly coupled pose estimation (fourth box plot) improves the camera poses significantly, the estimated eye-to-eye transformation parameters are not optimized further. This indicates that visual eye-to-eye

[9] Each box is centered around the median, marked with a thick line, and extends to the first and third quartile of the errors, i. e., 50% of all error values lie within the box. The whiskers mark the minimal and maximal errors.

calibration already provides the optimal results w.r.t. reprojection errors in this application.

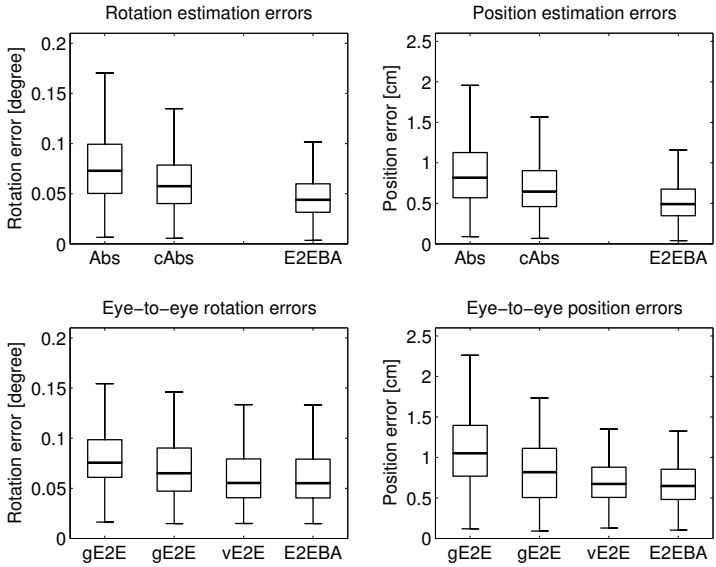

Figure 5.9. Comparison of eye-to-eye calibration methods based on pose estimation from 2d/3d correspondences (from top left to bottom right: box plots for rotation, translation, eye-to-eye rotation, and eye-to-eye translation errors using different methods).

Next, we will evaluate the accuracy of the proposed methods under variation of different parameters affecting absolute pose estimation and/or eye-to-eye calibration.

- ▻ In the first test case (see Fig. 5.10), the magnitude of 2d point error $\sigma_{\boldsymbol{u}_\varepsilon}$ is increased from 0.5 px to 4 px. Pose estimation and eye-to-eye calibration errors increase proportionally.
- ▻ In the second test case, the number of input images resp. poses per camera is increased from $m = 4$ to 16. Figure 5.11 (upper part) shows

in accordance with the results from Sec. 3.8.1 how the accuracy of eye-to-eye calibration is increased significantly with rising number of input poses. Estimation of camera poses via joint bundle adjustment E2EBA is only slightly improved with rising number of images.

▻ In the third test case (see Fig. 5.11, lower part), the number of rigidly coupled cameras is increased from $n = 2$ to 6. Both pose estimation and eye-to-eye calibration results are only slightly improved by considering constraint between coupled cameras via methods cAbs and E2EBA. The effect on eye-to-eye calibration becomes insignificant when more than three cameras are used.

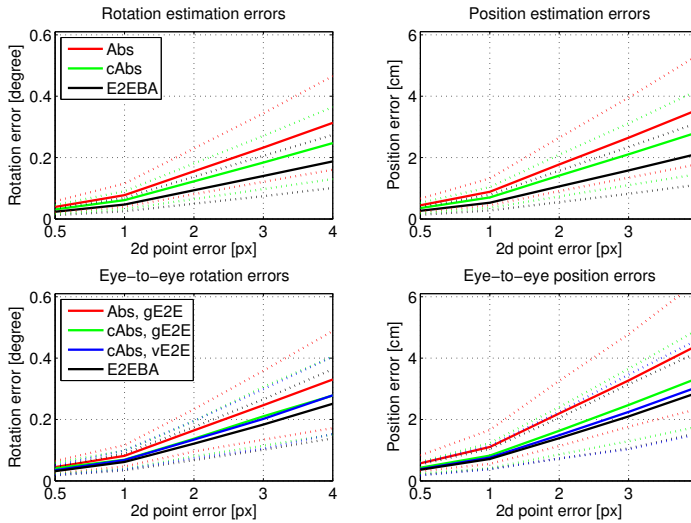

Figure 5.10. Comparison of eye-to-eye calibration methods based on pose estimation from 2d/3d correspondences w.r.t. 2d point error.

All test cases show that visual eye-to-eye calibration improves the results from geometric eye-to-eye calibration significantly. Applying the method E2EBA with fixed 3d points improves the estimated camera poses by enforcing full rigid motion constraints. However, the results from vE2E are not refined any further.

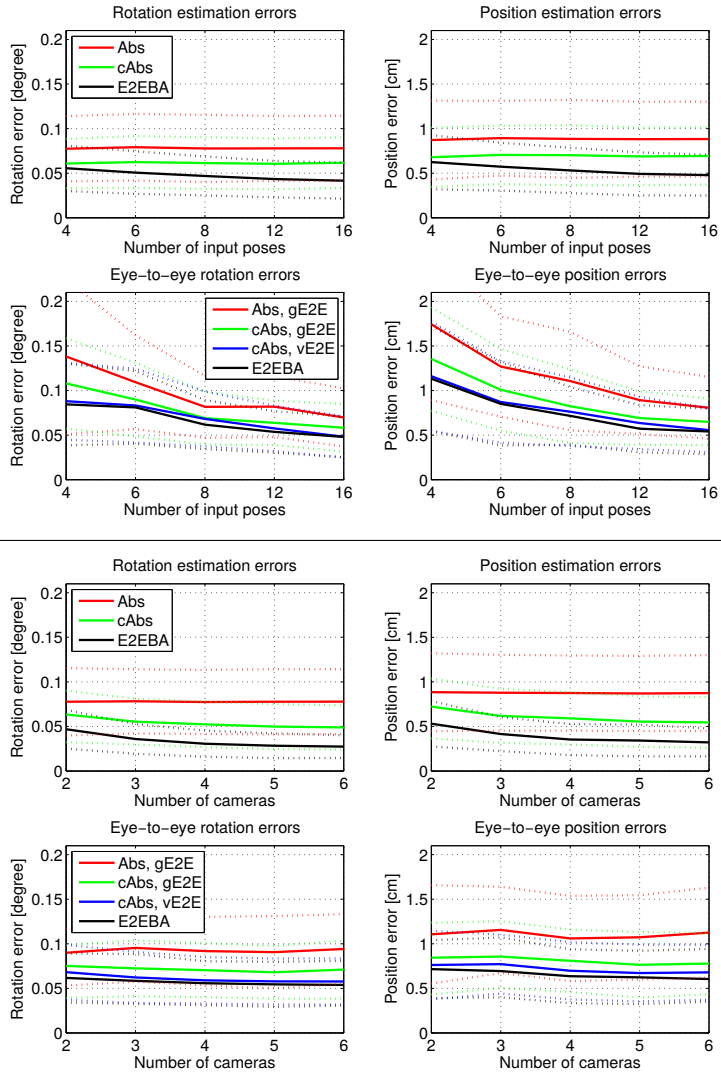

Figure 5.11. Comparison of eye-to-eye calibration methods based on pose estimation from 2d/3d correspondences w.r.t. number of poses per camera (upper part) and number of rigidly coupled cameras (lower part).

Structure from Motion In the second simulation, both 3d points $\hat{\boldsymbol{X}}_j^{(i)}$ and camera poses $\hat{\mathbf{T}}_k^{(i)}$ are estimated from 2d/2d correspondences via incremental Structure from Motion as described in Sec. 5.3. Correspondences between images k, ℓ are given by $\{(\hat{\boldsymbol{u}}_{k,j}^{(i)}, \hat{\boldsymbol{u}}_{\ell,j}^{(i)}) \mid (k,j) \in \mathcal{V}_i \wedge (\ell,j) \in \mathcal{V}_i\}$ for the i-th camera. The maximal number of 2d/2d correspondences per image pair is limited to $M_{\max} = 50$ points. The first pose is fixed as the origin and the baseline (i. e., the estimated distance of the second camera location w.r.t. the origin) is fixed to $\hat{b}_i = \|\hat{\boldsymbol{t}}_2^{(i)} - \hat{\boldsymbol{t}}_1^{(i)}\| = 1$ for each camera, resulting in reconstructions up to scale. The solutions are optionally refined via SfM with partial rigid motion constraints. Decoupled and jointly constrained SfM is briefly referred to as `SfM` vs. `cSfM` in the plots.

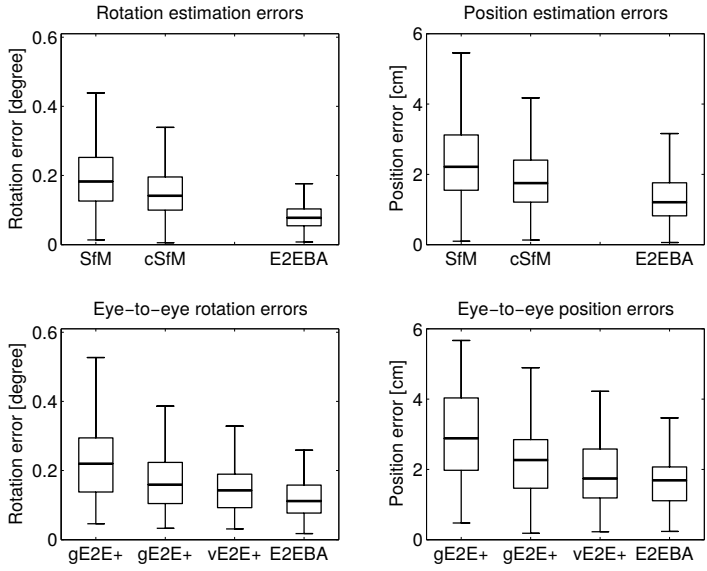

Figure 5.12. Comparison of eye-to-eye calibration methods based on pose estimation from 2d/2d correspondences via Structure from Motion (from top left to bottom right: box plots for rotation, translation, eye-to-eye rotation, and eye-to-eye translation errors using different methods).

In this case, eye-to-eye transformations $\Delta\hat{\mathbf{T}}_{1,2}, \ldots, \Delta\hat{\mathbf{T}}_{1,n}$ are estimated via *extended* eye-to-eye calibration methods. Geometric eye-to-eye calibration (gE2E$^+$) is based on the extended method Tmat$^+$ refined by quat$^+_{\text{NL}}$. Note that proper bundle adjustment of all camera poses, 3d points, and eye-to-eye transformation parameters is performed in this case.

To evaluate absolute pose errors, all poses and eye-to-eye translations are scaled with the ground truth baselines $b_i = \|\boldsymbol{t}_2^{(i)} - \boldsymbol{t}_1^{(i)}\|$ for comparison, i. e., we consider $\|\boldsymbol{t}_k^{(i)} - b_i\hat{\boldsymbol{t}}_k^{(i)}\|$ and $\|\Delta\boldsymbol{t}_{1,i} - b_1\Delta\hat{\boldsymbol{t}}_{1,i}\|$ in the plots. The resulting scale factors $\Delta\lambda_{1,i}$ are limited to the range $[\frac{2}{3}, \frac{3}{2}]$ since the difference of the initial baselines between cameras is bounded by $2\sin(\frac{\alpha}{2})\Delta d = 0.5\,m$.

The resulting errors are visualized as box plots in Fig. 5.12 as in the previous simulation. As expected, the results for SfM-based eye-to-eye calibration are in general less accurate than the results from pose estimation with known 3d points.

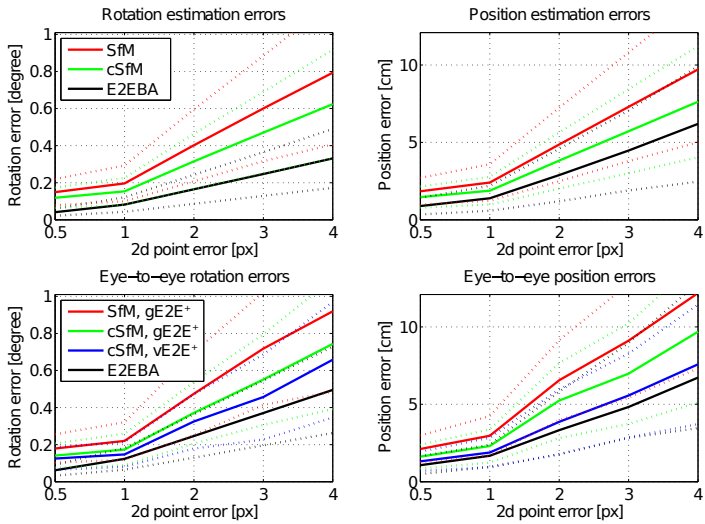

Figure 5.13. Comparison of eye-to-eye calibration methods based on pose estimation from 2d/2d correspondences via SfM w.r.t. 2d point error.

The impact of varying 2d point errors σ_{u_ε}, numbers of poses m, and numbers of rigidly coupled cameras n is evaluated as in the absolute pose esimation test. The results shown in Fig. 5.13–5.15 are in general similar.

However, enforcing rigid motion constraints - partially via `cSfM` and completely via rigidly coupled bundle adjustment `E2EBA` – has a stronger effect on pose estimation and eye-to-eye calibration in this application. The results also show a more distinctive improvement of the eye-to-eye calibration results from `vE2E` via rigidly coupled bundle adjustment as compared to the previous test.

Both experiments show that 2d point errors and number of images have the highest influence on eye-to-eye calibration from image features which is self-evident. While increasing the number of rigid motion constraints in `cAbs`/`cSfM` and `E2EBA` by taking more cameras into account at the same time can improve individual pose estimation accuracy, it has only minor impact on the eye-to-eye calibration results in the tested scenarios.

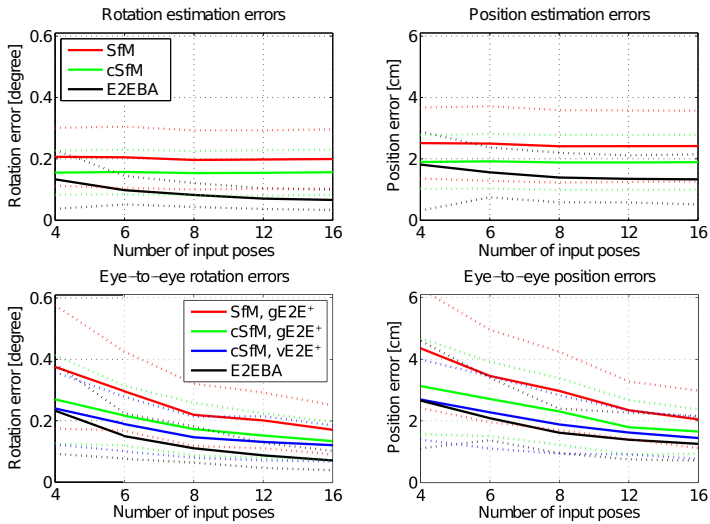

Figure 5.14. Comparison of eye-to-eye calibration methods based on pose estimation from 2d/2d correspondences via SfM w.r.t. number of poses per camera.

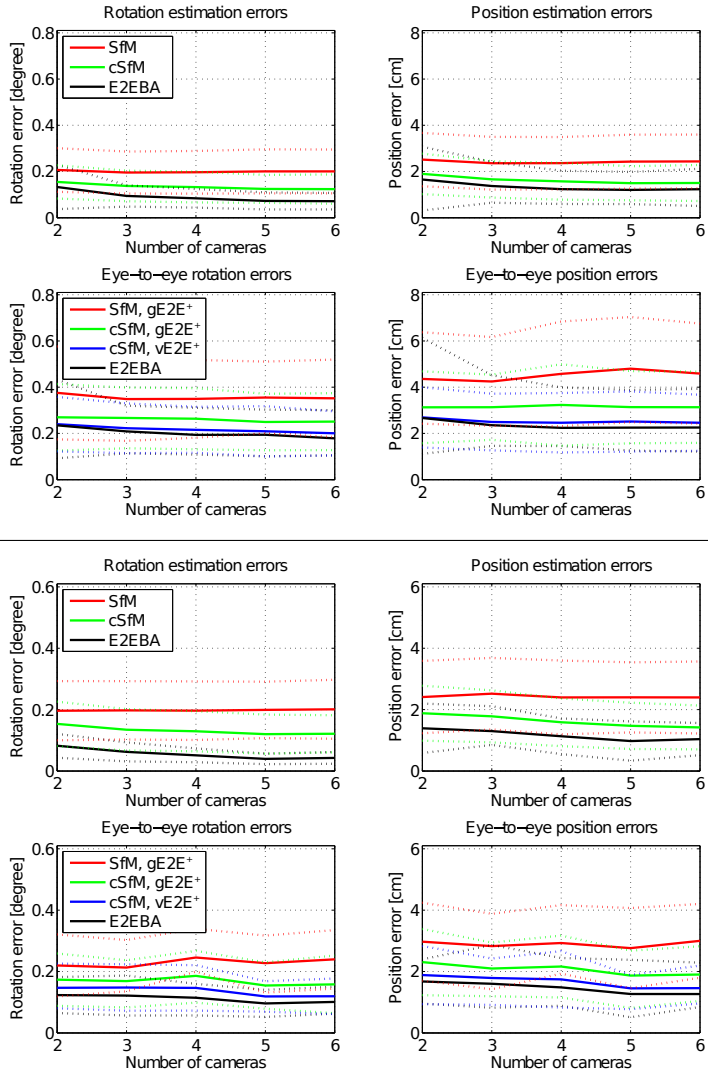

Figure 5.15. Comparison of eye-to-eye calibration methods based on pose estimation from 2d/2d correspondences via SfM w.r.t. number of rigidly coupled cameras (upper plots: $m = 4$ poses, lower plots: $m = 8$ poses).

5.7.2 Tests with Rendered Images

In this section, we use the entire Structure from Motion pipeline – including image preprocessing, feature detection and matching, pose estimation, and 3d point triangulation – for eye-to-eye calibration. We will evaluate the enforcement of partial rigid motion constraints and consider planar motion as a separate case. The tests are based on images rendered from virtual scenes in order to provide substantial ground truth data for comparison of the estimation results.

From the large number of rendered test sequences we will only present two cases that are similar to the real applications in Chapter 6, resembling typical indoor and outdoor calibration scenarios. In both cases the same virtual camera rig consisting of four cameras as illustrated in Fig. 5.16 was used. The ground truth extrinsic parameters can be found in Table 5.2. Orientations are descriptively noted down as Euler angles in XYZ order. The intrinsic parameters are defined by the image size 800×600 pixels and central principal point. The first and second camera facing the front have focal length $f = 1000\,\text{px}$ ($43.6^\circ \times 33.4^\circ$ field of view), the 3rd camera panned to the right has $f = 692.82\,\text{px}$ ($60^\circ \times 46.8^\circ$ fov), and the 4th camera observing the lower left side has $f = 800\,\text{px}$ ($53.1^\circ \times 41.1^\circ$ fov). The overlapping view fields of the front cameras and slight overlap with the other cameras are only used to judge the results visually.

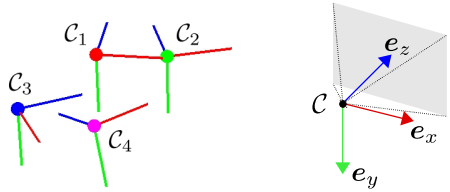

Figure 5.16. Virtual camera rig consisting of 4 cameras used in the tests. The color coding indicated by the camera centers (red: $\mathcal{C}_1$, green: $\mathcal{C}_2$, blue: $\mathcal{C}_3$, magenta: $\mathcal{C}_4$) distinguishes individual camera poses and scene parts in the following figures.

Table 5.2. Ground truth extrinsic parameters for the virtual camera system.

Cameras	$\Delta\alpha^*$	$\Delta\beta^*$	$\Delta\gamma^*$	Δt_x^*	Δt_y^*	Δt_z^*	$\|\Delta \boldsymbol{t}^*\|$
1-2	0°	-20°	0°	50	10	20	54.7 cm
1-3	0°	60°	0°	-50	20	-50	73.5 cm
1-4	-20.75°	-43.08°	0°	0	50	-20	53.9 cm

Indoor Scene The first test case emulates a typical indoor scene (see Fig. 5.17). The environment is enclosed by sparsely textured planar surfaces representing walls, floor, and ceiling. Several richly textured planes are distributed along the walls, representing for example windows, posters, or furniture in real scenarios. The camera was moved along a distance of $d = 3.73\,\text{m}$ while rotating up to $\alpha = 35^\circ$ from its initial position around different axes. The maximal angle between rotation axes of different orientation changes is $\beta = 90^\circ$. 61 images were captured during motion.

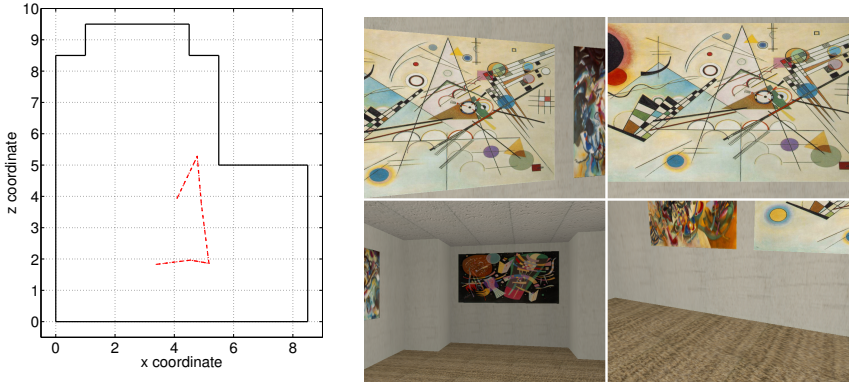

Figure 5.17. Floorplan sketch for the *indoor scene* test sequence and trajectory of the first camera within the scene (left), and images captured by all 4 cameras simultaneously (right). The picture textures are taken from the *WebMuseum* database © 1994–2006 Nicolas Pioch (`http://www.ibiblio.org/wm/about/license.html`).

The incremental Stucture from Motion pipeline was used to estimate poses for all cameras simultaneously for a specific subset of input images denoted as *keyframes*. To ensure distinct motion between subsequent images,

only every 10-th image was used as keyframe, providing 7 keyframes for this test. Features were tracked between keyframes using the KLT algorithm [LK81; TK91]. Following the basic SfM pipeline as shown in Fig. 5.5, the initial pose for each camera is fixed as $\mathbf{T}_1^{(i)} = [\mathbf{I} \mid \mathbf{0}]$, defining the camera reference coordinate frames $\mathcal{C}_i$. The pose for the second keyframe is estimated via RANSAC essential matrix estimation from 2d/2d correspondences (`Emat`) followed by nonlinear refinement (`Rel`), relative pose estimation with partial rigid motion constraints (`cRel`), separate local bundle adjustment (`BA`), and joint bundle adjustment with partial rigid motion constraints (`cBA`) to refine the initial scenes. The orientation and position estimation errors resulting from these five steps are plotted in the entries 1 to 5 in Fig. 5.18 (upper part). Apparently, computation of the essential matrix and extraction of the initial pose fails for the first camera. This is most likely due to the fact that this camera is looking frontally onto a plane in the first and second keyframe which constitutes a degenerate case for essential matrix estimation. However, by coupling the initial pose estimation to the other cameras via `cRel`, the ambiguity can be resolved and all cameras converge to approximately the same magnitude of error.

Afterwards, poses are tracked from 2d/3d correspondences while new 3d points are triangulated. Processing each keyframe consists of the steps: absolute estimation from 2d/3d correspondences via RANSAC (`PnP`) followed by nonlinear refinement (`Abs`), `cRel`, `BA`, and `cBA` to keep the reconstructions consistent. The pose error graph in Fig. 5.18 shows how enforcing partial rigid motion constraints is capable of reducing the pose estimation error in each keyframe leading to stable final results.

As a postprocessing step, the estimated camera poses are used for incremental eye-to-eye calibration, beginning with the linear solution `Tmat`$^+$, refined by the nonlinear method `quat`$^+$, visual eye-to-eye calibration (`vE2E`), and finally global eye-to-eye bundle adjustment (`E2EBA`). As can be seen in Fig. 5.18 (lower part), the estimation error is significantly decreased with every step, leading to $< 0.1^\circ$ error in orientation and $\approx 1\,\text{cm}$ error in position ($0.5\% - 2\%$) although the overall motion is rather limited.

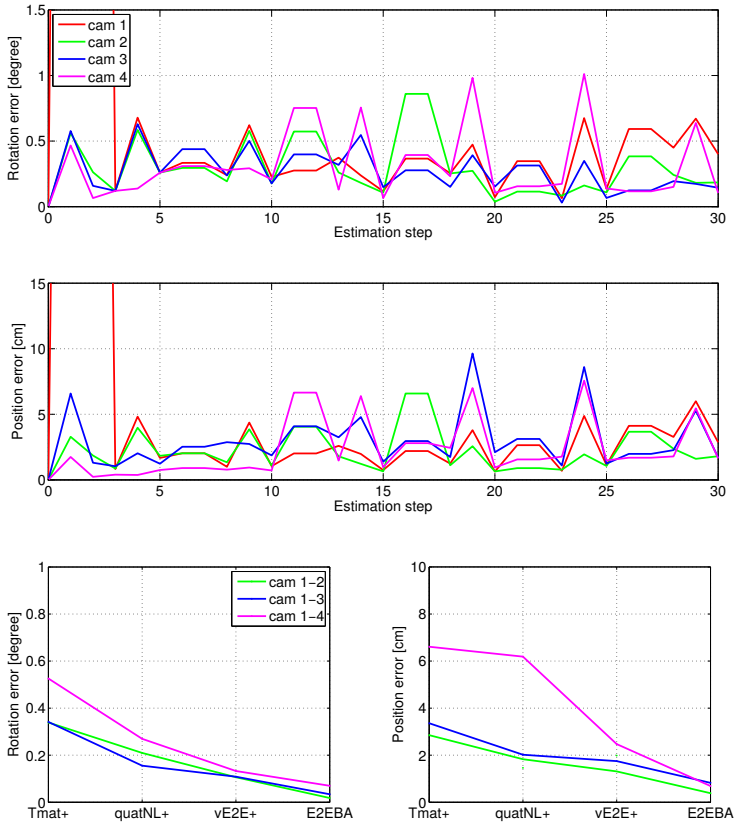

Figure 5.18. Results for Structure from Motion and eye-to-eye calibration for *indoor scene* test sequence using 4 cameras (top: pose estimation errors during SfM for each camera, bottom: final eye-to-eye calibration errors).

Outdoor Scene The second test case emulates a typical outdoor scene (see Fig. 5.19). The environment consists of a moderately textured ground plane and distant vertical planes representing the background and far away objects. Several richly structured cubes are scattered across the ground plane representing close buildings in real scenarios. The same

camera setup as in the previous test was used here, however, the whole camera rig is tilted downwards to provide a better view of the ground plane. 91 images were captured during almost planar motion over a range of $d = 7.5\,\mathrm{m}$ with maximal pan angle $\alpha \approx 45^\circ$. The maximal tilt/roll angle during motion is 4.51°.

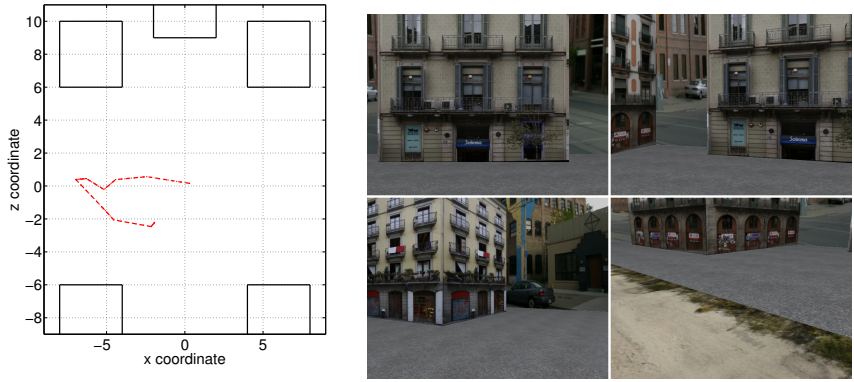

Figure 5.19. Floorplan sketch for the *outdoor scene* test sequence with trajectory of the first camera (left), and images captured by all 4 cameras simultaneously (right). Building textures are taken from the *CMP Facade Database* [TŠ13], background textures are taken from the *York Urban Database* [DEE08].

In contrast to the first test case, every 2nd image is used for pose estimation after initialization, providing 42 keyframes in total. Intermediate bundle adjustment is however only performed for every 10-th image. Errors accumulate during pose tracking as can be seen in Fig. 5.20 (upper part). Particular rise in pose estimation errors occurs when the pose is estimated from recently triangulated 3d points. The effect is partly counteracted by enforcing partial rigid motion constraints. Bundle adjustment after each 10-th image further reduces the error significantly by enforcing global consistency constraints on the scene reconstructed so far. Note that pose tracking of the 3rd camera fails after keyframe 22 since too many 3d points are lost from view.

In order to apply the planar eye-to-eye calibration approach described

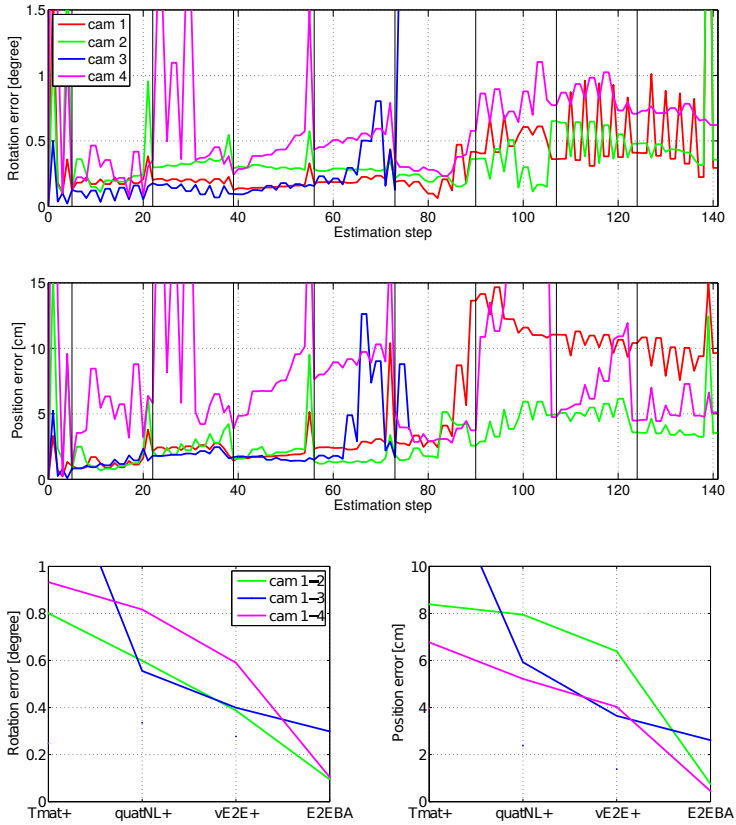

Figure 5.20. Results for Structure from Motion and eye-to-eye calibration for *outdoor scene* test sequence using 4 cameras (top: pose estimation errors during SfM for each camera, bottom: final eye-to-eye calibration errors).

in Sec. 3.6.2, normal $\boldsymbol{n}_i$ and height h_i of the ground plane w.r.t. each camera $\mathcal{C}_i$ are provided as additional input, simulating the case that a common reference plane has been identified prior to calibration. To model measurement inaccuracies, the ground truth normals $\boldsymbol{n}_i$ were rotated by 0.15° and errors $\varepsilon_h \sim \mathcal{N}(0, \sigma_h^2)$ with $\sigma_h = 3\,\text{mm}$ were added to h_i.

The final eye-to-eye calibration results are shown in Fig. 5.20 (lower part). The final results for the 3rd camera are less accurate, probably due to the limited range of orientation change observed during the first 22 keyframes.

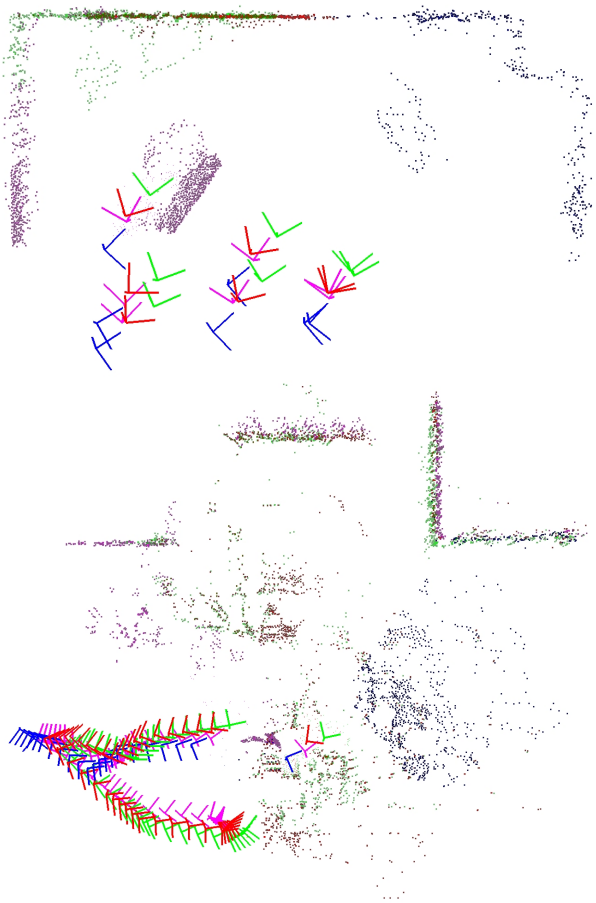

Figure 5.21. Renderings of the combined 3d models resulting from Structure from Motion and eye-to-eye calibration for *indoor scene* (top) and *outdoor scene* test sequence (bottom) using 4 cameras. The scenes were merged via `E2EBA`.

5.7.3 Complete Eye-to-Eye Calibration

In the following, complete eye-to-eye calibration from pair-wise eye-to-eye transformations based on the algorithms described in Sec. 5.6 is evaluated using the results from Sec. 5.7.2. Eye-to-eye transformations $\Delta\hat{\mathbf{T}}_{i,j}$ estimated via pair-wise eye-to-eye bundle adjustment are used as input data. The edges $E_{i,j}$ of the pose graph $\mathcal{G}$ are weighted by the average normalized reprojection error per measurement resulting from `E2EBA`.

The accuracy of complete calibration is measured by the average position and rotation error for all inferred pair-wise eye-to-eye transformations:

$$\epsilon_{\text{rot}} = \text{mean}\{d_{\angle}(\hat{\mathbf{R}}_i^{\mathsf{T}}\hat{\mathbf{R}}_j, \Delta\mathbf{R}_{i,j}^{*}) \mid 1 \leqslant i < j \leqslant n\}$$

and

$$\epsilon_{\text{pos}} = \text{mean}\{\|\hat{\mathbf{R}}_i^{\mathsf{T}}(\hat{\boldsymbol{t}}_j - \hat{\boldsymbol{t}}_i) - \Delta\boldsymbol{t}_{i,j}^{*}\| \mid 1 \leqslant i < j \leqslant n\}$$

where $\hat{\mathbf{R}}_i, \hat{\boldsymbol{t}}_i$ are the estimated absolute poses of the cameras within a common reference coordinate frame. In Table 5.3, the results from the graph-based solution followed by nonlinear optimization are compared to the initial solution inferred from pair-wise eye-to-eye transformations $\Delta\hat{\mathbf{T}}_{1,2}, \ldots, \Delta\hat{\mathbf{T}}_{1,n}$ by choosing $\mathcal{C}_1$ as reference arbitrarily.

In the *indoor scene* test sequence, choosing the first camera as reference provides the best result w.r.t. global consistency. This solution is also found by the graph-based approach. In the *outdoor scene* test sequence, the graph-based approach retrieves a better solution where the 2nd camera is used as reference. The final results are close to `E2EBA` using all cameras.

Table 5.3. Average eye-to-eye transformation errors for complete eye-to-eye calibration evaluated for *indoor scene* and *outdoor scene* test sequences.

	Indoor scene		**Outdoor scene**	
	ϵ_{rot}	ϵ_{pos}	ϵ_{rot}	ϵ_{pos}
Initial	$0.12^\circ \pm 0.05^\circ$	$0.91 \pm 0.32\,\text{cm}$	$0.35^\circ \pm 0.18^\circ$	$1.90 \pm 0.73\,\text{cm}$
Graph-based	$0.12^\circ \pm 0.05^\circ$	$0.91 \pm 0.32\,\text{cm}$	$0.22^\circ \pm 0.09^\circ$	$1.43 \pm 0.81\,\text{cm}$
Refined	$0.08^\circ \pm 0.04^\circ$	$0.98 \pm 0.35\,\text{cm}$	$0.18^\circ \pm 0.08^\circ$	$1.31 \pm 0.64\,\text{cm}$
Global BA	$0.04^\circ \pm 0.03^\circ$	$0.63 \pm 0.21\,\text{cm}$	$0.17^\circ \pm 0.10^\circ$	$1.27 \pm 0.75\,\text{cm}$

5.8 Summary

In this chapter, we described "visual" eye-to-eye calibration based on 2d/2d and 2d/3d correspondences and reprojection errors in contrast to the "geometric" methods based on Euclidean transformation correspondences and error measures in $\mathrm{SE}(3)$ discussed in Part I. Both geometric and visual eye-to-eye calibration are integrated into sequential and hierarchical Structure from Motion pipelines where geometric solutions are used as starting points for image-based refinement. A general rigidly coupled bundle adjustment approach was described that can be applied as a final step for global refinement of the individual reconstructions and eye-to-eye transformation parameters, obtaining a global model of the scene parts observed by the individual cameras. Furthermore, we discussed methods to achieve global consistency of pairwise eye-to-eye calibration of rigs consisting of many cameras. The evaluation with synthetic data proved the advantage of combining Structure from Motion and eye-to-eye calibration rather than decoupling pose acquisition and extrinsic calibration entirely.

We showed that enforcing partial rigid motion constraints, in particular equal rotation angle for corresponding relative rotations, during sequential Structure from Motion is able to increase the robustness as compared to individual unconstrained pose estimation. Moreover, drift of the relative reconstruction scale between rigidly coupled cameras can be monitored during Structure from Motion by evaluating the equal pitch constraint. Fixing this scale is however not as straightforward as constraining the rotation angle, especially since relative scale observations are rather sensitive to errors in the estimated poses.

Application of the whole Structure from Motion pipeline was presented for two rendered datasets representing typical indoor and outdoor calibration scenarios. In these and other experiments, a trade-off between confined motion range in order to support visual pose tracking and distinct orientation and position changes which are beneficial for eye-to-eye calibration had to be made.

The proposed eye-to-eye calibration framework is tested with two real camera systems in the following chapter.

Chapter 6

Applications

6.1 Calibration of a Portable Camera System

Portable devices combining multiple cameras are useful for practical applications such as on-site visual inspection and surveying of buildings or Augmented Reality. It is often advantageous to supplement cameras providing the user's field of vision by cameras with separate orientation that are used for localization. An example is given by Augmented Reality binoculars or see-through displays where a wide-angle lens camera is used to provide scene context and recover the position and orientation of the device while cameras with narrower field of view capturing the front are used for the actual augmentation (see [HK08]). In specific environments, cameras for localization looking upwards, downwards, or sidewards can be used in combination with tracking markers on either the ceiling, the floor, or the walls.

In this section, eye-to-eye calibration of a portable multi-camera system in the style of such devices is evaluated. Since the mobility of the system allows for a considerably large variety of different motions, we will apply calibration techniques for the general motion model here.

6.1.1 System Description

The camera system used for this test consists of two *Point Grey Grasshopper®* (GRAS-20S4C-C) cameras equipped with *Schneider-Kreuznach Cinegon 1.8/4.8* lenses. The second camera is located approx. 25 cm to the right of the first

camera, 5 – 10 cm above and behind it, and is rotated towards the upper left direction (see Fig. 6.1). Note that the cameras have partly overlapping fields of view that are used to evaluate the results visually.

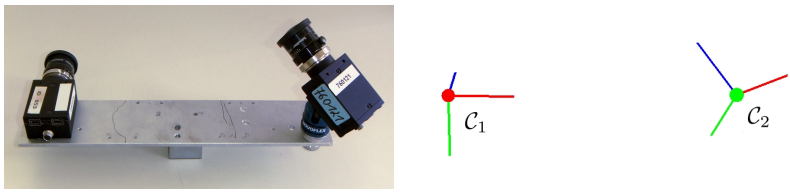

Figure 6.1. Image of portable camera system consisting of two cameras (left) and estimated setup resulting from eye-to-eye calibration (right).

The camera was moved in front of a scene composed of textured cardboard boxes shown in Fig. 6.2 along a distance of $d \approx 1\,\text{m}$ while rotating up to $\alpha \approx 60^\circ$ from the initial position around different axes. 400 RGB images with $800 \times 600\,\text{px}$ resolution were captured during motion from which 245 images were used for Structure from Motion.

Figure 6.2. Test scene for portable camera system (left) and images captured by both cameras simultaneously during motion (right).

Intrinsic camera parameters were estimated from 95 resp. 91 images showing a checkerboard with 8×5 tiles of size 11.7 cm, resulting in final

average reprojection errors of 0.06 ± 0.04 px resp. 0.07 ± 0.03 px. The estimated parameters are shown in Table 6.1.

Table 6.1. Estimated intrinsic camera parameters for portable rig.

Camera	Resolution	Field of view	f_x	f_y	p_x	p_y
1 left	800×600	$70^\circ \times 56^\circ$	567.17	569.26	382.3	305.3
2 right	800×600	$70^\circ \times 56^\circ$	567.66	570.02	414.3	291.0

6.1.2 Results

Structure from Motion was computed for every 4-th image in the input sequence. Initial relative pose estimation succeeded after 17 frames. Every 12-th frame was used as keyframe for incremental bundle adjustment. The final reconstruction merged by the eye-to-eye bundle adjustment step is shown in Fig. 6.4. The visualization shows that 3d points in the central part of the scene that was observed by both cameras during SfM are well aligned. The average reprojection error per 2d point measurement resulting from `E2EBA` is 1.07 px.

The estimated eye-to-eye transformation parameters are listed in Table 6.2 under `SfM+E2EBA`. Rotation angles are defined w.r.t. YXZ order here since this sequence provides more intuitive angles than XYZ order here. Absolute scaling of the translation vector was achieved by manually measuring the distance between two 3d points within the scene part of the first camera reconstructed during initialization. The distance to a third manually measured point differed by approx. 1.5 mm indicating that these 3 points were reconstructed very accurately and consistently.

Table 6.2. Eye-to-eye transformation parameters estimated for the portable camera system via Structure from Motion vs. from images of a checkerboard.

Method	$\Delta\alpha$	$\Delta\beta$	$\Delta\gamma$	Δt_x	Δt_y	Δt_z	$\|\Delta \boldsymbol{t}\|$
`SfM+E2EBA`	30.22°	-43.13°	-2.92°	24.68	-7.94	-5.36	26.47 cm
`Abs+E2EBA`	30.17°	-43.06°	-3.10°	24.83	-8.50	-7.40	27.26 cm
Difference	0.06°	0.05°	0.20°	0.15	0.57	2.05	2.13 cm

To evaluate the accuracy of the results, a second eye-to-eye calibration was performed using 69 images of two checkerboards instead of SfM to estimate local camera poses, providing an instance of the *eye-and-world calibration* problem. The corresponding eye-to-eye calibration problem was obtained by transforming all poses into the coordinate frame of the first image. Note that the location of the checkerboards with respect to each other is not known and only one checkerboard is used by each camera respectively, i. e., no stereo calibration is computed although both checkerboards are at least partly visible in several input images. The same eye-to-eye calibration pipeline as for the Structure from Motion case is applied here, with the only modification that 3d points are fixed during eye-to-eye bundle adjustment, resulting in an average reprojection error of 0.15 px. The estimated eye-to-eye transformation parameters are listed in Table 6.2 under `Abs+E2EBA`.

The results are in fact very close to each other, indicating that the SfM-based solution is considerably precise, especially for the orientation. The most notable difference of approx. 2 cm is observed in the eye-to-eye translation along the z-axis of the first camera. Analysis of the eigenvectors of the estimated parameter covariance matrix $\boldsymbol{\Sigma}_{\Delta\mathbf{q},\Delta t}$ resulting from `SfM+E2EBA` confirms that the direction in which the position parameter uncertainty is the strongest is given by $\boldsymbol{v} = (0.2941, 0.4516, 0.8424)$.

Figure 6.3. Rendering of a checkerboard detected in the first camera's image (left) and transformed into the second camera's image via the estimated eye-to-eye transformation (center: `SfM+E2EBA`, right: `Abs+E2EBA`).

In order to evaluate the estimated parameters visually, 12 images of the checkerboard used for `Abs+E2EBA` were selected where the checkerboard of the first camera is complete visible in the second camera's image.

The pose of the checkerboard was detected in the first camera's images, transformed into the coordinate frame of the second camera via $\Delta\mathbf{T}^{-1}$, and rendered into the second image. Figure 6.3 shows an example from the dataset. The average reprojection errors of the checkerboard corners are 1.49 ± 0.83 px for `SfM+E2EBA` and 0.68 ± 0.44 px for `Abs+E2EBA`. Hence, judging from visual inspection, the calibration results are sufficiently accurate for purposes like common Augmented Reality applications.

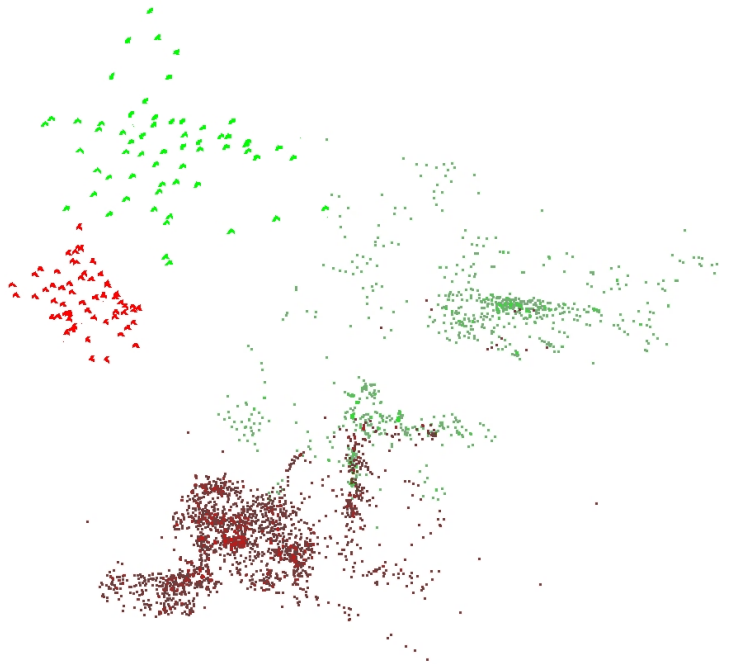

Figure 6.4. Rendering of the combined 3d models resulting from Structure from Motion and eye-to-eye calibration for the portable camera system (side view). The scenes were merged via `E2EBA`.

6.2 Calibration of Vehicle-Mounted Cameras

Multi-camera systems with large combined field of view are of specific interest in the automotive industry, e. g., for advanced driver assistent systems or autonomous navigation. As a review of the recent literature on extrinsic camera calibration without overlapping views reveals, special attention is spent on the case of vehicle-mounted cameras (see for example [Pag10; Muh11; RPK11]).

Calibration of vehicle-mounted cameras using eye-to-eye calibration techniques is complicated by the fact that motion of the camera system is in general restricted to almost planar motion. Moreover, motion of the vehicle is often restricted by the Ackermann steering principle, i. e., sideward motion or pure rotations cannot be performed. In Sec. 3.6.2 we described methods for eye-to-eye calibration from planar motion that will be tested with real data in this section.

6.2.1 System Description

The proposed algorithms for eye-to-eye calibration from planar motion are tested with datasets captured with the *Urban Traffic Assistant (UTA)* system of Daimler AG as part of the research initiative *AKTIV – Adaptive and cooperative technologies for intelligent transport* [EG10].

The setup of the UTA test vehicle consists of five cameras that are assembled to view the front, side, and rear part of the environment. Two sets of stereo cameras are located behind the front windshield and the right side window capturing the scene in front and to the right of the car respectively. Note that the front cameras are mounted upside down. The wide-angle lens camera on the rear side is fixed to the trunk lid, slighted tilted towards the ground. Figure 6.5 illustrated the locations and orientations of the cameras on the test vehicle. The coordinate frames of the cameras are identified by $\mathcal{C}_1$ (rear), $\mathcal{C}_2$ (front left), $\mathcal{C}_3$ (front right), $\mathcal{C}_4$ (side left), and $\mathcal{C}_5$ (side right). In addition to the cameras, UTA is equipped with sensors like GPS, yaw rate sensors, and inertial sensors, providing pose measurements that will also only be used to evaluate the results of

visual odometry via Structure from Motion. We will however use a single distance measurement from the odometer to provide an absolute scale for the reconstructed scenes.

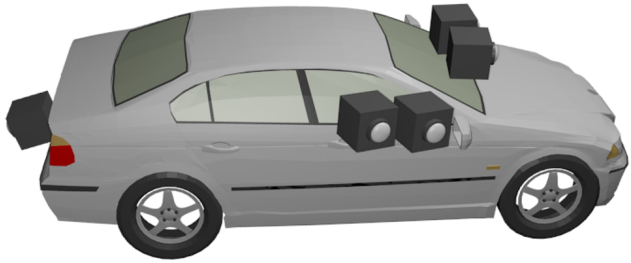

Figure 6.5. Illustration of cameras mounted onto the UTA test vehicle.

Intrinsic calibration Intrinsic parameters were estimated for each camera individually using images of a checkerboard as described in Sec. 4.2.4. Extrinsic parameters of the front and side stereo cameras were estimated from stereo correspondences yielding relative poses $\Delta\mathbf{T}^*_{2,3}$ and $\Delta\mathbf{T}^*_{4,5}$.[1] However, we will use this information only for evaluation of the parameters resulting from eye-to-eye calibration without overlapping views. The resulting intrinsic and extrinsic parameters are listed in Table 6.3 and 6.4. Extrinsic rotations $\Delta\mathbf{R}^*_{2,3}$ and $\Delta\mathbf{R}^*_{4,5}$ are described as Euler angles in XYZ order for the sake of clarity. Note that the right front camera is located to the left within $\mathcal{C}_1$ since both cameras are mounted upside down.

Table 6.3. Estimated intrinsic camera parameters for UTA test vehicle.

Camera	Resolution	Field of view	f_x	f_y	p_x	p_y
1 rear	640×480	$64^\circ \times 51^\circ$	508.32	508.32	293.5	270.4
2 front left	640×480	$42^\circ \times 32^\circ$	827.39	829.36	332.2	238.3
3 front right	640×480	$42^\circ \times 32^\circ$	828.69	831.15	337.3	251.8
4 side left	640×480	$42^\circ \times 32^\circ$	829.91	835.05	294.2	279.7
5 side right	640×480	$42^\circ \times 32^\circ$	827.50	828.86	331.9	248.6

[1] The calibration software from [SBK08] was used for intrinsic and extrinsic calibration.

Table 6.4. Extrinsics for the front and side stereo camera systems estimated via classical stereo calibration.

Cameras	$\Delta\alpha^*$	$\Delta\beta^*$	$\Delta\gamma^*$	Δt_x^*	Δt_y^*	Δt_z^*	$\|\Delta \boldsymbol{t}^*\|$
2-3 front	-0.17°	0.76°	-0.62°	-24.2	0.2	-0.8	24.2 cm
4-5 side	2.79°	2.51°	-0.35°	23.6	-0.9	9.5	25.5 cm

Ground plane detection In order to apply the planar eye-to-eye calibration approach described in Sec. 3.6.2, the location of a virtual ground plane is estimated for all cameras from images of a vertical checkerboard that is moved along on the floor over a range of $1.5 - 2$ m. Example images from the calibration sequences consisting of $14 - 21$ images per camera are shown in Fig. 6.6. As result we obtain the local plane parameters[2] $(\boldsymbol{n}_i, h_i)$ with normal $\boldsymbol{n}_i$ and distance h_i for each camera $\mathcal{C}_i$, $i \in \{1, \ldots, 5\}$ listed in Table 6.5. The standard deviations of the estimated height values are below 1 mm for all cameras. Note that the location of the virtual ground plane is located 115 cm below the origin of the checkerboard within the upper left corner.

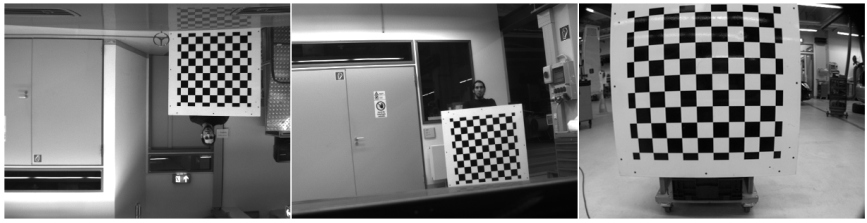

Figure 6.6. Images of a checkerboard pattern used for ground plane registration of cameras on the UTA test vehicle (from left to right: left front camera $\mathcal{C}_2$, left side camera $\mathcal{C}_4$, wide-angle rear camera $\mathcal{C}_1$). Each square has size 8×8 cm.

Image acquisition Image sequences consisting of up to 600 grayscale images with 640×480 px resolution were captured synchronously with all 5 cameras while the test vehicle was driving on planar ground. The paths contain straight parts where the car is driving forwards as well as several

[2] The 3d plane with normal $\boldsymbol{n}$ and height h is defined by $\Pi = \{\boldsymbol{X} \in \mathbb{R}^3 \mid \boldsymbol{n}^\intercal \boldsymbol{X} = h\}$.

Table 6.5. Estimated ground plane parameters for cameras in UTA test vehicle.

Camera	Plane normal vector $\boldsymbol{n}$	$\angle(\boldsymbol{n}, -\boldsymbol{e}_y)$	$\lvert h \rvert$
1 rear	$(-0.022684, -0.980634, -0.194532)$	11.29°	79.1 cm
2 front left	$(-0.006033, 0.999918, -0.011314)$	179.27°	116.0 cm
3 front right	$(-0.018519, 0.999791, -0.008641)$	178.83°	117.6 cm
4 side left	$(-0.092643, -0.995691, -0.004218)$	5.32°	104.3 cm
5 side right	$(-0.094759, -0.991833, 0.085370)$	7.33°	104.5 cm

turns with different circumference ($3-4$ turns per image sequence). Two examples for simultaneously captured images and the path of the car for sequences of 500 images are shown in Fig. 6.7 and 6.8. The trajectories displayed in the lower right image have been estimated from odometry data and are used to give a visual impression of the camera motion during capturing the scene. The width of the odometry plots corresponds to approx. 30 m. Feature detection was limited to the valid image region that was set manually for each camera prior to image processing.

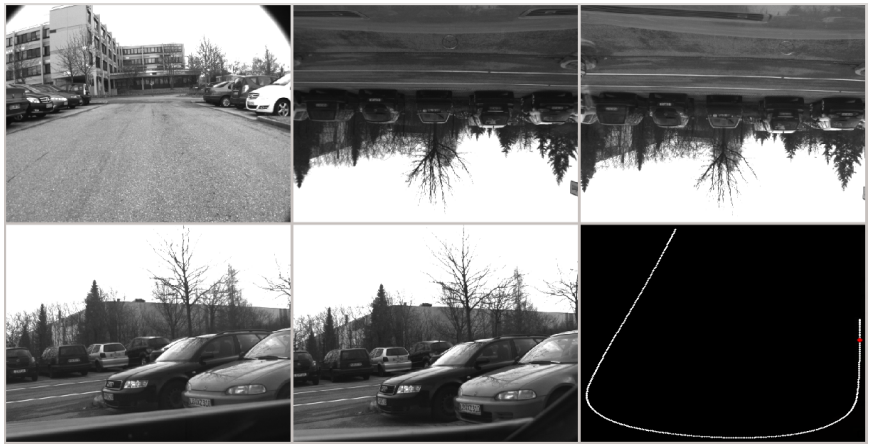

Figure 6.7. Images of the cameras captured during a U-shaped drive of the UTA test vehicle (from top left to bottom right: wide-angle rear camera $\mathcal{C}_1$, upside-down front cameras $\mathcal{C}_2$, $\mathcal{C}_3$, side cameras $\mathcal{C}_4$, $\mathcal{C}_5$, path estimated from odometry data).

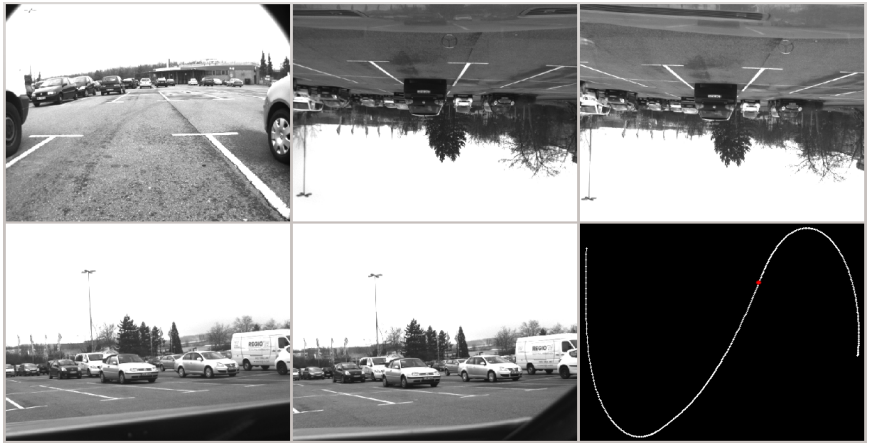

Figure 6.8. Images of the cameras captured during an S-shaped drive of the UTA test vehicle (order as in Fig. 6.7).

6.2.2 Results

A major problem in this case was the selection of a subsequence from the captured video that was suitable for SfM. Initialization during a straight drive is problematic for the cameras looking to the front since triangulation is likely to fail for forwards motion. Hence, we selected subsequences during cornering. Since pose estimation over long distance proved to be difficult using standard SfM techniques for the given camera setup, only up to 100 frames were tracked after initialization. The results presented here have been computed from the beginning of the sequence shown in Fig. 6.7. Initialization succeeded after 48 frames. Poses were estimated for every 4-th frame, and incremental bundle adjustment was applied every 12-th frame. The reconstructed 3d point clouds and camera trajectories merged by the final `E2EBA` step are shown in Fig. 6.11.

To provide absolute scale for comparison with the stereo calibration results, all resulting poses were scaled with the absolute distance driven up to the 48-th image as measured by the vehicle odometry (6.58 m). The estimated camera configuration is illustrated in Fig. 6.9. Despite the rather short

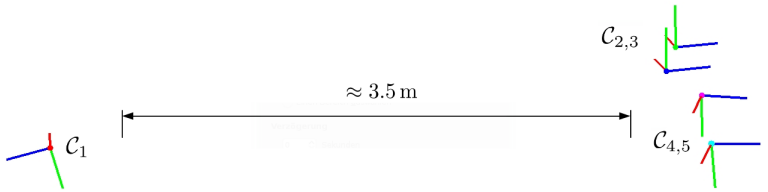

Figure 6.9. Estimated camera setup resulting from eye-to-eye calibration.

input sequence and lacking variation of motion, the results are in general reasonably close to the real setup. Note that the vertical offset of the cameras w.r.t. the motion plane cannot be estimated from planar motion. In the presented calibration results, the cameras were shifted due to the height of the ground planes measured prior to SfM as described above. The estimated and scaled eye-to-eye transformation parameters are listed in Table 6.6. The table lists the pair-wise transformations that have been selected by complete eye-to-eye calibration as the most reliable results. Comparison of the transferred poses to the results from stereo calibration (last two rows) reveals that the camera orientation has been reconstructed rather accurately. The position estimates are significantly less precise, although the error is still within the range of $1-2\%$ considering that the relative poses have been transferred via the distant rear camera.

Table 6.6. Eye-to-eye transformation parameters estimated for the vehicle-mounted cameras via Structure from Motion with absolute scale from vehicle odometry and vertical offset from ground plane detection.

Cam.	$\Delta\alpha$	$\Delta\beta$	$\Delta\gamma$	Δt_x	Δt_y	Δt_z	$\|\Delta \boldsymbol{t}\|$
1-2	$-172.01°$	$-6.14°$	$-0.98°$	-16.8	33.9	-358.8	360.8 cm
1-3	$-172.15°$	$-4.95°$	$-1.65°$	-38.7	32.3	-353.8	357.4 cm
1-4	$-171.74°$	$-32.21°$	$-176.86°$	-65.6	46.0	-353.8	362.8 cm
1-5	$-174.53°$	$-34.40°$	$-178.71°$	-93.1	44.9	-345.2	360.3 cm
2-3	$-0.16°$	$1.19°$	$-0.65°$	-22.3	0.5	-2.8	22.5 cm
4-5	$2.42°$	$2.09°$	$-0.36°$	27.9	-1.4	7.3	28.9 cm
Diff.							
2-3	$-0.01°$	$-0.43°$	$0.03°$	1.9	0.3	-2.0	2.8 cm
4-5	$0.37°$	$0.42°$	$0.02°$	4.3	-0.5	-2.2	4.9 cm

The estimated parameters were evaluated visually as in the previous test case by projecting a checkerbord detected in images of the left front camera $\mathcal{C}_2$ into cameras with partly overlapping views, i. e., the right front camera $\mathcal{C}_3$ and left side camera $\mathcal{C}_4$ (see Fig. 6.10), exhibiting notable reprojection errors due to the absolute horizontal position estimation errors. It has to be noted that pairwise eye-to-eye calibration from the stereo camera pairs yields distinctly better results. However, this follows indirectly from the large view overlap since SfM performs very similar for these cameras.

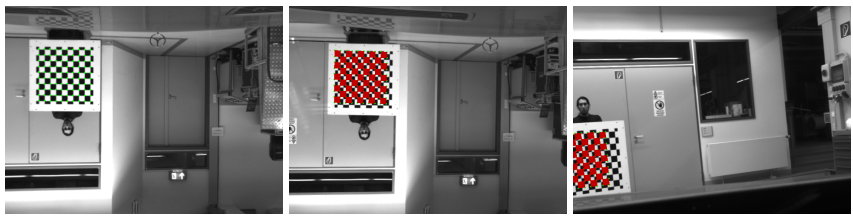

Figure 6.10. Rendering of a checkerboard detected by camera 2 (left) and transformed into the views of camera 3 (center) and camera 4 (right) via the estimated eye-to-eye transformations.

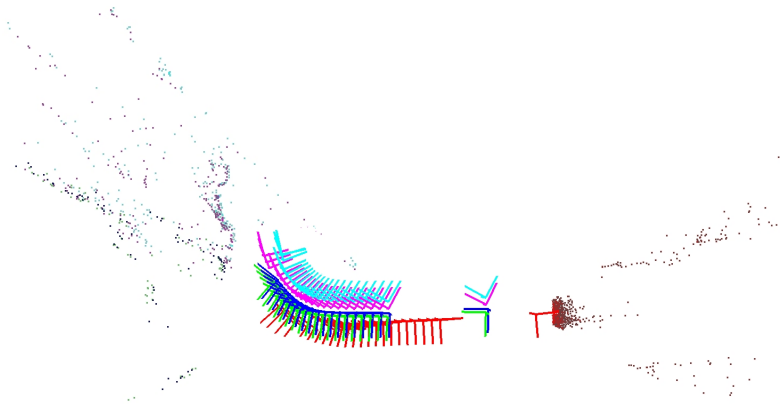

Figure 6.11. Rendering of the combined 3d models resulting from Structure from Motion and eye-to-eye calibration for the vehicle-mounted cameras (top view). The scenes were merged via `E2EBA`.

6.3 Summary

In this chapter we demonstrated the applicability of the proposed eye-to-eye calibration methods based on visual measurements for two real test cases related to calibration of portable and vehicle-mounted multi-camera systems.

Due to the constrained motion of the vehicle in the latter case, additional measurements of a common reference plane had to be included to provide a full calibration. While the orientation of the camera could be recovered accurately, the translational parameters are less well defined. This is also due to the fact that the absolute positions measured by visual odometry are significantly larger than the offset between the cameras. Moreover, positional localization of the vehicle via Structure from Motion suffers from error accumulation and ill-posed triangulation for the cameras looking along the driving direction. The available test data did not allow for large-scale Structure from Motion but succeeded only for short sequences, reducing the amount of variation in observable pose changes further.

In the first test case, SfM-based eye-to-eye calibration and eye-and-world calibration from images of an object with known geometry were evaluated and compared to each other. Both methods provided very similar results, indicating that the Structure from Motion framework was able to handle this problem very accurately. The estimated eye-to-eye parameters were evaluated visually by transforming projections of an object between the camera views. This demonstrated that the extrinsic calibration of the camera setup is suitable for purposes like Augmented Reality applications. However, in both test cases a certain degree of manual user interaction is necessary to provide a solution with absolute metric scale, given that no additional sensors or markers are used.

Chapter 7

Conclusions

7.1 Summary

Multi-view image capturing using camera systems comprised of various cameras has many practical applications, providing advantages for computer vision tasks like 3d scene reconstruction, object tracking, and visual odometry due to the large combined field of view. In contrast to specialized omnidirectional image capturing systems based on lenses or mirrors, coupling several default camera that are readily available off-the-shelf allows for inexpensive and flexible setups. The number of cameras involved can be further reduced by limiting the view overlap to a minimum.

Beside intrinsic camera calibration, solving the *eye-to-eye calibration* problem (i. e., finding the coordinate frame transformations between the coupled cameras) is a crucial prerequisite for the use of multi-camera systems. For sparse camera systems, methods that do not rely on visual correspondences between cameras are needed. Therefore, in this thesis we investigated methods that do not depend on information from mutually visible parts of the scene but on geometric constraints between rigidly coupled motion. First, we solved this problem as an instance of classical hand-eye calibration w.r.t. error measures based on relative poses. Motivated by results from Structure from Motion w/o overlapping views we extended this approach to use error measures based on visual correspondences for individual cameras. Numerical solutions for global refinement of pose parameters, 3d scene geometry, and eye-to-eye transformations based on image measurements via rigidly coupled bundle adjustment were described to provide accurate calibration results.

Specialized solutions were presented for critical motion configurations, in particular for the case of planar motion. However, these solutions require further knowledge about the captured scene such as the location of a common plane. Otherwise, eye-to-eye transformations can only be partially recovered.

A common disadvantage of visual odometry using SfM techniques is pose drift due to error accumulation which can severely impact the results of eye-to-eye calibration. Therefore, we presented an approach to stabilize local egomotion estimation for rigidly coupled cameras by imposing partial rigid motion constraints in joint refinement of all camera poses relating to the same point in time. This techniques proved to be capable of increasing the pose estimation accuracy and attenuating drift in experimental evaluations.

Considering the case of multi-camera system consisting of more than two cameras, it is often not possible to capture synchronous images for all cameras at the same time due to practical limitations. For this reason, we also described methods to extend pair-wise eye-to-eye calibration to complete extrinsic calibration in a meaningful way.

We compared implementations of the proposed methods and evaluated the limits of the achievable precision based on synthetic data. Practical feasibility of the presented methods and concepts was demonstrated by implementing a Structure from Motion framework with integrated eye-to-eye calibration that was used for extrinsic camera calibration from video streams in two real-world applications.

Given that adequate conditions for visual odometry estimation are met – e. g., scenes with textured surfaces, appropriate lighting, and availability of precise camera model parameters – eye-to-eye calibration from images of camera w/o overlapping views using SfM techniques can provide proper extrinsic parameters that are suitable for applications like Augmented Reality or environment surveillance. Comparing our experimental results with existing extrinsic calibration methods for stereo cameras, we found that recovering extrinsic parameters with accurate metric scaling still requires to capture some images of fiducial markers or objects with known geometry. To achieve absolute scale, it is theoretically sufficient to provide

at least one such image for a single camera in the rig. As a matter of fact, extrinsic calibration using general Structure from Motion techniques is distinctively less accurate than calibration from markers as presented by Lébraly et al. [Léb+11], providing position errors below 1 mm and orientation errors below 0.05° given sufficiently high image resolution, although these can also be used within our framework. However, the demanded accuracy depends on the actual application.

Although the presented framework provides a purely vision-based solution to extrinsic camera calibration, it can also be used as a basis for extrinsic calibration of integrated multi-sensor systems, comprising for example inertial measurement units, odometers, laser rangefinders, or depth sensors next to conventional cameras.

7.2 Future Work

In this work we focused on rigidly coupled pose estimation based on simultaneously captured images. However, when using unsynchronized cameras or video cameras with different capturing frame rates, camera synchronization methods for multi-view capturing must be applied. We assume that the equal angle and equal pitch constraints resulting from rigid motion coupling should be helpful to solve this problem, e. g., by finding correlations between the rotation angle and pitch profiles of different cameras over time. Investigations into this direction provide an interesting task for future work.

Partial rigid motion constraints were used to stabilize rigidly coupled pose estimation. The same approach could also be applied to other problems like joint intrinsic calibration, relative rotation estimation, or homography estimation resulting from views of a planar scene. It would also be interesting to investigate if direct solutions for relative and absolute pose estimation such as the 5-point algorithm could benefit from these constraints.

We found that a major drawback of global eye-to-eye bundle adjustment used as a final step in extrinsic calibration of multi-camera systems with

many cameras is the large number of parameters involved. In the practical application in Sec. 6.2 where five cameras were involved, we circumvented this problem by computing pair-wise eye-to-eye calibrations only. Since eye-to-eye calibration depends primary on precise local egomotion as opposed to detailed scene geometry, there are in general two alternative strategies to tackle this problem: Either sophisticated methods to select the most salient 3d points could be applied to keep the number of parameters for the scene geometry as low as possible. On the other hand, bundle adjustment without explicit modeling of 3d points as described by Rodríguez et al. [RLR11] could be investigated for this purpose.

The fundamental idea of integrating rigid motion constraints into Structure from Motion was realized within a very basic SfM implementation. Depending on the field of application, modern Structure from Motion approaches include additional features to increase robustness (e. g., compensation of motion blur and overexposure, removal of non-static background, or robust key frame selection) and efficiency (e. g., parallelization of pose estimation and scene reconstruction or hierarchical processing steps, computation on the GPU). In general, the methods and concepts proposed in this work could be fit easily into existing SfM implementations. However, certain parallel or hierarchical workflows might require adaptations of the eye-to-eye calibration procedure to allow for full integration. It would be interesting to integrate the proposed methods into established SfM implementations such as *Bundler* [SSS06] for hierarchical SfM, *PTAM* [KM07] for parallel tracking and mapping, or *VisualSfM* [Wu11] for hardware-accelerated Structure from Motion.

Part III

Appendix

Appendix A

Geometry

A.1 Quaternion Algebra

This section briefly summarizes the properties and algebra of real and dual quaternions which are used for rotation and rigid motion representation in the context of this work for the case that the reader is not familiar with these concepts. For a comprehensive discussion of quaternions within the context of rigid body motion we refer the reader to [Del12].

A.1.1 Quaternions

In mathematics, quaternions[1] describe a type of higher complex number system which extends the real number space by three additional imaginary dimensions. The space of quaternions is given by

$$\mathbb{H} = \{a + \mathbf{i}b + \mathbf{j}c + \mathbf{k}d \mid a, b, c, d \in \mathbb{R}\}$$

with imaginary units $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ with the property $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$.

Quaternions $\mathbf{q} \in \mathbb{H}$ are often represented as pairs $(\boldsymbol{q}, q)$ where the scalar part $q \in \mathbb{R}$ and vector part $\boldsymbol{q} \in \mathbb{R}^3$ are the coefficients of the real and imaginary parts of the quaternion. An equivalent representation is the coefficient vector $\mathbf{q} \in \mathbb{R}^4$ with $\boldsymbol{q} = (q_1, q_2, q_3)^\intercal$ and $q = q_4$ which is directly

[1] also named *Hamiltonian numbers* after the Irish mathematician and physicist William Rowan Hamilton (*1805, †1865) who described them first in 1843 [Ham44]

related to the *Euler-Rodrigues vector* used to describe 3d rotations. Both representations are used interchangeably in this work.

The following operations are defined on quaternions:

Addition:	$\mathbf{q} + \mathbf{q}'$	$= (\boldsymbol{q} + \boldsymbol{q}', q + q')$
Scalar product:	$\lambda \mathbf{q}$	$= (\lambda \boldsymbol{q}, \lambda q)$
Multiplication:	$\mathbf{q} \cdot \mathbf{q}'$	$= (q\boldsymbol{q}' + q'\boldsymbol{q} + \boldsymbol{q} \times \boldsymbol{q}', qq' - \boldsymbol{q}^\intercal \boldsymbol{q}')$
Conjugate:	$\bar{\mathbf{q}}$	$= (-\boldsymbol{q}, q)$
Magnitude:	$\|\mathbf{q}\|$	$= \sqrt{q^2 + \boldsymbol{q}^\intercal \boldsymbol{q}} = \sqrt{\mathbf{q}^\intercal \mathbf{q}}$
Inverse:	$\mathbf{q}^{-1}$	$= \frac{1}{\|\mathbf{q}\|^2} \bar{\mathbf{q}}$

Note that within quaternion algebra, scalar values are often considered as equivalent to quaternions with zero vector part and real 3-vectors are considered as equivalent to quaternions with zero scalar part (*pure quaternion*). Obviously, the identity element of $\mathbb{H}$ with respect to multiplication is given by $\mathbf{1} = (\mathbf{0}, 1)$ while the additive identity or zero element is $\mathbf{0} = (\mathbf{0}, 0)$.

Since quaternion multiplication is bilinear in the entries of the quaternions, left/right multiplication by $\mathbf{q}$ can be described in terms of matrix-vector-multiplication by matrices $\mathbf{M}^\ell_\mathbf{q}, \mathbf{M}^r_\mathbf{q} \in \mathbb{R}^{4\times 4}$ using the 4-vector representation for quaternions:

$$\mathbf{M}^\ell_\mathbf{q} = \begin{pmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{pmatrix}, \quad \mathbf{q} \cdot \mathbf{q}' = \mathbf{M}^\ell_\mathbf{q} \mathbf{q}' \tag{A.1}$$

and

$$\mathbf{M}^r_\mathbf{q} = \begin{pmatrix} q_4 & q_3 & -q_2 & q_1 \\ -q_3 & q_4 & q_1 & q_2 \\ q_2 & -q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{pmatrix}, \quad \mathbf{q}' \cdot \mathbf{q} = \mathbf{M}^r_\mathbf{q} \mathbf{q}' \tag{A.2}$$

A unit quaternion has magnitude $\|\mathbf{q}\| = 1$. Hence, the multiplicative inverse $\mathbf{q}^{-1}$ is given by the conjugate quaternion $\bar{\mathbf{q}}$. Unit quaternions are of special interest since they can be used to describe rotations in 3d space as explained in Sec. 2.3.2.

Minimal Parametrization for Unit Quaternions

Ikits [Iki00] describes a simple minimal parametrization for unit quaternions within the context of corregistration of pose measurement devices which is closely releated to the hand-eye calibration problem. Assuming w.l.o.g. that p_4 is the entry with largest absolute value in $\mathbf{p}$ with sign $\sigma \in \{-1, 1\}$, a unit quaternion $\mathbf{q}$ close to $\mathbf{p}$ can be expressed with the remaining entries $\boldsymbol{\varrho} = (q_1, q_2, q_3)$ and $q_4 = \sigma\sqrt{1 - \boldsymbol{\varrho}^\intercal\boldsymbol{\varrho}}$. However, this parametrization is only well-defined under the constraint $\boldsymbol{\varrho}^\intercal\boldsymbol{\varrho} \leqslant 1$.

Alternatively, unit quaternions can be described in the tangential hyperplane $\Omega_{\perp\mathbf{p}} := \{\mathbf{q} \mid \mathbf{q}^\intercal\mathbf{p} = 1\}$ with respect to $\mathbf{p}$. Quaternions in $\Omega_{\perp\mathbf{p}}$ are parametrized with coordinates $\boldsymbol{\varrho} \in \mathbb{R}^3$ and projected onto the unit sphere to obtain the corresponding unit quaternion. Assume for instance that $\mathbf{p} = (\mathbf{0}, 1)$, $\Omega_{\perp\mathbf{p}}$ consists of all quaternions $(\boldsymbol{\varrho}, 1)$ with $\boldsymbol{\varrho} \in \mathbb{R}^3$. The unit quaternion represented by $\boldsymbol{\varrho}$ is then given by $\mathbf{q}_{\boldsymbol{\varrho}} = (\boldsymbol{\varrho}, 1)/\sqrt{\boldsymbol{\varrho}^\intercal\boldsymbol{\varrho} + 1}$. For a general reference quaternion $\mathbf{p}$, the tangential hyperplane is given by $\Omega_{\perp\mathbf{p}} = \{\mathbf{B}\boldsymbol{\varrho} + \mathbf{p} \mid \boldsymbol{\varrho} \in \mathbb{R}^3\}$ for some basis $\mathbf{B} \in \mathbb{R}^{4\times3}$ of the tangent space, resulting in the unit quaternion $\mathbf{q}_{\boldsymbol{\varrho}} = (\mathbf{B}\boldsymbol{\varrho} + \mathbf{p})/\|\mathbf{B}\boldsymbol{\varrho} + \mathbf{p}\|$. This parametrization cannot describe unit quaternions $\mathbf{q}$ with $d_\angle(\mathbf{q}, \mathbf{p}) \geqslant \frac{\pi}{2}$.

Schmidt et al. [SN01] suggest to compute $\mathbf{q}_{\boldsymbol{\varrho}} = \sin(\theta)/\theta\,\mathbf{B}\boldsymbol{\varrho} + \cos(\theta)\mathbf{p}$ with $\theta = \|\mathbf{B}\boldsymbol{\varrho}\|$ instead, i. e., rotating $\mathbf{p}$ by angle θ towards $\mathbf{B}\boldsymbol{\varrho}$. This parametrization can describe all unit quaternions and preserves the angular distance between $\mathbf{q}$ and $\mathbf{p}$, i. e., $\|\boldsymbol{\varrho}\| = d_\angle(\mathbf{q}_{\boldsymbol{\varrho}}, \mathbf{p})$.

Terzakis et al. [Ter+12] use stereographic projection of a 3d equatorial hyperplane onto the 4d unit quaternion sphere for nonlinear optimization. Any unit quaternion $\mathbf{q} = (\boldsymbol{q}, q)$ with $q > -1$ can be described by its projection from the southpole onto the hyperplane through the equator of the unit sphere, yielding coordinates $\boldsymbol{\varrho} = \frac{\boldsymbol{q}}{q+1}$. The inverse mapping of an unconstrained minimal parameter vector $\boldsymbol{\varrho} \in \mathbb{R}^3$ to the respective unit quaternion is given by $\mathbf{q}_{\boldsymbol{\varrho}} = (2\boldsymbol{\varrho}, 1 - \boldsymbol{\varrho}^\intercal\boldsymbol{\varrho})/(\boldsymbol{\varrho}^\intercal\boldsymbol{\varrho} + 1)$. This method is computationally less complex than the approaches based on the tangential hyperplane and does not depend on a predefined reference point.

A.1.2 Dual Quaternions

Dual numbers are an extension of the real space similar to complex numbers. A dual number is defined as $\check{z} = a + \varepsilon b$ with $\varepsilon^2 = 0$ where a defines the real part and b the dual part of $\check{z}$. The space of dual numbers is referred to as $\mathbb{D}$.

Dual quaternions $\check{\mathbf{q}}$ are defined in a similar way to real quaternions as pairs $(\check{\boldsymbol{q}}, \check{q})$ where $\check{q} \in \mathbb{D}$ is a dual number and $\check{\boldsymbol{q}} \in \mathbb{D}^3$ is a dual vector. A dual quaternion can also be represented conveniently as $\check{\mathbf{q}} = \mathbf{q} + \varepsilon\mathbf{p}$ by a pair of real quaternions or 4-vectors $(\mathbf{q}, \mathbf{p})$ representing the real and dual part of $\check{\mathbf{q}}$, or as an 8-vector $\check{\mathbf{q}} = \left(\begin{smallmatrix}\mathbf{q}\\ \mathbf{p}\end{smallmatrix}\right)$.

Operations on dual quaternions are defined just as for real quaternions:

Addition:	$\check{\mathbf{q}} + \check{\mathbf{q}}'$	$= \mathbf{q} + \mathbf{q}' + \varepsilon(\mathbf{p} + \mathbf{p}')$
Scalar product:	$\lambda\check{\mathbf{q}}$	$= \lambda\mathbf{q} + \varepsilon\lambda\mathbf{p}$
Multiplication:	$\check{\mathbf{q}} \cdot \check{\mathbf{q}}'$	$= \mathbf{q} \cdot \mathbf{q}' + \varepsilon(\mathbf{q} \cdot \mathbf{p}' + \mathbf{p} \cdot \mathbf{q}')$
Conjugate:	$\bar{\check{\mathbf{q}}}$	$= \bar{\mathbf{q}} + \varepsilon\bar{\mathbf{p}}$
Magnitude:	$\lVert\check{\mathbf{q}}\rVert$	$= \sqrt{\mathbf{q}^\intercal\mathbf{q} + \varepsilon 2\mathbf{q}^\intercal\mathbf{p}} = \sqrt{\mathbf{q}^\intercal\mathbf{q}} + \varepsilon\frac{\mathbf{q}^\intercal\mathbf{p}}{\sqrt{\mathbf{q}^\intercal\mathbf{q}}}$
Inverse:	$\check{\mathbf{q}}^{-1}$	$= \mathbf{q}^{-1} - \varepsilon(\mathbf{q}^{-1} \cdot \mathbf{p} \cdot \mathbf{q}^{-1})$

Additional to the quaternion conjugate, there are specific conjugates based on the dual number conjugate $\check{z}^* = a - \varepsilon b$:

Dual conjugate:	$\check{\mathbf{q}}^*$	$= \mathbf{q} - \varepsilon\mathbf{p}$
Mixed conjugate:	$\bar{\check{\mathbf{q}}}^*$	$= \bar{\mathbf{q}} - \varepsilon\bar{\mathbf{p}}$

Similar to eq. (A.1) and eq. (A.2), multiplication by a dual quaternion $(\mathbf{q}, \mathbf{p})$ can be described in terms of matrix-vector-multiplication by matrices $\mathbf{M}^\ell_{\mathbf{q},\mathbf{p}}, \mathbf{M}^r_{\mathbf{q},\mathbf{p}} \in \mathbb{R}^{8\times 8}$ using the stacked vector representation $\check{\mathbf{q}} = \left(\begin{smallmatrix}\mathbf{q}\\ \mathbf{p}\end{smallmatrix}\right)$:

$$\mathbf{M}^\ell_{\mathbf{q},\mathbf{p}} = \begin{pmatrix} \mathbf{M}^\ell_{\mathbf{q}} & \mathbf{0}_{3\times 3} \\ \mathbf{M}^\ell_{\mathbf{p}} & \mathbf{M}^\ell_{\mathbf{q}} \end{pmatrix}, \quad \check{\mathbf{q}} \cdot \check{\mathbf{q}}' = \mathbf{M}^\ell_{\mathbf{q},\mathbf{p}} \begin{pmatrix} \mathbf{q}' \\ \mathbf{p}' \end{pmatrix} \tag{A.3}$$

and

$$\mathbf{M}^r_{\mathbf{q},\mathbf{p}} = \begin{pmatrix} \mathbf{M}^r_{\mathbf{q}} & \mathbf{0}_{3\times 3} \\ \mathbf{M}^r_{\mathbf{p}} & \mathbf{M}^r_{\mathbf{q}} \end{pmatrix}, \quad \check{\mathbf{q}}' \cdot \check{\mathbf{q}} = \mathbf{M}^r_{\mathbf{q},\mathbf{p}} \begin{pmatrix} \mathbf{q}' \\ \mathbf{p}' \end{pmatrix} \tag{A.4}$$

Note that the magnitude of a dual quaternion is a dual number with positive real part. For a unit dual quaternion, the real part of $\|\check{\mathbf{q}}\|$ is 1 and the dual part vanishes, i. e., the real part $\mathbf{q}$ of $\check{\mathbf{q}}$ is a unit quaternion and orthogonal to the dual part $\mathbf{p}$:

$$\|\check{\mathbf{q}}\| = 1 \quad \Leftrightarrow \quad \mathbf{q}^\mathsf{T}\mathbf{q} = 1 \text{ and } \mathbf{q}^\mathsf{T}\mathbf{p} = 0 \tag{A.5}$$

For unit dual quaternions, the multiplicative inverse $\check{\mathbf{q}}^{-1}$ is given by the conjugate quaternion $\bar{\check{\mathbf{q}}}$ due to the identity $\bar{\mathbf{q}} \cdot \mathbf{p} = -\bar{\mathbf{p}} \cdot \mathbf{q}$ for $\mathbf{q}^\mathsf{T}\mathbf{p} = 0$:

$$\begin{aligned} \bar{\mathbf{q}} \cdot \mathbf{p} &= (q\boldsymbol{p} - p\boldsymbol{q} - \boldsymbol{q} \times \boldsymbol{p},\ \underbrace{q\boldsymbol{p} + \boldsymbol{q}^\mathsf{T}\boldsymbol{p}}_{0}) \\ &= -(p\boldsymbol{q} - q\boldsymbol{p} - \boldsymbol{p} \times \boldsymbol{q},\ 0) = -\bar{\mathbf{p}} \cdot \mathbf{q} \end{aligned} \tag{A.6}$$

Unit dual quaternions relate to rigid motion (i. e., rotation and translation) of lines and points in 3d space as explained in Sec. 2.3.3.

A.2 Rotation Averaging

Given multiple rotations $\mathbf{R}_1, \ldots, \mathbf{R}_n \in \mathrm{SO}(3)$, the problem of *rotation averaging*, i. e., finding the L_p*-mean rotation* $\bar{\mathbf{R}} \in \mathrm{SO}(3)$ with respect to some metric $d : \mathrm{SO}(3) \times \mathrm{SO}(3) \to \mathbb{R}_{\geqslant 0}$ and an exponent $p \geqslant 1$, is defined as:

$$\bar{\mathbf{R}} = \underset{\mathbf{R} \in \mathrm{SO}(3)}{\operatorname{argmin}} \sum_{i=1}^{n} d(\mathbf{R}, \mathbf{R}_i)^p \tag{A.7}$$

The most commonly used algorithms to solve eq. (A.7) refer to the L_2-mean of the geodesic metric – the *Karcher mean* or *geometric mean* – or the L_2-mean of the quaternion metric [Dai+10]. The latter provides a simple solution in terms of unit quaternions:

$$\bar{\mathbf{q}} = \left(\sum_{i=1}^{n} \mathbf{q}_i \right) / \| \sum_{i=1}^{n} \mathbf{q}_i \| \tag{A.8}$$

Using the L_2-mean of the chordal metric yields a similar solution:

$$\bar{\mathbf{R}} = \text{orth}(\tfrac{1}{n}\sum_{i=1}^{n}\mathbf{R}_i) \tag{A.9}$$

where orth : $\mathbb{R}^{3\times3} \to \text{SO}(3)$ denotes an optimal orthonormalization strategy, e. g., via singular value decomposition.

However, the L_1-mean is known to be more robust than the L_2-mean, especially in the presence of outliers. The geodesic L_1-mean can be computed with a variant of the Weiszfeld algorithm [HAT11] or a Riemannian gradient descent algorithm with geodesic line search [Dai+10].

A.3 Absolute Orientation

A.3.1 Relative Pose Between Points

Given n corresponding 3d points $(\boldsymbol{X}_i, \boldsymbol{X}'_i)$, $i = 1, \ldots, n$ we want to find the rotation $\mathbf{R} \in \text{SO}(3)$ and translation vector $\boldsymbol{t} \in \mathbb{R}^3$ aligning the points optimally with respect to the Euclidean distance:

$$\min_{\mathbf{R},\boldsymbol{t}} \sum_{i=1}^{n} \|\boldsymbol{X}_i - \mathbf{R}\boldsymbol{X}'_i - \boldsymbol{t}\|^2 \quad \text{subject to } \mathbf{R} \in \text{SO}(3) \tag{A.10}$$

This problem can be reduced to finding the relative rotation $\mathbf{R}$ between two sets of vectors $(\boldsymbol{v}_i, \boldsymbol{v}'_i)$ between the 3d points and the centroids of the point sets, i. e., $\boldsymbol{v}_i = \boldsymbol{X}_i - \bar{\boldsymbol{X}}$, $\bar{\boldsymbol{X}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{X}_i$, and $\boldsymbol{v}'_i$, $\bar{\boldsymbol{X}}'$ are defined analogously.

Once $\mathbf{R}$ is found, the relative translation is given by $\boldsymbol{t} = \bar{\boldsymbol{X}} - \mathbf{R}\bar{\boldsymbol{X}}'$. The relative rotation can be estimated with the methods described in the following section.

A.3.2 Relative Rotation Between Vectors

Given n corresponding 3d vectors $(\boldsymbol{v}_i, \boldsymbol{v}'_i)$, $i = 1, \ldots, n$ we want to find the rotation $\mathbf{R} \in \mathrm{SO}(3)$ aligning the vectors optimally with respect to the Euclidean distance:

$$\min_{\mathbf{R}} \sum_{i=1}^{n} \|\boldsymbol{v}_i - \mathbf{R}\boldsymbol{v}'_i\|^2 \quad \text{subject to } \mathbf{R} \in \mathrm{SO}(3) \tag{A.11}$$

A closed-form solution for the rotation matrix $\mathbf{R}$ is described in [PM94]:

$$\mathbf{R} = \mathbf{U}\mathbf{V}^{-\frac{1}{2}}\mathbf{U}^{-1}\mathbf{A}^{\mathsf{T}} \tag{A.12}$$

where $\mathbf{A} = \sum_{i=1}^{n} \boldsymbol{v}_i \boldsymbol{v}_i'^{\mathsf{T}}$ and the eigendecomposition of $\mathbf{A}^{\mathsf{T}}\mathbf{A} = \mathbf{U}\mathbf{V}\mathbf{U}^{-1}$.

A closed-form solution based on unit quaternions for rotation representation is described by Horn w.r.t. the absolution orientation problem [Hor87]:

$$\max_{\mathbf{q} \in \mathbb{R}^4} \sum_{i=1}^{n} \mathbf{v}_i^{\mathsf{T}} (\mathbf{q} \cdot \mathbf{v}'_i \cdot \bar{\mathbf{q}}) \quad \text{subject to } \|\mathbf{q}\| = 1 \tag{A.13}$$

where $\mathbf{q} \cdot \mathbf{v}'_i \cdot \bar{\mathbf{q}}$ defines vector rotation in terms of quaternion multiplication as defined in eq. (2.26).[2] The inner scalar product can be written as:

$$\mathbf{v}_i^{\mathsf{T}} (\mathbf{q} \cdot \mathbf{v}'_i \cdot \bar{\mathbf{q}}) = (\mathbf{v}_i \cdot \mathbf{q})^{\mathsf{T}} (\mathbf{q} \cdot \mathbf{v}'_i) = (\mathbf{M}^{\ell}_{\mathbf{v}_i} \mathbf{q})^{\mathsf{T}} \mathbf{M}^{r}_{\mathbf{v}'_i} \mathbf{q} = \mathbf{q}^{\mathsf{T}} {\mathbf{M}^{\ell}_{\mathbf{v}_i}}^{\mathsf{T}} \mathbf{M}^{r}_{\mathbf{v}'_i} \mathbf{q}$$

in terms of the quaternion multiplication matrices defined in eq. (A.1) and (A.2). Hence, equation (A.13) can be rewritten as:

$$\max_{\mathbf{q} \in \mathbb{R}^4} \mathbf{q}^{\mathsf{T}} \mathbf{N} \mathbf{q} \quad \text{subject to } \|\mathbf{q}\| = 1 \tag{A.14}$$

with 4×4 matrix $\mathbf{N} = \sum_{i=1}^{n} {\mathbf{M}^{\ell}_{\mathbf{v}_i}}^{\mathsf{T}} \mathbf{M}^{r}_{\mathbf{v}'_i}$.

Equation (A.14) can be solved similar to the constrained linear least squares

[2] Note that 3d vectors are described by pure quaternions $\mathbf{v} = (v, 0)$ here.

problem (C.18) in C.3.2. The unit quaternion maximizing eq. (A.14) is given by the eigenvector corresponding to the largest eigenvalue of $\mathbf{M}$.

A very similar closed-form solution is proposed by Faugeras & Hébert [FH86] that minimizes the quaternion distance instead:

$$\min_{\mathbf{q}\in\mathbb{R}^4} \sum_{i=1}^{n} \|\mathbf{v}_i - \mathbf{q}\cdot\mathbf{v}_i'\cdot\bar{\mathbf{q}}\|^2 \quad \text{subject to } \|\mathbf{q}\| = 1 \tag{A.15}$$

where the inner part can be written in terms of the quaternion multiplication matrices as:

$$\|\mathbf{v}_i - \mathbf{q}\cdot\mathbf{v}_i'\cdot\bar{\mathbf{q}}\|^2 = \|\mathbf{v}_i\cdot\mathbf{q} - \mathbf{q}\cdot\mathbf{v}_i'\|^2 = \|(\mathbf{M}^{\ell}_{\mathbf{v}_i} - \mathbf{M}^{r}_{\mathbf{v}_i'})\mathbf{q}\|^2$$

Hence, equation (A.15) can be rewritten as:

$$\min_{\mathbf{q}\in\mathbb{R}^4} \mathbf{q}^{\mathsf{T}}\mathbf{M}\mathbf{q} \quad \text{subject to } \|\mathbf{q}\| = 1 \tag{A.16}$$

with 4×4 matrix $\mathbf{M} = \sum_{i=1}^{n} (\mathbf{M}^{\ell}_{\mathbf{v}_i} - \mathbf{M}^{r}_{\mathbf{v}_i'})^{\mathsf{T}}(\mathbf{M}^{\ell}_{\mathbf{v}_i} - \mathbf{M}^{r}_{\mathbf{v}_i'})$.

The solution of eq. (A.16) is equivalent to Horn's approach.

A.4 Distance Measures

Distances measures between points In general, distances between two n-dimensional points $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^n$ are measured either using the Euclidean distance or the Mahalanobis distance, depending on the availability of point covariance matrices.

The *Euclidean* or *geometric distance* between $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^n$ is defined by $d_{\text{geom}}(\boldsymbol{x}, \boldsymbol{x}') = \|\boldsymbol{x} - \boldsymbol{x}'\|$ with respect to the Euclidean norm $\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^{\mathsf{T}}\boldsymbol{x}}$.

The *Mahalanobis distance* (or "general interpoint distance") between $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^n$ assuming normally distributed measurement error for $\boldsymbol{x}$ with zero mean and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{x}} \in \mathbb{R}^{n\times n}$ is defined by $d_{\text{Maha}}(\boldsymbol{x}, \boldsymbol{x}') =$

$\sqrt{(\boldsymbol{x}-\boldsymbol{x}')^\intercal \boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1}(\boldsymbol{x}-\boldsymbol{x}')}$. Apparently, the Euclidean distance is a special case of the Mahalanobis distance with covariance matrix $\mathbf{I}_n$.

For measurements with internal constraints (e. g., points on a hyperplane or sphere), the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ is singular. In this case, the matrix inverse $\boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1}$ can be replaced by the pseudoinverse $\boldsymbol{\Sigma}_{\boldsymbol{x}}^{\dagger}$. This can be interpreted as measuring the distance between projections of $\boldsymbol{x}$ and $\boldsymbol{x}'$ onto the subspace perpendicular to the nullspace of $\boldsymbol{\Sigma}_{\boldsymbol{x}}$.

Distance measures to surfaces Next to interpoint distance we consider measures for distances from a point to a surface implicitly described by $f(\boldsymbol{x}) = 0$ for a function $f : \mathbb{R}^n \to \mathbb{R}$.

The *algebraic distance* $d_{\text{alg}}(\boldsymbol{x}; f) = f(\boldsymbol{x})$ is widely used due to its low complexity although in general it provides no geometrically reasponable value. The sign of the distance measure specifies if $\boldsymbol{x}$ lies "inside" or "outside" of the surface.

The *geometric distance* $d_{\text{geom}}(\boldsymbol{x}; f) = \min\{d_{\text{geom}}(\boldsymbol{x}, \boldsymbol{x}') \mid \boldsymbol{x}' \in \mathbb{R}^n, f(\boldsymbol{x}') = 0\}$ provides the geometrically most reasonable measure but is often difficult to compute.

By replacing the Euclidean interpoint distance d_{geom} by the Mahalanobis distance d_{Maha} we obtain the Mahalanobis distance $d_{\text{Maha}}(\boldsymbol{x}; f)$ between an uncertain point $\boldsymbol{x}$ and a surface described by f.

The *Sampson error* $d_{\text{Samp}}(\boldsymbol{x}; f) = \frac{f(\boldsymbol{x})}{\|\nabla f(\boldsymbol{x})\|}$ is an approximation of $d_{\text{geom}}(\boldsymbol{x}; f)$. It is defined as the geometric distance between $\boldsymbol{x}$ and the hyperplane described by the first order approximation[3] of the function f. A similar approximation can be derived for the Mahalanobis distance.

Spherical distance measures For unit vectors $\boldsymbol{v}, \boldsymbol{v}' \in \mathbb{S}^2$, distance measures relating to the inter-vector angle can be considered instead.

[3] Note that the Sampson error is *not* the first order approximation of the geometric distance from $\boldsymbol{x}$ to the surface described by f as often assumed [HO06].

The *geodesic* or *angle distance* is defined by $d_{\angle}(\boldsymbol{v}, \boldsymbol{v}') = \arccos(\boldsymbol{v}^\top \boldsymbol{v}')$ or equivalently by $d_{\angle}(\boldsymbol{v}, \boldsymbol{v}') = \arcsin(\|\boldsymbol{v} \times \boldsymbol{v}'\|)$.[4] This metric yields the absolute value α of the actual angle between $\boldsymbol{v}$ and $\boldsymbol{v}'$.

To avoid trigonometric functions, the angle distance can be replaced by $d_{\cos}(\boldsymbol{v}, \boldsymbol{v}') = 1 - \boldsymbol{v}^\top \boldsymbol{v}' = 1 - \cos(\alpha)$, $d_{\sin}(\boldsymbol{v}, \boldsymbol{v}') = \|\boldsymbol{v} \times \boldsymbol{v}'\| = \sin(\alpha)$, or by the Euclidean distance $d_{\text{geom}}(\boldsymbol{v}, \boldsymbol{v}') = \|\boldsymbol{v} - \boldsymbol{v}'\|$ which is related to $d_{\angle}$ by $d_{\text{geom}}(\boldsymbol{v}, \boldsymbol{v}') = \sqrt{2 - 2\cos(\alpha)} = 2\sin(\alpha/2) \approx \alpha$ for unit vectors $\boldsymbol{v}$, $\boldsymbol{v}'$.

A.4.1 Distance Measures for Essential Matrix Estimation

Common error measures for essential matrix estimation from normalized 2d/2d image correspondences are:

- the *algebraic distance* $d_{\text{alg}}(\boldsymbol{x}, \boldsymbol{x}'; \mathbf{E}) = \boldsymbol{x}^\top \mathbf{E} \boldsymbol{x}'$
- the *geometric epipolar line distance* $d_{\text{geom}}(\boldsymbol{x}, \boldsymbol{x}'; \mathbf{E}) = \frac{\boldsymbol{x}^\top \mathbf{E} \boldsymbol{x}'}{\|\mathbf{E}_{[1\ldots2]} \boldsymbol{x}'\|}$
  i. e., the signed distance between $\boldsymbol{x}$ and the epipolar line with respect to $\boldsymbol{x}'$ within the normalized image plane
- the *Sampson error* $d_{\text{Samp}}(\boldsymbol{x}, \boldsymbol{x}'; \mathbf{E}) = \frac{\boldsymbol{x}^\top \mathbf{E} \boldsymbol{x}'}{\sqrt{(\mathbf{E}\boldsymbol{x}')_1^2 + (\mathbf{E}\boldsymbol{x}')_2^2 + (\mathbf{E}^\top \boldsymbol{x})_1^2 + (\mathbf{E}^\top \boldsymbol{x})_2^2}}$
- the *spherical distance* $d_{\text{sph}}(\boldsymbol{v}, \boldsymbol{v}'; \mathbf{E}) = \frac{\boldsymbol{v}^\top \mathbf{E} \boldsymbol{v}'}{\|\mathbf{E}\boldsymbol{v}'\|}$
  for the spherical case, i. e., the signed distance between $\boldsymbol{v}$ and the epipolar plane with respect to $\boldsymbol{v}'$
- the *angular distance* $d_{\text{ang}}(\boldsymbol{v}, \boldsymbol{v}'; \mathbf{E}) = \arcsin(d_{\text{sph}}(\boldsymbol{v}, \boldsymbol{v}'; \mathbf{E}))$
  for the spherical case, i. e., the signed angle between $\boldsymbol{v}$ and the epipolar plane with respect to $\boldsymbol{v}'$

The *symmetric distance* w.r.t. to a certain error measure is given by

$$d^2_{\text{sym}}(\boldsymbol{x}, \boldsymbol{x}'; \mathbf{E}) = d^2(\boldsymbol{x}, \boldsymbol{x}'; \mathbf{E}) + d^2(\boldsymbol{x}', \boldsymbol{x}; \mathbf{E}^\top)$$

[4] In practice, the angle distance is often calculated as $d_{\angle}(\boldsymbol{v}, \boldsymbol{v}') = \text{atan2}(\|\boldsymbol{v} \times \boldsymbol{v}'\|, \boldsymbol{v}^\top \boldsymbol{v}')$ combining both definitions which is numerically more stable.

A.4.2 Distance Measures for Absolute Pose Estimation

Common error measures for absolute pose estimation from 2d/3d point correspondences are:

- ▻ the *orthographic distance* $d_{\text{orth}}(\boldsymbol{x}, \boldsymbol{X}) = \|X_z \boldsymbol{x} - \boldsymbol{X}\|$ for the planar case
- ▻ the *image reprojection error* $d_{\mathcal{K}}(\boldsymbol{u}, \boldsymbol{X}) = \|\boldsymbol{u} - \mathcal{K}(\boldsymbol{X})\|$ for the general case, i. e., the reprojection error within the actual image in pixels
- ▻ the *normalized reprojection error* $d_{\mathcal{P}}(\boldsymbol{x}, \boldsymbol{X}) = \|\boldsymbol{x} - \mathcal{P}(\boldsymbol{X})\|$ for the planar case, i. e., the geometric distance between $\boldsymbol{x}$ and the projection of $\boldsymbol{X}$ within the normalized image plane
- ▻ the *spherical reprojection error* $d_{\mathcal{S}}(\boldsymbol{v}, \boldsymbol{X}) = \|\boldsymbol{v} - \mathcal{S}(\boldsymbol{X})\|$ for the spherical case, i. e., the geometric distance between $\boldsymbol{v}$ and the direction vector towards $\boldsymbol{X}$
- ▻ the *angular distance* $d_{\text{ang}}(\boldsymbol{v}, \boldsymbol{X}) = \arccos(\boldsymbol{v}^\intercal \mathcal{S}(\boldsymbol{X}))$ for the spherical case, i. e., the angle between $\boldsymbol{v}$ and the direction vector towards $\boldsymbol{X}$
- ▻ the *projection ray distance* $d_{\text{ray}}(\boldsymbol{v}, \boldsymbol{X}) = \|\boldsymbol{v} \times \boldsymbol{X}\|$ for the spherical case, i. e., the geometric distance between $\boldsymbol{X}$ and the line through the origin with direction $\boldsymbol{v}$

A.5 Uncertainty Handling

Since geometric entities estimated from images are inherently uncertain, we have to consider statistical properties of these entities in estimation processes.

Assuming that a parameter vector $\boldsymbol{x} \in \mathbb{R}^n$ follows a Gaussian distribution[5], its uncertainty is represented by the second moments of the probability

[5] Note that this is often not the case for parameters describing geometric entities such as 3d points created via triangulation or pose parameters estimated from 2d/3d correspondences. However, this assumption is commonly considered as an approximation to the actual probability distribution for practical reasons.

density function, i. e., the *covariance matrix* $\boldsymbol{\Sigma}_{\boldsymbol{x}} \in \mathbb{R}^{n \times n}$ of the form:

$$\boldsymbol{\Sigma}_{\boldsymbol{x}} = \begin{pmatrix} \sigma_1^2 & \sigma_{1,2} & \dots & \sigma_{1,n} \\ \sigma_{2,1} & \sigma_2^2 & \dots & \sigma_{2,n} \\ \vdots & & \ddots & \vdots \\ \sigma_{n,1} & \sigma_{n,2} & \dots & \sigma_n^2 \end{pmatrix} \tag{A.17}$$

where $\sigma_{i,j} = \mathrm{cov}(x_i, x_j)$ is the covariance between the i-th and j-th parameter. The main diagonal entries of $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ contain the variances $\sigma_i^2 = \mathrm{var}(x_i)$ for each parameter entry. For statistically independent parameters, the covariance matrix has diagonal form $\boldsymbol{\Sigma}_{\boldsymbol{x}} = \mathrm{diag}(\sigma_1^2, \dots, \sigma_n^2)$. For equally distributed uncorrelated parameters the covariance matrix is $\sigma^2 \mathbf{I}_n$.

Propagation of uncertainty with respect to a linear transformation $\boldsymbol{y} = \mathbf{A}\boldsymbol{x}$ with $\mathbf{A} \in \mathbb{R}^{m \times n}$ is described by:

$$\boldsymbol{\Sigma}_{\boldsymbol{y}} = \mathbf{A}\boldsymbol{\Sigma}_{\boldsymbol{x}}\mathbf{A}^\mathsf{T} \tag{A.18}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{y}} \in \mathbb{R}^{m \times m}$ is the covariance matrix of the transformed parameter vector $\boldsymbol{y} \in \mathbb{R}^m$.

The uncertainty of nonlinear functions of $\boldsymbol{x}$ is approximated via linearization. Given a function $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$, the covariance matrix of $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$ is approximated as:

$$\boldsymbol{\Sigma}_{\boldsymbol{y}} = \mathbf{J}_{\boldsymbol{f}} \boldsymbol{\Sigma}_{\boldsymbol{x}} \mathbf{J}_{\boldsymbol{f}}^\mathsf{T} \tag{A.19}$$

where $\mathbf{J}_{\boldsymbol{f}}$ is the Jacobian matrix of $\boldsymbol{f}$ evaluated at $\boldsymbol{x}$.

A.5.1 Error Propagation for Common Functions

In the following we consider propagation of uncertainty for common transformations of parameter vectors representing geometric entities such as homogeneous 3d points, quaternions, or direction vectors. For a comprehensive treatment of this topic we refer the reader to Meidow et al. [MBF09].

Vector products Uncertainty of the scalar product $\boldsymbol{x}^\intercal \boldsymbol{y}$ of uncorrelated vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ with covariance matrices $\boldsymbol{\Sigma}_x, \boldsymbol{\Sigma}_y \in \mathbb{R}^{n \times n}$ is given by:

$$\sigma^2 = \begin{pmatrix} \boldsymbol{y}^\intercal & \boldsymbol{x}^\intercal \end{pmatrix} \begin{pmatrix} \boldsymbol{\Sigma}_x & \mathbf{0}_{n \times n} \\ \mathbf{0}_{n \times n} & \boldsymbol{\Sigma}_y \end{pmatrix} \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{x} \end{pmatrix} = \boldsymbol{y}^\intercal \boldsymbol{\Sigma}_x \boldsymbol{y} + \boldsymbol{x}^\intercal \boldsymbol{\Sigma}_y \boldsymbol{x} \tag{A.20}$$

For $n = 3$, the uncertainty of the cross product $\boldsymbol{z} = \boldsymbol{x} \times \boldsymbol{y}$ is given by:

$$\boldsymbol{\Sigma}_z = [\boldsymbol{y}]_\times^\intercal \boldsymbol{\Sigma}_x [\boldsymbol{y}]_\times + [\boldsymbol{x}]_\times \boldsymbol{\Sigma}_y [\boldsymbol{x}]_\times^\intercal \tag{A.21}$$

Normalization Normalization of a vector $\boldsymbol{x} \in \mathbb{R}^n$ is described by the function $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n, \boldsymbol{x} \mapsto \boldsymbol{x}/\|\boldsymbol{x}\|$. The Jacobian of $\boldsymbol{f}$ evaluated at $\boldsymbol{x}$ is:

$$\mathbf{J}_f = \frac{1}{\|\boldsymbol{x}\|} \left(\mathbf{I}_n - \frac{\boldsymbol{x}\boldsymbol{x}^\intercal}{\boldsymbol{x}^\intercal \boldsymbol{x}} \right) \tag{A.22}$$

Note that the resulting covariance matrix $\boldsymbol{\Sigma}_y = \mathbf{J}_f \boldsymbol{\Sigma}_x \mathbf{J}_f^\intercal$ is singular with rank $n-1$. Descriptively, uncertainty is only present within the hyperplane tangent to the normalized vector $\boldsymbol{y}$ while uncertainty along its direction is zero.

Projection Orthogonal projection of a vector $\boldsymbol{x} \in \mathbb{R}^n$ onto the hyperplane with normal vector $\boldsymbol{n} \in \mathbb{R}^n$, $\|\boldsymbol{n}\| = 1$, is described by the linear function $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n, \boldsymbol{x} \mapsto \boldsymbol{x} - (\boldsymbol{n}^\intercal \boldsymbol{x})\boldsymbol{n}$. The Jacobian of $\boldsymbol{f}$ is:

$$\mathbf{J}_f = \mathbf{I}_n - \boldsymbol{n}\boldsymbol{n}^\intercal \tag{A.23}$$

The resulting covariance matrix is again singular with rank $n-1$. In this case, uncertainty is only present within the projection plane while uncertainty along $\boldsymbol{n}$ is zero.

Projection with respect to an uncertain normal vector $\boldsymbol{n}$ with covariance matrix $\boldsymbol{\Sigma}_n$ yields:

$$\mathbf{J}_f = \begin{pmatrix} (\mathbf{I}_n - \boldsymbol{n}\boldsymbol{n}^\intercal) & -(\boldsymbol{n}\boldsymbol{x}^\intercal + \boldsymbol{n}^\intercal \boldsymbol{x}\, \mathbf{I}_n) \end{pmatrix} \tag{A.24}$$

Homogenization Homogenization of $\boldsymbol{x} \in \mathbb{R}^n$ is described by the function $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n, \boldsymbol{x} \mapsto \boldsymbol{x}/x_n$. The Jacobian of $\boldsymbol{f}$ evaluated at $\boldsymbol{x}$ is:

$$\mathbf{J}_f = \begin{pmatrix} \frac{1}{x_n}\mathbf{I}_{n-1} & -\frac{1}{x_n^2}\boldsymbol{x}_{[1\ldots n-1]} \\ \mathbf{0}^\intercal & 0 \end{pmatrix} \tag{A.25}$$

The resulting covariance matrix has rank $n-1$ with zero uncertainty in the n-th entry of the homogenized vector $\boldsymbol{y}$.

Dual quaternion normalization Given a dual quaternion $(\mathbf{q}, \mathbf{p}) \in \mathbb{R}^8$, the unit length constaint is enforced by scaling w.r.t. $\|\mathbf{q}\| = 1$ and orthogonal projection of $\mathbf{p}$ onto $\mathbf{q}$:

$$f(\mathbf{q}, \mathbf{p}) = \frac{1}{\|\mathbf{q}\|}\left(\mathbf{q}, \mathbf{p} - \frac{\mathbf{q}^\intercal\mathbf{p}}{\mathbf{q}^\intercal\mathbf{q}}\mathbf{q}\right)$$

which can be written as a composite function $f(\mathbf{q}, \mathbf{p}) = g(h(\mathbf{q}, \mathbf{p}))$ with $g(\mathbf{q}, \mathbf{p}) = (\mathbf{q}, \mathbf{p} - (\mathbf{q}^\intercal\mathbf{p})\mathbf{q})$ and $h(\mathbf{q}, \mathbf{p}) = (\mathbf{q}/\|\mathbf{q}\|, \mathbf{p}/\|\mathbf{q}\|)$.

Since $f = g \circ h$, the Jacobian of f evaluated at $(\mathbf{q}, \mathbf{p})$ is given according to the chain rule by $\mathbf{J}_f = \mathbf{J}_g\mathbf{J}_h$ where $\mathbf{J}_g$ is the Jacobian matrix of g evaluated at $h(\mathbf{q}, \mathbf{p})$ and $\mathbf{J}_h$ is the Jacobian matrix of h evaluated at $(\mathbf{q}, \mathbf{p})$:

$$\mathbf{J}_g = \begin{pmatrix} \mathbf{I}_4 & \mathbf{0}_{4\times4} \\ -\frac{\mathbf{q}^\intercal\mathbf{p}}{\mathbf{q}^\intercal\mathbf{q}}\mathbf{I}_4 - \frac{\mathbf{q}\mathbf{p}^\intercal}{\mathbf{q}^\intercal\mathbf{q}} & \mathbf{I}_4 - \frac{\mathbf{q}\mathbf{q}^\intercal}{\mathbf{q}^\intercal\mathbf{q}} \end{pmatrix}$$

and

$$\mathbf{J}_h = \frac{1}{\|\mathbf{q}\|}\begin{pmatrix} \mathbf{I}_4 - \frac{\mathbf{q}\mathbf{q}^\intercal}{\mathbf{q}^\intercal\mathbf{q}} & \mathbf{0}_{4\times4} \\ -\frac{\mathbf{p}\mathbf{q}^\intercal}{\mathbf{q}^\intercal\mathbf{q}} & \mathbf{I}_4 \end{pmatrix}$$

The Jacobian matrix of f can be written in closed form as:

$$\mathbf{J}_f = \frac{1}{\|\mathbf{q}\|}\begin{pmatrix} \mathbf{I}_4 - \frac{\mathbf{q}\mathbf{q}^\intercal}{\mathbf{q}^\intercal\mathbf{q}} & \mathbf{0}_{4\times4} \\ -\frac{1}{\mathbf{q}^\intercal\mathbf{q}}\mathbf{A} & \mathbf{I}_4 - \frac{\mathbf{q}\mathbf{q}^\intercal}{\mathbf{q}^\intercal\mathbf{q}} \end{pmatrix} \tag{A.26}$$

with

$$\mathbf{A} = \mathbf{p}\mathbf{q}^\intercal + \mathbf{q}\mathbf{p}^\intercal + (\mathbf{q}^\intercal\mathbf{p})\left(\mathbf{I}_4 - \frac{3\mathbf{q}\mathbf{q}^\intercal}{\mathbf{q}^\intercal\mathbf{q}}\right)$$

The resulting 8×8 covariance matrix is singular with rank 6. Uncertainty of both the real and dual part of the normalized dual quaternion along the direction of the real part is zero.

Quaternion rotation Given a unit quaternion $\mathbf{q} \in \mathbb{R}^4$ with covariance matrix $\mathbf{\Sigma_q} \in \mathbb{R}^{4\times4}$ and a 3d point $\boldsymbol{X} \in \mathbb{R}^3$ with covariance matrix $\mathbf{\Sigma}_X \in \mathbb{R}^{3\times3}$, the uncertainty for the rotated[6] point $\boldsymbol{X}' = \mathbf{R_q}\boldsymbol{X}$ is approximated via uncertainty propagation as:

$$\mathbf{\Sigma}_{X'} = \begin{pmatrix}\mathbf{J}_f & \mathbf{R_q}\end{pmatrix}\begin{pmatrix}\mathbf{\Sigma_q} & \mathbf{0}_{4\times3} \\ \mathbf{0}_{3\times4} & \mathbf{\Sigma}_X\end{pmatrix}\begin{pmatrix}\mathbf{J}_f^\intercal \\ \mathbf{R_q}^\intercal\end{pmatrix} = \mathbf{J}_f\mathbf{\Sigma_q}\mathbf{J}_f^\intercal + \mathbf{R_q}\mathbf{\Sigma}_X\mathbf{R_q}^\intercal \tag{A.27}$$

where $\mathbf{J}_f$ is the Jacobian matrix of the function $f(\mathbf{q}) = \mathbf{R_q}\boldsymbol{X}$ evaluated at $\mathbf{q}$. The Jacobian matrix can be derived from eq. (2.30) as:

$$\mathbf{J}_f = 2\begin{pmatrix}\boldsymbol{q}\boldsymbol{X}^\intercal - \boldsymbol{X}\boldsymbol{q}^\intercal - (q\mathbf{I} + [\boldsymbol{q}]_\times)[\boldsymbol{X}]_\times & (q\mathbf{I} + [\boldsymbol{q}]_\times)\boldsymbol{X}\end{pmatrix} \tag{A.28}$$

Given an additional translation vector $\boldsymbol{t}$ with covariance matrix $\mathbf{\Sigma}_t \in \mathbb{R}^{3\times3}$, the approximate uncertainty of the transformed point $\boldsymbol{X}' = \mathbf{R_q}\boldsymbol{X} + \boldsymbol{t}$ is:

$$\mathbf{\Sigma}_{X'} = \mathbf{J}_f\mathbf{\Sigma_q}\mathbf{J}_f^\intercal + \mathbf{R_q}\mathbf{\Sigma}_X\mathbf{R_q}^\intercal + \mathbf{\Sigma}_t \tag{A.29}$$

Rigid motion composition Composing two rigid motions described by unit quaternions $\mathbf{q}_1, \mathbf{q}_2 \in \mathbb{R}^4$ and translation vectors $\boldsymbol{t}_1, \boldsymbol{t}_2 \in \mathbb{R}^3$ results in $\mathbf{q} = \mathbf{q}_1 \cdot \mathbf{q}_2$ and $\boldsymbol{t} = \mathbf{R}_{\mathbf{q}_1}\boldsymbol{t}_2 + \boldsymbol{t}_1$. Given covariance matrices $\mathbf{\Sigma}_{\mathbf{q}_1}, \mathbf{\Sigma}_{\mathbf{q}_2} \in \mathbb{R}^{4\times4}$ and $\mathbf{\Sigma}_{t_1}, \mathbf{\Sigma}_{t_2} \in \mathbb{R}^{3\times3}$, the covariance matrix of the composed rigid motion

[6] and possibly scaled if the unit length constraint of $\mathbf{q}$ is dropped

is approximated as:

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathbf{q}} &= \begin{pmatrix}\mathbf{M}^{r}_{\mathbf{q}_2} & \mathbf{M}^{\ell}_{\mathbf{q}_1}\end{pmatrix}\begin{pmatrix}\boldsymbol{\Sigma}_{\mathbf{q}_1} & \mathbf{0}_{4\times4}\\ \mathbf{0}_{4\times4} & \boldsymbol{\Sigma}_{\mathbf{q}_2}\end{pmatrix}\begin{pmatrix}\mathbf{M}^{r}_{\mathbf{q}_2}{}^{\mathsf{T}}\\ \mathbf{M}^{\ell}_{\mathbf{q}_1}{}^{\mathsf{T}}\end{pmatrix}\\ &= \mathbf{M}^{r}_{\mathbf{q}_2}\boldsymbol{\Sigma}_{\mathbf{q}_1}\mathbf{M}^{r}_{\mathbf{q}_2}{}^{\mathsf{T}} + \mathbf{M}^{\ell}_{\mathbf{q}_1}\boldsymbol{\Sigma}_{\mathbf{q}_2}\mathbf{M}^{\ell}_{\mathbf{q}_1}{}^{\mathsf{T}}\end{aligned} \tag{A.30}$$

where $\mathbf{M}^{\ell}_{\mathbf{q}}, \mathbf{M}^{r}_{\mathbf{q}}$ are the left and right quaternion multiplication matrices from eq. (A.1) and (A.2), and:

$$\boldsymbol{\Sigma}_{\boldsymbol{t}} = \mathbf{J}_f\boldsymbol{\Sigma}_{\mathbf{q}_1}\mathbf{J}_f^{\mathsf{T}} + \mathbf{R}_{\mathbf{q}_1}\boldsymbol{\Sigma}_{\boldsymbol{t}_2}\mathbf{R}_{\mathbf{q}_1}^{\mathsf{T}} + \boldsymbol{\Sigma}_{\boldsymbol{t}_1} \tag{A.31}$$

where $\mathbf{J}_f$ is the matrix defined in eq. (A.28) with $\mathbf{q}, \boldsymbol{X}$ replaced by $\mathbf{q}_1, \boldsymbol{t}_2$.

Conversion from angle/axis to unit quaternion Given a rotation angle $\alpha \in [0, 2\pi]$ with error $\alpha_\varepsilon \sim \mathcal{N}(0, \sigma^2_{\alpha_\varepsilon})$ and rotation axis $\boldsymbol{r} \in \mathbb{S}^2$ with error $\boldsymbol{r}_\varepsilon \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{r}_\varepsilon})$, the corresponding error for the unit quaternion parametrization can be derived from $\mathbf{q} = \boldsymbol{f}(\boldsymbol{r}, \alpha) = (\sin(\frac{\alpha}{2})\boldsymbol{r}^{\mathsf{T}}, \cos(\frac{\alpha}{2}))^{\mathsf{T}}$:

$$\mathbf{J}_f = \begin{pmatrix}\sin(\frac{\alpha}{2})\mathbf{I} & \frac{1}{2}\cos(\frac{\alpha}{2})\boldsymbol{r}\\ \mathbf{0}_{3\times1} & -\frac{1}{2}\sin(\frac{\alpha}{2})\end{pmatrix} \tag{A.32}$$

and

$$\boldsymbol{\Sigma}_{\mathbf{q}_\varepsilon} = \mathbf{J}_f\begin{pmatrix}\boldsymbol{\Sigma}_{\boldsymbol{r}_\varepsilon} & \mathbf{0}_{3\times1}\\ \mathbf{0}_{1\times3} & \sigma^2_{\alpha_\varepsilon}\end{pmatrix}\mathbf{J}_f^{\mathsf{T}}$$

Note that $\boldsymbol{\Sigma}_{\boldsymbol{r}_\varepsilon}$ is singular with rank 2 due to the unit length constraint. The resulting covariance matrix $\boldsymbol{\Sigma}_{\mathbf{q}_\varepsilon}$ is also singular with rank 3. The rotation axis $\boldsymbol{r}$ lies in the nullspace of both matrices.

Appendix B

Structure from Motion

B.1 Spherical Camera Model

Due to the perspective projection, the pinhole camera model is limited to cameras with a field of view of below 180°. In order to model cameras with a larger field of view – such as omnidirectional cameras or fisheye lens cameras – the normalized image plane can be replaced by the unit sphere, i. e., 3d points are not projected onto the $z = 1$ plane in the camera coordinate frame but onto the unit sphere around the camera center. There are different mathematical models describing the actual camera function for spherical cameras such as the omnidirectional camera model proposed by Scaramuzza et al. [SMS06] where the camera function is composed of a mapping from 3d point $\boldsymbol{X}$ to spherical coordinates $(\Phi = \text{atan2}(Y, X), \Theta = \arccos(Z/\|\boldsymbol{X}\|))$ and a mapping from the 2d point $(\Theta\cos(\Phi), \Theta\sin(\Phi))$ to image coordinates using a camera matrix $\mathbf{K}$ as defined in eq. (4.2):

$$\boldsymbol{u} = \mathcal{K}(\boldsymbol{X}) = \begin{pmatrix} f & 0 & p_u \\ 0 & f & p_v \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \Theta\cos(\Phi) \\ \Theta\sin(\Phi) \\ 1 \end{pmatrix} \tag{B.1}$$

The calibrated *spherical camera function* is described by $\mathcal{S} : \mathbb{R}^3 \to \mathbb{S}^2, \boldsymbol{X} \mapsto \boldsymbol{v}$ where $\mathcal{S}(\boldsymbol{X}) = \boldsymbol{X}/\|\boldsymbol{X}\|$ scales the vector between camera center and 3d point to unit length. We also define the *spherical unprojection function* $\mathcal{U}_\mathcal{S} : \mathbb{P}^2 \to \mathbb{S}^2$ mapping from image pixels $\boldsymbol{u}$ to the corresponding point on the unit sphere $\boldsymbol{v} = \mathcal{S}(\boldsymbol{X})$, which is the unit direction vector of the ray

from camera center to projected 3d point $\boldsymbol{X}$ within the camera coordinate frame.

In the Structure from Motion problem and related problems described in Sec. 4.5, normalized 2d points $\boldsymbol{x}$ are used as input that are derived from actual image positions $\boldsymbol{u}$ via the unprojection function $\boldsymbol{x} = \mathcal{U}_{\mathcal{P}}(\boldsymbol{u})$ resp. $\mathbf{K}^{-1}\boldsymbol{u}$ for an ideal pinhole camera. This transformation might not be feasible for cameras that cannot be described properly using the planar camera model but by the spherical camera model instead.

Hence, "spherical" versions of these problems based on the spherical camera model are defined by replacing normalized 2d points $\boldsymbol{x} = \mathcal{U}_{\mathcal{P}}(\boldsymbol{u})$ by direction vectors $\boldsymbol{v} = \mathcal{U}_{\mathcal{S}}(\boldsymbol{u})$, planar projection $\mathcal{P}$ by spherical projection $\mathcal{S}$, and using an appropriate error metric d on $\mathbb{S}^2$ (see A.4). The same modification can be applied to bundle adjustment (see B.5).

B.2 Relative Pose Estimation

Given two images of a camera that have been captured at different locations, the relative pose between these locations can be obtained up to scale from a number of corresponding 2d image points between both images. For the case of a calibrated planr camera, the relative pose problem is in general solved via computation of the *essential matrix*[1] relating corresponding normalized image points $\boldsymbol{x}, \boldsymbol{x}'$ in two views via the *epipolar constraints*.

B.2.1 The Essential Matrix

We identify the first camera pose w.l.o.g. with the canonical pose $[\mathbf{I} \mid \mathbf{0}]$ and the second pose by $[\mathbf{R} \mid \boldsymbol{t}]$. Given are image coordinates $\boldsymbol{x}$ and $\boldsymbol{x}'$ of a 3d point $\boldsymbol{X}$ in both views, i. e., $\boldsymbol{x} = \mathcal{P}(\boldsymbol{X})$ and $\boldsymbol{x}' = \mathcal{P}(\mathbf{R}^\intercal(\boldsymbol{X} - \boldsymbol{t}))$. The essential matrix $\mathbf{E}$ is a non-zero singular 3×3 matrix relating $\boldsymbol{x}$ and $\boldsymbol{x}'$

[1] originally proposed by Hugh Christopher Longuet-Higgins in [Lon81]

according to the *epipolar constraint*:

$$\boldsymbol{x}^{\mathsf{T}}\mathbf{E}\boldsymbol{x}' = 0 \tag{B.2}$$

This relation is also known as the *Longuet-Higgins equation* [Lon81].

Although originally developed with respect to planar cameras, the notion of the essential matrix with respect to eq. (B.2) is also valid for the spherical camera model where normalized image points are identified with direction vectors $\boldsymbol{v} = \mathcal{S}(\boldsymbol{X})$ in the local camera coordinate frame, since the scale of the corresponding vectors is arbitrary. Nevertheless, different error measures must be considered for the estimation of the essential matrix from spherical points.

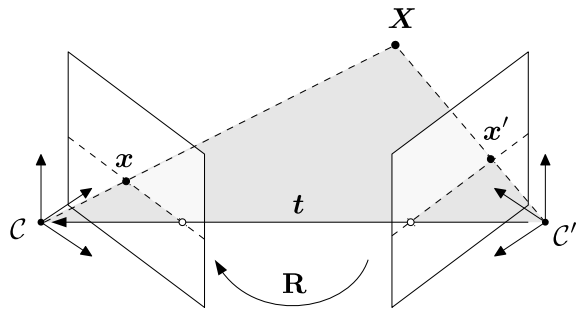

Figure B.1. Illustration of the epipolar geometry for a 2d point correspondence between two views for the classical case described by Longuet-Higgins [Lon81].

The epipolar constraint is derived from the observation that the vectors between the camera centers and the 3d point $\boldsymbol{X}$ are coplanar with the translation vector $\boldsymbol{t}$ between the camera centers in as depicted in Fig. B.1, forming the *epipolar plane*. For the planar camera model, the intersection of the epipolar plane with the image plane yields a line – denoted as the *epipolar line* with respect to a 2d point $\boldsymbol{x}$ – that contains all possible corresponding 2d points in the second image. These geometric properties of stereo views constitute the *epipolar geometry*.

Since $\boldsymbol{t}$ and $\mathbf{R}\boldsymbol{x}'$ lie on the epipolar plane, its normal is described by $\boldsymbol{n} \sim \boldsymbol{t} \times \mathbf{R}\boldsymbol{x}'$. Hence, the coplanarity constraint can be formalized as:

$$\boldsymbol{x}^\mathsf{T}(\boldsymbol{t} \times \mathbf{R}\boldsymbol{x}') = 0 \tag{B.3}$$

Using the matrix notation $[\boldsymbol{t}]_\times$ for the cross product with $\boldsymbol{t}$, the essential matrix in eq. (B.2) is related to the relative pose up to scale by:[2]

$$\mathbf{E} \sim [\boldsymbol{t}]_\times \mathbf{R} \tag{B.4}$$

Note that the essential matrix is only defined when the translation is non-zero, hence the pure rotation case will not be considered here. Since the essential matrix is defined up to scale only, there are several ambiguities for the decomposition of $\mathbf{E}$ into $\mathbf{R}$ and $\boldsymbol{t}$. Most importantly, the absolute scale of the translation vector $\boldsymbol{t}$ cannot be recovered without further knowledge about the scene. We will return to this topic later in B.2.3.

B.2.2 Estimation of the Essential Matrix

In the following we will discuss how the essential matrix can be computed from eq. (B.2) given multiple normalized 2d point correspondences $(\boldsymbol{x}_j, \boldsymbol{x}'_j)$, $j = 1, \ldots, N$ between two views.

The description of the essential matrix $\mathbf{E}$ in eq. (B.4) reveals that the essential matrix has only five degrees of freedom: Both rotation and translation have three degrees of freedom but since $\mathbf{E}$ is only defined up to scale, the degrees of freedom for the translation are reduced to two. These internal constraints are satisfied if and only if two of the singular values of $\mathbf{E}$ are equal and the third is zero (see Sec. 9.6.1 in [HZ04]).[3] These

[2] Different definitions of the essential matrix in terms of the relative pose are found in the literature, depending on the definition of the transformation between the camera coordinate frames and the position of $\boldsymbol{x}$ and $\boldsymbol{x}'$ in the epipolar equation. In this work we will keep to the classical formulation by Longuet-Higgins.

[3] Note that a non-zero 3×3 matrix $\mathbf{A}$ is skew-symmetric if and only if two of its singular values are equal and non-zero and the third is zero. The multiplication of $\mathbf{A}$ with a rotation matrix does not change the singular values.

algebraic properties can be formulated with a cubic *rank constraint*:

$$\det(\mathbf{E}) = 0 \tag{B.5}$$

and a cubic *trace constraint*:

$$\det(\mathbf{E}) = \quad 2\mathbf{E}\mathbf{E}^\intercal\mathbf{E} - \operatorname{trace}(\mathbf{E}\mathbf{E}^\intercal)\mathbf{E} = \mathbf{0} \tag{B.6}$$

In the presence of measurement noise, a matrix satisfying eq. (B.2) will not necessarily be valid with respect to these contraints. Therefore, the rank and trace constraints must be either enforced after estimation or the estimation algorithm must take them into account already.

There are several direct methods for the essential matrix estimation that are commonly named after the minimal number of correspondences needed to retrieve a solution. Since the essential matrix has five degrees of freedom, at least 5 point correspondences are needed. We will briefly introduce the most commonly used algorithms here. Detailed descriptions and comparisons can be found in [BBD08; RHH08].

The linear 8-point algorithm The simplest method is to estimate a linear solution for eq. (B.2) first and enforce the essential matrix constraints afterwards. First, a general 3×3 matrix $\hat{\mathbf{E}}$ satisfying the linear equation system (B.2) is computed from eight point correspondences. The arbitrary scale is fixed by the constraint $\|\hat{\mathbf{E}}\| = 1$:

$$\min_{\boldsymbol{e}} \| \begin{pmatrix} \boldsymbol{x}_1^\intercal \otimes \boldsymbol{x}_1'^\intercal \\ \vdots \\ \boldsymbol{x}_N^\intercal \otimes \boldsymbol{x}_N'^\intercal \end{pmatrix} \boldsymbol{e} \|^2 \quad \text{s.t. } \|\boldsymbol{e}\| = 1 \tag{B.7}$$

where $\boldsymbol{e} = \operatorname{vec}(\hat{\mathbf{E}}) \in \mathbb{R}^9$ with $\|\boldsymbol{e}\| = 1$, $\otimes$ is the Kronecker product and $N \geqslant 8$. A numerical solution is described in C.3.2. Afterwards, the proper essential matrix $\mathbf{E}$ closest to $\hat{\mathbf{E}}$ with respect to the Frobenius norm is computed. The solution is given by $\mathbf{E} = \mathbf{U}\operatorname{diag}(\sigma, \sigma, 0)\,\mathbf{V}^\intercal$ with $\sigma = \frac{1}{2}(\sigma_1 + \sigma_2)$ where $\hat{\mathbf{E}} = \mathbf{U}\operatorname{diag}(\sigma_1, \sigma_2, \sigma_3)\,\mathbf{V}^\intercal$ is the singular value decomposition of $\mathbf{E}$ with ordered singular values $\sigma_1 \geqslant \sigma_2 \geqslant \sigma_3$.

The 5-, 6- and 7-point algorithm Different authors propose to compute a basis $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n$ of the n-dimensional nullspace for $9 - n$ epipolar constraints with $2 \leqslant n \leqslant 4$ and find the solution $\mathbf{E} = \sum_{i=1}^{n} \lambda_i \mathbf{E}_i$ with $\lambda_n = 1$ from either the rank constraint (7-point algorithm), the trace constraint (6-point algorithm), or both constraints (5-point algorithm). The 7-point algorithm proposed by Hartley & Zisserman [HZ04] leads to finding the roots of a third-order polynomial in λ_1 with up to 3 possible solutions while the 5-point algorithm proposed by Nistér [Nis04b] yields a tenth-order polynomial equation with up to 10 possible solutions.[4] The right solution can be selected via the remaining constraints, geometric considerations, or error metrics on the essential matric (see below).

Robust estimation In order to deal with outliers, i. e., wrong 2d/2d correspondences that result from erroneous matching, robust estimation techniques must be used for relative pose estimation. A common approach is the RANSAC algorithm [FB81] (see also C.4 for a detailed description): First, preferably exact solutions are computed from a sufficiently large number of random sample sets of minimal size from the input data using one of the direct methods described above (e. g., using 5 2d/2d correspondences for the 5-point algorithm). For each sample solution $\mathbf{E}$, the set of inliers $\mathcal{I}_\mathbf{E} = \{(\boldsymbol{x}_j, \boldsymbol{x}'_j) \mid d(\boldsymbol{x}_j, \boldsymbol{x}'_j; \mathbf{E}) \leqslant \varepsilon\}$ is computed with respect to some error measure d and threshold $\varepsilon > 0$ (see next paragraph). The solution with maximal inlier count $|\mathcal{I}_\mathbf{E}|$ is used for further nonlinear refinement using all inliers.

Nonlinear refinement In the following, the parametrization of an essential matrix is described by a general parameter vector $\boldsymbol{\eta} \in \mathbb{R}^\eta$. The corresponding matrix is denoted by $\mathbf{E}_\boldsymbol{\eta}$. The set of all parameter vectors representing valid essential matrices with respect to the rank and trace constraints is denoted as $\Omega_\mathrm{E} \subset \mathbb{R}^\eta$.

Refinement of an essential matrix from $N > \eta$ 2d point correspondences is

[4] Rodehorst et al. [RHH08] advise to compute the polynomial roots from the companion matrix as described in C.2 instead of using Sturm sequences as proposed by Nistér.

defined by the following constrained nonlinear least squares problem:

$$\min_{\boldsymbol{\eta}} \sum_{j=1}^{N} d(\boldsymbol{x}_j, \boldsymbol{x}'_j; \mathbf{E}_{\boldsymbol{\eta}})^2 \quad \text{subject to } \boldsymbol{\eta} \in \Omega_{\mathrm{E}} \tag{B.8}$$

where d defines an error measure for corresponding 2d points $\boldsymbol{x}$, $\boldsymbol{x}'$ with respect to the essential matrix $\mathbf{E}$, e. g., the algebraic distance $d_{\mathrm{alg}}(\boldsymbol{x}, \boldsymbol{x}'; \mathbf{E}) = \boldsymbol{x}^{\mathsf{T}}\mathbf{E}\boldsymbol{x}'$ or geometric distance $d_{\mathrm{geom}}(\boldsymbol{x}, \boldsymbol{x}'; \mathbf{E}) = \frac{\boldsymbol{x}^{\mathsf{T}}\mathbf{E}\boldsymbol{x}'}{\|\mathbf{E}_{[1\ldots2]}\boldsymbol{x}'\|}$. An overview of common error measures can be found in A.4.1.

Since errors are not necessarily equal in both images, eq. (B.8) is often extended to minimize the symmetric distance:

$$\min_{\boldsymbol{\eta}} \sum_{j=1}^{N} d(\boldsymbol{x}_j, \boldsymbol{x}'_j; \mathbf{E}_{\boldsymbol{\eta}})^2 + d(\boldsymbol{x}'_j, \boldsymbol{x}_j; \mathbf{E}_{\boldsymbol{\eta}}^{\mathsf{T}})^2 \quad \text{subject to } \boldsymbol{\eta} \in \Omega_{\mathrm{E}} \tag{B.9}$$

Parametrization of the essential matrix in terms of a unit quaternion $\mathbf{q}$ and translation vector $\boldsymbol{t}$ with unit length

$$\mathbf{E}_{\mathbf{q},\boldsymbol{t}} = [\boldsymbol{t}]_{\times}\mathbf{R}_{\mathbf{q}} \tag{B.10}$$

reduces the ten cubic constraints (B.5) and (B.6) to two quadratic constraints $\mathbf{q}^{\mathsf{T}}\mathbf{q} = 1$ and $\boldsymbol{t}^{\mathsf{T}}\boldsymbol{t} = 1$. Combined with the geometric distance this leads to the following constrained nonlinear least squares problem:

$$\min_{\mathbf{q},\boldsymbol{t}} \sum_{j=1}^{N} \frac{(\boldsymbol{x}_j^{\mathsf{T}}[\boldsymbol{t}]_{\times}\mathbf{R}_{\mathbf{q}}\boldsymbol{x}'_j)^2}{-\boldsymbol{x}_j'^{\mathsf{T}}\mathbf{R}_{\mathbf{q}}^{\mathsf{T}}[\boldsymbol{t}]_{\times}^2\mathbf{R}_{\mathbf{q}}\boldsymbol{x}'_j} \quad \text{subject to } \|\mathbf{q}\| = 1 \text{ and } \|\boldsymbol{t}\| = 1 \tag{B.11}$$

Numerical solutions for constrained nonlinear least squares problems can be found in C.3.4. An initial solution can be provided by one of the direct methods described above.

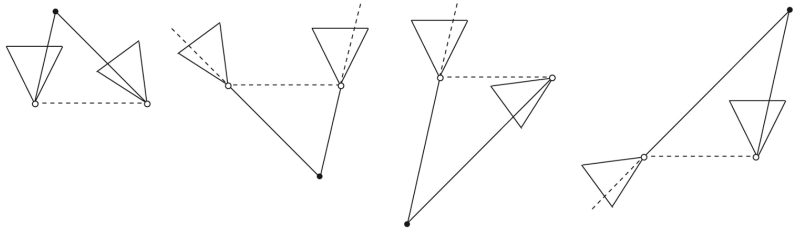

Figure B.2. Illustration of the four possible solutions for extracting the relative pose and a 3d point from the essential matrix. Only in the leftmost solution the reconstructed 3d point is in front of both cameras.

B.2.3 Recovering the Relative Pose

Given an essential matrix $\mathbf{E}$ for two images, the relative pose $[\mathbf{R} \mid \boldsymbol{t}]$ can be retrieved up to scale with a four-fold ambiguity via decomposition of $\mathbf{E}$ into $\hat{\mathbf{S}}\hat{\mathbf{R}}$ where $\hat{\mathbf{S}}$ is a skew-symmetric matrix and $\hat{\mathbf{R}}$ is an orthogonal matrix (see Sec. 9.6.2 in [HZ04]):

$$\hat{\mathbf{R}} = \mathbf{U}\mathbf{W}\mathbf{V}^\mathsf{T} \text{ or } \mathbf{U}\mathbf{W}^\mathsf{T}\mathbf{V}^\mathsf{T} \quad \text{and} \quad \hat{\mathbf{S}} = \mathbf{U}\mathbf{Z}\mathbf{U}^\mathsf{T} \tag{B.12}$$

where $\mathbf{E} = \mathbf{U}\,\mathrm{diag}(\sigma, \sigma, 0)\mathbf{V}^\mathsf{T}$ is the singular value decomposition of the essential matrix and matrices $\mathbf{W}, \mathbf{Z}$ are defined by:

$$\mathbf{W} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{Z} = \begin{pmatrix} 0 & \sigma & 0 \\ -\sigma & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

From $\hat{\mathbf{S}} \sim [\boldsymbol{t}]_\times$ we obtain two solutions for the relative translation up to scale as $\hat{\boldsymbol{t}} = \pm\frac{1}{\sigma}\boldsymbol{u}_3$ with $\|\hat{\boldsymbol{t}}\| = 1$.

A common strategy to find the right solution from the four possible solutions $[\hat{\mathbf{R}}_{1/2} \mid \hat{\boldsymbol{t}}_{1/2}]$ without a priori knowledge is to triangulate 3d points subject to the relative poses. The solution that yields the highest number of points in front of both cameras is supposed to be the most plausible one. An example is shown in Fig. B.2.

B.2.4 Critical Motions

For certain motion configuration, the epipolar equation degenerates leading to ambiguous solutions. This is the case for the two familiar scenarios of pure translation and planar motion. A third degenerate case is associated with pure rotation of the camera.

Pure rotation For motions contains insignificant translation, the epipolar equation degenerates to $\boldsymbol{x} \sim \mathbf{R}\boldsymbol{x}'$ and the essential matrix becomes a scaled rotation matrix. Although this case can be solved as an instance of the relative rotation problem (see A.3.2), it is impossible to recover the distance of 3d points $\boldsymbol{X}_j$ to the cameras. Hence, pure rotation is not suitable for initialization of Structure from Motion.[5]

Pure translation If the motion consists of non-zero translation and zero rotation, the essential matrix is defined up to scale by $\mathbf{E} \sim [\boldsymbol{t}]_\times$. The pure translational essential matrix has only two degrees of freedom.

Planar motion Essential matrices for planar motion (i. e., translation is perpendicular to the rotation axis) have been examined in detail by Maybank [May93]. It is shown that the symmetric part of the essential matrix has rank 2 in this case, imposing the additional constraint $\det(\mathbf{E}_S) = 0$ with $\mathbf{E}_S = \frac{1}{2}(\mathbf{E} + \mathbf{E}^\intercal)$. The number of degrees of freedom is reduced to four. If the essential matrix is parametrized with a unit quaternion and unit translation vector as in eq. (B.11), the constraint is given by $\boldsymbol{q}^\intercal \boldsymbol{t} = 0$.

B.3 Absolute Pose Estimation

Finding the absolute pose from normalized image coordinates $\boldsymbol{x}$ of 3d points $\boldsymbol{X}$ for the perspective camera model is also known as the *Perspective-*

[5] Note that the pure rotation case can be detected automatically using for instance the *Geometric Robust Information Criterion (GRIC)* proposed by Torr [Tor97].

n-Point (PnP) problem[6] where n denotes the number of 2d/3d correspondences used. Since each 3d points contributes 2 scalar equations and the camera pose has 6 degrees of freedom, at least 3 point correspondences are needed to provide a solution [Nis04a]. Hence, the minimal problem is also denoted as the *P3P problem*. However, P3P solvers typically yield up to 4 solutions, so a 4-th point correspondence is used in general to select the best pose configuration [Gao+03]. The minimal configuration for the PnP problem is depicted in Fig. B.3.

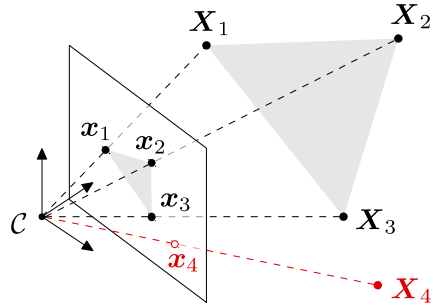

Figure B.3. Illustration of the Perspective-Three-Point (P3P) problem estimating the absolute camera pose from three 2d/3d correspondences. An additional point is used to determine a unique solution.

In photogrammetry, the historically first and most simple closed-form approach to solve eq. (4.9) is given by the *Direct Linear Transform (DLT)* [HZ04], finding an affine transformation $[\mathbf{A} \mid \boldsymbol{b}] = [\mathbf{R}^\intercal \mid -\mathbf{R}^\intercal \boldsymbol{t}]$ such that $\boldsymbol{x} \sim \mathbf{A}\boldsymbol{X} + \boldsymbol{b}$ for each normalized 2d point $\boldsymbol{x}$ and corresponding 3d point $\boldsymbol{X}$, without constraining $[\mathbf{A} \mid \boldsymbol{b}]$ to describe a proper Euclidean transformation:

$$\min_{\mathbf{A}\in\mathbb{R}^{3\times3},\, \boldsymbol{b}\in\mathbb{R}^3} \sum_{j=1}^{N} \|[\boldsymbol{x}_j]_\times(\mathbf{A}\boldsymbol{X}_j + \boldsymbol{b})\|^2 \quad \text{subject to } \|\boldsymbol{A}_3\|^2 = 1 \tag{B.13}$$

with $N \geqslant 6$. The orthonormality constraints for $\mathbf{R} = \mathbf{A}^\intercal$ are enforced

[6] introduced by Fischler & Boyles in their seminal paper on RANSAC [FB81]

afterwards, e. g., via SVD (see Sec. 2.3.2), leading to non-optimal solutions.

Apart from DLT, most existing approaches for calibrated cameras attempt to retrieve the 3d coordinates $\boldsymbol{X}'_j \sim \boldsymbol{x}_j$ of the projected points within the camera coordinate frame and solve for the camera pose from 3d/3d correspondences $(\boldsymbol{X}_j, \boldsymbol{X}'_j)$ afterwards (see A.3.2). In this work, we use state-of-the-art methods to solve this problem: the P3P solver proposed by Gao et al. [Gao+03] to estimate minimal solutions from 4 2d/3d correspondences and the PnP solver proposed by Lepetit et al. [LMF09] for $N > 4$ 2d/3d correspondences.[7] The first method solves $O(N)$ equations for the distances λ_j between 3d points $\boldsymbol{X}_j$ and the camera center while the second expresses their coordinates as a weighted sum of four virtual control points, leading to an equation system with fixed size. For more details we refer the reader to [Gao+03] and [LMF09].

Robust estimation in the presence of outliers is performed as described in B.2.2 using a RANSAC approach. An initial solution is found via the P3P solver. The resulting absolute pose is refined using all inliers with respect to an appropriate error measure as described next.

Nonlinear refinement Parametrizing the inverse camera pose by a general parameter vector $\bar{\boldsymbol{\mu}} \in \mathbb{R}^{\mu}$, i. e., $\mathbf{T}^{-1} = \mathbf{T}_{\bar{\boldsymbol{\mu}}}$, refinement of the absolute pose from $N > \mu$ 2d/3d point correspondences is defined by the following constrained nonlinear least squares problem:

$$\min_{\bar{\boldsymbol{\mu}}} \sum_{j=1}^{N} d(\boldsymbol{x}_j, \mathbf{T}_{\bar{\boldsymbol{\mu}}} \boldsymbol{X})^2 \quad \text{subject to } \bar{\boldsymbol{\mu}} \in \Omega_{\mathrm{M}} \tag{B.14}$$

where d defines an error measure between a 2d point $\boldsymbol{x}$ and a 3d point $\boldsymbol{X}$ with respect to the specific camera model. An overview of common error measures can be found in A.4.2.

In this thesis, we use the planar reprojection error $d_{\mathcal{P}}(\boldsymbol{x}, \boldsymbol{X}) = \|\boldsymbol{x} - \mathcal{P}(\boldsymbol{X})\|$ resp. spherical reprojection error $d_{\mathcal{S}}(\boldsymbol{v}, \boldsymbol{X}) = \|\boldsymbol{v} - \mathcal{S}(\boldsymbol{X})\|$ for calibrated cameras to keep the complexity of the error function as low as possible

[7] These methods are also implemented in the OpenCV function `solvePnP` [Bra00].

while minimizing a geometrically reasonable error related to the image domain. For the case that covariance matrices $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ are given for the 2d points, weighted variants of the reprojection errors are defined by replacing the Euclidean distance with the Mahalanobis distance $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2_{\boldsymbol{\Sigma}_{\boldsymbol{x}}} = (\boldsymbol{x} - \hat{\boldsymbol{x}})^\intercal \boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1} (\boldsymbol{x} - \hat{\boldsymbol{x}})$.[8]

The resulting constrained nonlinear least squares problem can be solved numerically via the Levenberg-Marquardt algorithm as described in C.3.4 starting with a solution provided by one of the direct PnP solvers mentioned above.

B.4 Triangulation

Once the relative camera pose between two images is known, depth information can be recovered from corresponding 2d points via *triangulation*. For sake of completeness we present a simple linear approach here. For further details and optimal methods we refer the reader to [HS97].

We assume that m camera poses are given by $[\mathbf{R}_k \mid \boldsymbol{t}_k]$, $k = 1, \ldots, m$. Given are normalized image coordinates $\boldsymbol{x}_k$ of the same 3d point $\boldsymbol{X}$ in all views. Triangulation methods[9] estimate the original 3d point from intersecting the projection rays, i. e., the 3d lines $\boldsymbol{L}_k(\lambda) = \boldsymbol{t}_k + \lambda \, \mathbf{R}_k \boldsymbol{v}_k$ with direction vectors $\boldsymbol{v}_k = \mathcal{S}(\boldsymbol{x}_k)$ through the camera centers and the projected points. However, due to measurement noise and pose estimation errors, corresponding projection rays do not necessarily meet at the same point in 3d space.

A solution to eq. (4.10) can be found by estimating the 3d point that is mutually closest to all projection rays – also known as *mid-point method* for the case of two images:

$$\min_{\boldsymbol{X} \in \mathbb{R}^3} \sum_{k=1}^{m} d(\boldsymbol{X}, \boldsymbol{L}_k)^2 \tag{B.15}$$

[8] Note that $\boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1}$ is replaced by the pseudoinverse $\boldsymbol{\Sigma}_{\boldsymbol{x}}^{\dagger}$ when $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ is singular.

[9] The term "triangulation" stems from the fact that the 3d lines between two camera centers and the projected 3d point form a triangle with the camera baseline in 3d space.

where d is a distance measure between 3d points and 3d lines. It is recommended to validate 3d points triangulated via the mid-point method with respect to their reprojection errors e. g., using the *X84* rule for outlier rejection as described in C.4.

Using the geometric point-line distance $d(\boldsymbol{X}, \boldsymbol{L}_k) = \min_\lambda \|\boldsymbol{X} - \boldsymbol{L}_k(\lambda)\|$ leads to the following linear least squares problem:

$$\min_{\boldsymbol{X}\in\mathbb{R}^3,\, \lambda_{1,\ldots,m}} \sum_{k=1}^{m} \|\boldsymbol{X} - \boldsymbol{t}_k - \lambda_k \mathbf{R}_k \boldsymbol{v}_k\|^2 \tag{B.16}$$

Since the geometric point-line distance can also be written in closed-form as $d(\boldsymbol{X}, \boldsymbol{L}_k) = \|\mathbf{R}_k \boldsymbol{v}_k \times (\boldsymbol{X} - \boldsymbol{t}_k)\|$, the distance parameters $\lambda_{1,\ldots,m}$ can be eliminated from eq. (B.15), leading to:

$$\min_{\boldsymbol{X}\in\mathbb{R}^3} \sum_{k=1}^{m} \|[\mathbf{R}_k \boldsymbol{v}_k]_\times \boldsymbol{X} - (\mathbf{R}_k \boldsymbol{v}_k \times \boldsymbol{t}_k)\|^2 \tag{B.17}$$

The resulting 3d point can be refined w.r.t. the reprojection error using non-linear optimization methods (e. g., the Levenberg-Marquardt algorithm).

Note that a scaling of the baseline length results in an equal scaling of the reconstructed 3d points. Given that the relative translation is defined only up to an unknown scale when estimated from the epipolar geometry, 3d structure is only provided up to the same scale.

B.5 Bundle Adjustment

Bundle adjustment[10] is often applied as an intermediate or final step of feature-based 3d reconstruction algorithms. Bundle adjustment is defined as refining camera parameters and 3d structure simultaneously to obtain

[10] Originally developed by the photogrammetry and geodesy community during the 1950s, bundle adjustment has been researched in the context of computer vision since the 1990s. The name refers to the bundles of light rays emerging from the 3d features and intersecting in the camera centers [Tri+00].

an optimal visual reconstruction with respect to some given cost function describing the model fitting error [Tri+00]. This can be described formally as a large sparse parameter estimation problem with respect to intrinsic and extrinsic camera parameters and 3d structure parameters.

First, we will formalize point-based bundle adjustment with general parametrization. The following notation is employed (as introduced in Sec. 2.3.1): Parameter vectors $\boldsymbol{\mu} \in \mathbb{R}^{\mu}$ are used to describe camera poses $\mathbf{T} = [\mathbf{R} \mid \boldsymbol{t}]$ resp. $\bar{\boldsymbol{\mu}}$ to describe inverse camera poses $\mathbf{T}^{-1}$. Parameter vectors $\boldsymbol{\chi} \in \mathbb{R}^{\chi}$ describe 3d points $\boldsymbol{X}$ respectively. Given constant intrinsic camera parameters $\boldsymbol{\kappa} \in \mathbb{R}^{\kappa}$ (see Sec. 4.2.1), m inverse camera poses $\mathbf{T}_k^{-1}$, $k = 1, \ldots, m$, parametrized by $\bar{\boldsymbol{\mu}}_k \in \mathbb{R}^{\mu}$ respectively, and N 3d points $\boldsymbol{X}_j$, $j = 1, \ldots, N$, parametrized by $\boldsymbol{\chi}_j \in \mathbb{R}^{\chi}$ each, the error function for bundle adjustment is given by the reprojection errors between 3d points and their corresponding 2d points in the camera images $\boldsymbol{u}_{k,j}$, $(k, j) \in \mathcal{V}$ where $\boldsymbol{u}_{k,j}$ is the projection of the j-th 3d point $\boldsymbol{X}_j$ into the k-th camera image and $\mathcal{V} \subset \{1, \ldots, m\} \times \{1, \ldots, N\}$ describes the visibility of 3d points. In general, the L_2-norm of the reprojection errors is minimized, resulting in the following nonlinear least squares problem:

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{F}(\boldsymbol{\theta})\|^2 = \sum_{(k,j) \in \mathcal{V}} d(\underbrace{\mathcal{K}(\mathcal{M}(\bar{\boldsymbol{\mu}}_k, \boldsymbol{\chi}_j))}_{\hat{\boldsymbol{u}}(\bar{\boldsymbol{\mu}}_k, \boldsymbol{\chi}_j)}, \boldsymbol{u}_{k,j})^2 \tag{B.18}$$

where $\boldsymbol{\theta} = (\boldsymbol{\kappa}, \bar{\boldsymbol{\mu}}_1, \ldots, \bar{\boldsymbol{\mu}}_m, \boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_N) \in \mathbb{R}^P$ with $P = \kappa + m\mu + N\chi$ is the joint parameter vector, $\mathcal{K}$ is the camera function defined in Sec. 4.2.3, and d is an error metric for 2d points in the image space which is in general either the Euclidean distance or the Mahalanobis distance if covariance matrices for the observed 2d points are supplied. $\mathcal{M}(\boldsymbol{\mu}, \boldsymbol{\chi})$ denotes rigid motion of a 3d point subject to rigid motion parameters $\boldsymbol{\mu}$ as defined in Sec. 2.3.1.

In eq. (B.18), $\hat{\boldsymbol{u}}_{k,j} = \mathcal{K}(\mathcal{M}(\bar{\boldsymbol{\mu}}_k, \boldsymbol{\chi}_j))$ is the predicted 2d point for camera pose $\mathbf{T}_k$ and 3d point $\boldsymbol{X}_j$. For convenience, the prediction function is abbreviated as $\hat{\boldsymbol{u}} = \mathcal{K} \circ \mathcal{M}$.

The number of parameters per camera pose is typically $\mu = 6$ for a minimal rotation parametrization, $\mu = 7$ for unit quaternions, or $\mu = 8$

for dual quaternion representation of motion. 3d points are in general parametrized by their Euclidean coordinates, i. e., $\boldsymbol{\chi}_j = \boldsymbol{X}_j$, with $\chi = 3$ parameters each. The number of intrinsic parameters κ depends on the actual camera model.

For the calibrated camera case, the intrinsic camera parameters $\boldsymbol{\kappa}$ can be omitted from the parameter vector and the 2d points can be replaced by their normalized representations $\boldsymbol{x}_{k,j} = \mathcal{U}_{\mathcal{P}}(\boldsymbol{u}_{k,j})$:

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{F}(\boldsymbol{\theta})\|^2 = \sum_{(k,j)\in\mathcal{V}} d(\underbrace{\mathcal{P}(\mathcal{M}(\bar{\boldsymbol{\mu}}_k, \boldsymbol{\chi}_j))}_{\hat{\boldsymbol{x}}(\bar{\boldsymbol{\mu}}_k, \boldsymbol{\chi}_j)}, \boldsymbol{x}_{k,j})^2 \tag{B.19}$$

where $\boldsymbol{\theta} = (\bar{\boldsymbol{\mu}}_1, \ldots, \bar{\boldsymbol{\mu}}_m, \boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_N)$ and d is an error metric for 2d points in the normalized image plane. The dimensionality of the parameter space is reduced to $P = m\mu + N\chi$.

Using the unit sphere to represent normalized points instead, i. e., $\boldsymbol{v}_{k,j} = \mathcal{U}_{\mathcal{S}}(\boldsymbol{u}_{k,j})$, we define spherical bundle adjustment that can be applied to fisheye lens or omnidirectional cameras:

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{F}(\boldsymbol{\theta})\|^2 = \sum_{(k,j)\in\mathcal{V}} d(\underbrace{\mathcal{S}(\mathcal{M}(\bar{\boldsymbol{\mu}}_k, \boldsymbol{\chi}_j))}_{\hat{\boldsymbol{v}}(\bar{\boldsymbol{\mu}}_k, \boldsymbol{\chi}_j)}, \boldsymbol{v}_{k,j})^2 \tag{B.20}$$

In this case, d defines an error metric on $\mathbb{S}^2$, e. g., the angle metric $d_{\angle}(\hat{\boldsymbol{v}}, \boldsymbol{v})$ or Euclidean distance $d_{\text{geom}}(\hat{\boldsymbol{v}}, \boldsymbol{v}) = \|\hat{\boldsymbol{v}} - \boldsymbol{v}\|$.

To avoid ambiguities in the solution known as *gauge freedoms*, a common solution is to fix the first camera pose in order to avoid the absolute orientation ambiguity of the reconstructed scene. Additionally, the length of the translation vector between the first and the second camera is fixed in order to avoid the absolute scale ambiguity [Bar03]. This reduces the number of camera pose parameters by $\mu + 1$ effectively.

In the following we will refer to planar bundle adjustment defined by eq. (B.19) using the Euclidean norm d. The error function is defined by the L_2-norm of the vector-valued function $\boldsymbol{F} : \mathbb{R}^P \to \mathbb{R}^{2M}$ where $M = |\mathcal{V}|$ is the number of 2d point observations in all images, denoted as $G : \mathbb{R}^P \to$

$\mathbb{R}, \boldsymbol{\theta} \mapsto \|\boldsymbol{F}(\boldsymbol{\theta})\|^2$. Enumerating the visibility set $\mathcal{V}$ as $((k_1, j_1), \ldots, (k_M, j_M))$, $\boldsymbol{F}$ is defined as:

$$\boldsymbol{F}(\boldsymbol{\theta}) = \begin{pmatrix} f_1(\boldsymbol{\theta}) \\ \vdots \\ f_M(\boldsymbol{\theta}) \end{pmatrix} \text{ with } f_i(\boldsymbol{\theta}) = (\hat{\boldsymbol{x}}(\bar{\boldsymbol{\mu}}_{k_i}, \boldsymbol{\chi}_{j_i}) - \boldsymbol{x}_{k_i,j_i})_{[1\ldots2]} \tag{B.21}$$

The prevailing approach to find a local minimizer of the nonlinear least squares problem $\min_{\boldsymbol{\theta}} \|\boldsymbol{F}(\boldsymbol{\theta})\|^2$ numerically given an initial solution $\boldsymbol{\theta}_0$ is via the *Levenberg-Marquardt algorithm* (see C.3.3). Similar to the *Gauss-Newton algorithm,* the solution of a nonlinear least squares problem is approximated by solving the linear least squares problem resulting from linearizing $\boldsymbol{F}$ at the current parameter estimate iteratively until convergence. This involves solving the *normal equations* in each step:

$$\mathbf{J}_{\boldsymbol{F}}{}^{\mathsf{T}}\mathbf{J}_{\boldsymbol{F}}(\boldsymbol{\theta} - \boldsymbol{\theta}_0) = -\mathbf{J}_{\boldsymbol{F}}{}^{\mathsf{T}}\boldsymbol{F}(\boldsymbol{\theta}_0) \tag{B.22}$$

where $\mathbf{J}_{\boldsymbol{F}} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}_0)$ is the Jacobian matrix of $\boldsymbol{F}$ evaluated at $\boldsymbol{\theta}_0$ and $\mathbf{J}_{\boldsymbol{F}}{}^{\mathsf{T}}\mathbf{J}_{\boldsymbol{F}}$ is an approximation to the Hessian matrix of the objective function G.

The Jacobian matrix of $\boldsymbol{F}$ is given by:

$$\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{\theta}} = \begin{pmatrix} \frac{\partial f_1}{\partial \bar{\boldsymbol{\mu}}_1} & \cdots & \frac{\partial f_1}{\partial \bar{\boldsymbol{\mu}}_m} & \frac{\partial f_1}{\partial \boldsymbol{\chi}_1} & \cdots & \frac{\partial f_1}{\partial \boldsymbol{\chi}_N} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_M}{\partial \bar{\boldsymbol{\mu}}_1} & \cdots & \frac{\partial f_M}{\partial \bar{\boldsymbol{\mu}}_m} & \frac{\partial f_M}{\partial \boldsymbol{\chi}_1} & \cdots & \frac{\partial f_M}{\partial \boldsymbol{\chi}_N} \end{pmatrix} \tag{B.23}$$

with partial derivatives

$$\frac{\partial f_i}{\partial \bar{\boldsymbol{\mu}}_k}(\boldsymbol{\theta}) = \begin{cases} \frac{\partial \hat{\boldsymbol{x}}}{\partial \bar{\boldsymbol{\mu}}}(\bar{\boldsymbol{\mu}}_{k_i}, \boldsymbol{\chi}_{j_i}) & \text{for } k = k_i \\ \mathbf{0} & \text{else} \end{cases}$$

and

$$\frac{\partial f_i}{\partial \boldsymbol{\chi}_j}(\boldsymbol{\theta}) = \begin{cases} \frac{\partial \hat{\boldsymbol{x}}}{\partial \boldsymbol{\chi}}(\bar{\boldsymbol{\mu}}_{k_i}, \boldsymbol{\chi}_{j_i}) & \text{for } j = j_i \\ \mathbf{0} & \text{else} \end{cases}$$

Since each observation depends on $\mu + \chi$ parameters only, the Jacobian matrix $\mathbf{J}_F$ has a sparse structure which is illustrated in Fig. B.4 and B.5. The ratio of non-zero elements is bounded by $\frac{\mu+\chi}{P}$ which is within the range of 5% to below 1% for typical bundle adjustment problems.

The matrix $\mathbf{H}_G = \mathbf{J}_F{}^\mathsf{T}\mathbf{J}_F$ exhibits the following sparse block structure:

$$\mathbf{H}_G = \begin{pmatrix} \mathbf{H}_{1,1} & \dots & \mathbf{H}_{1,m+N} \\ \vdots & \ddots & \vdots \\ \mathbf{H}_{m+N,1} & \dots & \mathbf{H}_{m+N,m+N} \end{pmatrix} \tag{B.24}$$

where all submatrices are zero except for

$$\begin{aligned} \mathbf{H}_{k,k} &= \sum_{i=1}^{M} \frac{\partial \boldsymbol{f}_i}{\partial \bar{\boldsymbol{\mu}}_k}(\boldsymbol{\theta}_0)^\mathsf{T} \frac{\partial \boldsymbol{f}_i}{\partial \bar{\boldsymbol{\mu}}_k}(\boldsymbol{\theta}_0) \quad \text{for } 1 \leqslant k \leqslant m \\ \mathbf{H}_{j+m,j+m} &= \sum_{i=1}^{M} \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{\chi}_j}(\boldsymbol{\theta}_0)^\mathsf{T} \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{\chi}_j}(\boldsymbol{\theta}_0) \quad \text{for } 1 \leqslant j \leqslant N \\ \mathbf{H}_{k,j+m} = \mathbf{H}_{j+m,k}^\mathsf{T} &= \sum_{i=1}^{M} \frac{\partial \boldsymbol{f}_i}{\partial \bar{\boldsymbol{\mu}}_k}(\boldsymbol{\theta}_0)^\mathsf{T} \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{\chi}_j}(\boldsymbol{\theta}_0) \quad \text{for } 1 \leqslant k \leqslant m,\ 1 \leqslant j \leqslant N \end{aligned}$$

Note that the off-diagonal block matrices $\mathbf{H}_{k,j+m}$ and $\mathbf{H}_{j+m,k}$ are only non-zero if the j-th 3d point is visible in the k-th image. Hence, the ratio of non-zero elements in $\mathbf{H}_G$ is bounded by $\frac{m\mu^2+N\chi^2+2M\mu\chi}{P^2}$ which is in general below 10%.

The sparse structure of the approximate Hessian matrix reveals the sparse nature of the normal equations (B.22) as illustrated in Fig. B.4 and B.5. Hence, bundle adjustment can be solved efficiently in spite of the large number of parameters and observations involved using sparse implementations of the Levenberg-Marquardt algorithm. A commonly used C/C++ implementation is the `sba` software package by Lourakis & Argyros [LA09]. For further details on sparse bundle adjustment we refer the reader to the comprehensive discussion in [Tri+00] or Sec. A6.3 in [HZ04].

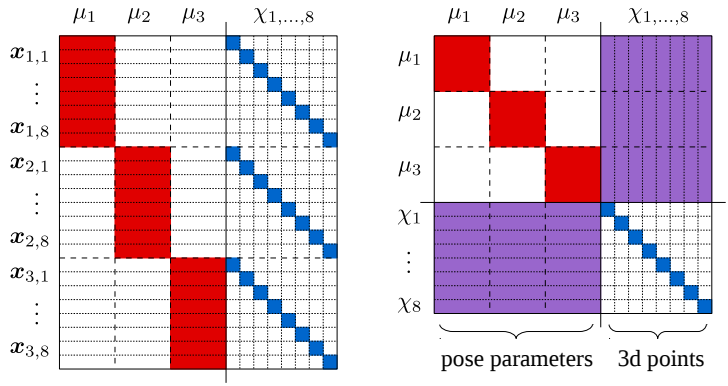

Figure B.4. Form of the Jacobian matrix $\mathbf{J}_F$ (left) and approximate Hessian matrix $\mathbf{H}_G = \mathbf{J}_F{}^\mathsf{T}\mathbf{J}_F$ (right) for a bundle adjustment example consisting of $m = 3$ images and $N = 8$ 3d points. Note that the lower right part of $\mathbf{H}_G$ constitutes the major part of the matrix in practical problem instances, leading to the "arrow head" form.

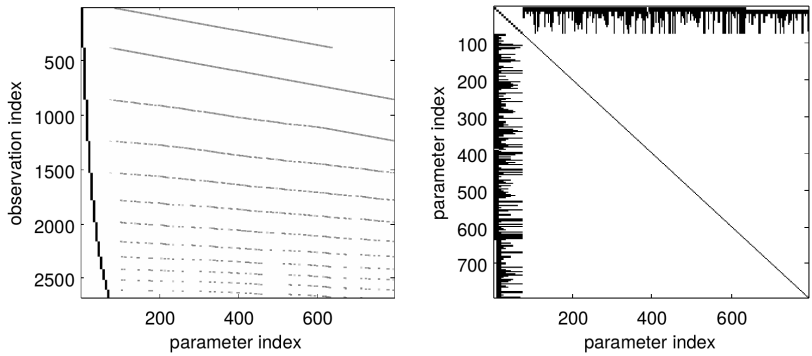

Figure B.5. Jacobian matrix $\mathbf{J}_F$ (left) and approximate Hessian matrix $\mathbf{H}_G = \mathbf{J}_F{}^\mathsf{T}\mathbf{J}_F$ (right) for a moderately sized bundle adjustment problem consisting of $m = 12$ images, $N = 240$ 3d points, $P = 792$ parameters, and $M = 1339$ 2d points. Black areas correspond to non-zero entries. The fill degree of $\mathbf{J}_F$ and $\mathbf{H}_G$ is 1.14% and 8.1% respectively.

Appendix C

Math and Numerics

C.1 Linear Algebra

The *cross product* between 3-vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^3$ is defined as:

$$\boldsymbol{a} \times \boldsymbol{b} = \begin{pmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{pmatrix} \boldsymbol{b} = [\boldsymbol{a}]_\times \boldsymbol{b} \tag{C.1}$$

with the property $\boldsymbol{a} \times \boldsymbol{b} = \|\boldsymbol{a}\|\|\boldsymbol{b}\| \sin(\varphi)\boldsymbol{c}$ where φ is the angle between $\boldsymbol{a}$ and $\boldsymbol{b}$ and $\boldsymbol{c}$ is a unit vector perpendicular to both $\boldsymbol{a}$ and $\boldsymbol{b}$. For $\boldsymbol{a} \parallel \boldsymbol{b}$, the cross product yields the zero vector $\mathbf{0}$.

The *Kronecker product* between matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{p \times q}$ is defined as the $mp \times nq$ block matrix:

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} A_{1,1}\mathbf{B} & \dots & A_{1,n}\mathbf{B} \\ \vdots & \dots & \vdots \\ A_{m,1}\mathbf{B} & \dots & A_{m,n}\mathbf{B} \end{pmatrix} \tag{C.2}$$

The *singular value decomposition (SVD)* of a real matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a factorization of the form:

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^\mathsf{T} \tag{C.3}$$

where $\mathbf{U} \in \mathbb{R}^{m \times m}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$ are orthogonal matrices and $\mathbf{S}$ is an $m \times n$ rectangular diagonal matrix with non-negative real values σ_i on the diagonal, called the *singular values* of $\mathbf{A}$.

An *eigenvector* defines a non-zero vector that is invariant up to scale under linear transformation described by a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$:

$$\mathbf{A}\boldsymbol{x} = \lambda \boldsymbol{x}, \quad \boldsymbol{x} \neq \mathbf{0} \tag{C.4}$$

where λ is called the *eigenvalue* corresponding to the eigenvector $\boldsymbol{x}$.

A *pseudoinverse* $\mathbf{A}^\dagger$ of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a generalization of the inverse matrix.[1] For matrices with full column rank, the pseudoinverse is defined by the $n \times m$ matrix:

$$\mathbf{A}^\dagger = (\mathbf{A}^\intercal \mathbf{A})^{-1} \mathbf{A}^\intercal \tag{C.5}$$

with the property $\mathbf{A}^\dagger \mathbf{A} = \mathbf{I}_n$. In general, a pseudoinverse of a singular matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ can be computed via SVD as $\mathbf{A}^\dagger = \mathbf{V}\mathbf{S}^\dagger \mathbf{U}^\intercal$ where $\mathbf{S}^\dagger$ is a diagonal matrix with $(\mathbf{S}^\dagger)_{i,i} = \frac{1}{\sigma_i}$ if $\sigma_i \neq 0$ and 0 else.

C.2 Solving Polynomial Equations

Computing the roots of a polynomial can be posed as an eigenvalue problem [EM95].[2] Consider the following equation for a univariate polynomial $p(t)$ of degree n:

$$p(t) = \sum_{i=1}^{n+1} a_i t^{i-1} = a_1 + a_2 t + \cdots + a_{n+1} t^n = 0 \tag{C.6}$$

We can describe eq. (C.6) equivalently by an eigenvalue problem with respect to the *companion matrix* of $p(t)$ which is given by the following non-symmetric $n \times n$ matrix:

$$\begin{pmatrix} \mathbf{0} & \mathbf{I}_{n-1} \\ \frac{-a_1}{a_{n+1}} & \cdots \; \frac{-a_n}{a_{n+1}} \end{pmatrix} \boldsymbol{x} = \lambda \boldsymbol{x} \tag{C.7}$$

[1] Although there exist different definitions, the term refers most commonly to the *Moore–Penrose pseudoinverse*, named after the American mathematician Eliakim H. Moore (*1862, †1932) and the British mathematician and theoretical physicist Roger Penrose (*1931).

[2] This method is also implemented in the MATLAB function `roots` [Mat13].

Hence, for each solution $\boldsymbol{x}$ and corresponding eigenvalue λ holds:

$$-\frac{1}{a_{n+1}}\sum_{i=1}^{n} a_i x_i = \lambda x_n \quad \Leftrightarrow \quad \sum_{i=1}^{n} a_i x_i + a_{n+1}\lambda x_n = 0 \tag{C.8}$$

and

$$x_2 = \lambda x_1,\ \ldots,\ x_n = \lambda x_{n-1} \tag{C.9}$$

Equation (C.9) implies $x_i = \lambda^{i-1} x_1$ for all $i = 1, \ldots, n$. Under the assumption that $x_1 \neq 0$, division of eq. (C.8) by x_1 yields:

$$\sum_{i=1}^{n} a_i \lambda^{i-1} + a_{n+1}\lambda\lambda^{n-1} = \sum_{i=1}^{n+1} a_i \lambda^{i-1} = 0 \tag{C.10}$$

so λ is a solution to eq. (C.6).

C.3 Least Squares Fitting

C.3.1 Linear Least Squares

Consider an unconstrained linear least squares problem of the form:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 \tag{C.11}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$ with $m \geqslant n$, $\boldsymbol{b} \in \mathbb{R}^m$ and $\boldsymbol{x} \in \mathbb{R}^n$. Considering the vector $\boldsymbol{b}$ to represent *observations* or *measurements* and $\mathbf{A}$ as the *model matrix* relating parameters $\boldsymbol{x}$ to predicted observations, eq. (C.11) describes the problem of *linear model fitting*. Setting the derivative of *eq.* (C.11) with respect to $\boldsymbol{x}$ to zero yields the so-called *normal equations*:[3]

$$\mathbf{A}^\intercal\mathbf{A}\boldsymbol{x} = \mathbf{A}^\intercal\boldsymbol{b} \tag{C.12}$$

[3] The name stems from the fact that the vector $\mathbf{A}\boldsymbol{x} - \boldsymbol{b}$ is normal to the column space of $\mathbf{A}$.

Under the assumption that $\mathbf{A}$ has full column rank, $\mathbf{A}^\intercal\mathbf{A}$ is a symmetric, positive definite $n \times n$ matrix and eq. (C.11) has a unique solution:

$$\boldsymbol{x}^* = (\mathbf{A}^\intercal\mathbf{A})^{-1}\mathbf{A}^\intercal\boldsymbol{b} \tag{C.13}$$

The pseudoinverse $\mathbf{A}^\dagger = (\mathbf{A}^\intercal\mathbf{A})^{-1}\mathbf{A}^\intercal$ in eq. (C.13) can be computed numerically, e. g., via QR factorization of $\mathbf{A}$.

Given a covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{b}} \in \mathbb{R}^{m\times m}$ for observations $\boldsymbol{b}$, the covariance matrix of the resulting parameter vector $\boldsymbol{x}^*$ is computed via *error propagation* (see also A.5):

$$\boldsymbol{\Sigma}_{\boldsymbol{x}^*} = (\mathbf{A}^\intercal\mathbf{A})^{-1}\mathbf{A}^\intercal\boldsymbol{\Sigma}_{\boldsymbol{b}}\mathbf{A}(\mathbf{A}^\intercal\mathbf{A})^{-1} \tag{C.14}$$

Otherwise the covariance of the resulting $\boldsymbol{x}^*$ is simply given by $(\mathbf{A}^\intercal\mathbf{A})^{-1}$.

Weighted Least Squares Given a covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{b}} \in \mathbb{R}^{m\times m}$ for observations $\boldsymbol{b}$, the linear least squares problem can alternatively be solved with respect to the Mahalanobis distance instead of the Euclidean distance as in eq. (C.11):

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2_{\boldsymbol{\Sigma}_{\boldsymbol{b}}} \tag{C.15}$$

where $\|\boldsymbol{x}\|^2_{\boldsymbol{\Sigma}_{\boldsymbol{b}}} = \boldsymbol{x}^\intercal\boldsymbol{\Sigma}_{\boldsymbol{b}}^{-1}\boldsymbol{x}$ is the respective Mahalanobis norm. The resulting *weighted normal equations* are:

$$\mathbf{A}^\intercal\boldsymbol{\Sigma}_{\boldsymbol{b}}^{-1}\mathbf{A}\boldsymbol{x} = \mathbf{A}^\intercal\boldsymbol{\Sigma}_{\boldsymbol{b}}^{-1}\boldsymbol{b} \tag{C.16}$$

which can be solved in the same way as eq. (C.12).

The uncertainty of the resulting parameter vector $\boldsymbol{x}^*$ is described by:

$$\boldsymbol{\Sigma}_{\boldsymbol{x}^*} = (\mathbf{A}^\intercal\boldsymbol{\Sigma}_{\boldsymbol{b}}^{-1}\mathbf{A})^{-1} \tag{C.17}$$

As motivated in A.4, the matrix inverse $\boldsymbol{\Sigma}_{\boldsymbol{b}}^{-1}$ can be replaced by the pseudoinverse $\boldsymbol{\Sigma}_{\boldsymbol{b}}^\dagger$ in eqs. (C.16, C.17) for singular covariance matrices $\boldsymbol{\Sigma}_{\boldsymbol{b}}$ resulting from intrinsically constrained measurements $\boldsymbol{b}$.

C.3.2 Constrained Linear Least Squares

Consider a constrained linear least squares problem of the form:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\mathbf{A}\boldsymbol{x}\|^2 \quad \text{subject to } \boldsymbol{x}^\mathsf{T}\mathbf{C}\boldsymbol{x} = 1 \tag{C.18}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$ with $m \geqslant n$, $\mathbf{C} \in \mathbb{R}^{n\times n}$ is symmetric and $\boldsymbol{x} \in \mathbb{R}^n$.

Such linear least squares problems with an absolute quadratic constraint are very common in computer vision, e. g., in shape fitting to scattered data [FPF99]. In this context, $\mathbf{A}$ is often called the *design matrix* and $\mathbf{C}$ the *constraint matrix*. A common constraint is unit length of $\boldsymbol{x}$, i. e., $\boldsymbol{x}^\mathsf{T}\boldsymbol{x} = 1$, described by the constraint matrix $\mathbf{C} = \mathbf{I}_n$.

Lagrange multipliers[4] are a commonly used tool in mathematical optimization to convert constrained linear least squares problems into unconstrained problem instances.

Equation (C.18) can be transformed into an unconstrained minimization problem by introducing a new variable λ called the Lagrange multiplier and considering the Lagrange function defined by:

$$f(\boldsymbol{x}, \lambda) = \|\mathbf{A}\boldsymbol{x}\|^2 + \lambda(1 - \boldsymbol{x}^\mathsf{T}\mathbf{C}\boldsymbol{x}) \tag{C.19}$$

If $\boldsymbol{x}$ is a solution of the constrained minimization problem eq. (C.18), then there exists a Lagrange multiplier λ so that $f(\boldsymbol{x}, \lambda)$ is a stationary point of f, i. e., the derivative of f with respect to $\boldsymbol{x}$ and λ is required to disappear:

$$\nabla_{\boldsymbol{x}} f = 2\mathbf{A}^\mathsf{T}\mathbf{A}\boldsymbol{x} - 2\lambda\mathbf{C}\boldsymbol{x} = \mathbf{0} \tag{C.20}$$

and

$$\nabla_\lambda f = 1 - \boldsymbol{x}^\mathsf{T}\mathbf{C}\boldsymbol{x} = 0 \tag{C.21}$$

Hence, stationary points of f satisfy the constraint $\boldsymbol{x}^\mathsf{T}\mathbf{C}\boldsymbol{x} = 1$ and are

[4] named after the French mathematician and astronomer Joseph-Louis Lagrange (*1736, †1812)

solutions of the general eigenvalue problem:

$$\mathbf{A}^\intercal \mathbf{A}\boldsymbol{x} = \lambda \mathbf{C}\boldsymbol{x} \tag{C.22}$$

where $\mathbf{S} := \mathbf{A}^\intercal \mathbf{A}$ is a symmetric, positive semi-definite $n \times n$ matrix, called the *scatter matrix*.

Unit length constraint For $\mathbf{C} = \mathbf{I}$, i. e., the unit length constraint for $\boldsymbol{x}$, eq. (C.22) defines a standard eigenvalue problem:

$$\mathbf{A}^\intercal \mathbf{A}\boldsymbol{x} = \lambda \boldsymbol{x} \tag{C.23}$$

Equation (C.23) can be solved either algebraically or by numerical methods for eigenvalue decomposition such as QR factorization.
At each stationary point of f, the vector $\boldsymbol{x}$ must be a unit eigenvector of $\mathbf{S}$ and the Lagrange multiplier λ is the associated eigenvalue. Considering the properties of $\mathbf{S}$, we obtain n possible solutions $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ and the associated eigenvalues $\lambda_1, \ldots, \lambda_n$ have to be real non-negative values. W.l.o.g. we assume that the solutions are ordered according to $\lambda_1 \leqslant \ldots \leqslant \lambda_n$.

For each solution $\boldsymbol{x}_i$, we obtain from $\|\boldsymbol{x}_i\| = 1$ and eq. (C.23):

$$\|\mathbf{A}\boldsymbol{x}_i\|^2 = \boldsymbol{x}_i^\intercal \mathbf{A}^\intercal \mathbf{A}\boldsymbol{x}_i = \lambda_i \boldsymbol{x}_i^\intercal \boldsymbol{x}_i = \lambda_i \tag{C.24}$$

Since we seek to minimize $\|\mathbf{A}\boldsymbol{x}\|^2$, the solution of eq. (C.18) subject to the unit length constraint is given by the unit vector $\boldsymbol{x}_1$ corresponding to the smallest eigenvalue λ_1.

Absolute quadratic constraint The general eigenvalue problem eq. (C.22) with $\mathbf{C} + \mathbf{I}$ can be solved using numerical methods such as the generalized Schur decomposition (QZ algorithm) or Cholesky factorization of the constraint matrix.
Another way to solve it is to transform it to the equivalent standard eigenvalue problem $\mathbf{C}^{-1}(\mathbf{A}^\intercal \mathbf{A})\boldsymbol{x} = \lambda \boldsymbol{x}$. If $\mathbf{C}$ has not full rank, we consider

instead the eigenvalue problem $\mathbf{S}'\boldsymbol{x}' = \lambda'\boldsymbol{x}'$ with respect to the matrix

$$\mathbf{S}' = \left(\mathbf{C}'^{-1}(\mathbf{G} - \mathbf{H}\,\mathbf{J}^{-1}\mathbf{H}^\mathsf{T})\right) \tag{C.25}$$

with

$$\mathbf{S} = \begin{pmatrix} \mathbf{G} & \mathbf{H} \\ \mathbf{H}^\mathsf{T} & \mathbf{J} \end{pmatrix} \quad \text{and} \quad \mathbf{C} = \begin{pmatrix} \mathbf{C}' & \mathbf{0}_{k\times\ell} \\ \mathbf{0}_{\ell\times k} & \mathbf{0}_{\ell\times\ell} \end{pmatrix} \tag{C.26}$$

resulting from reordering $\mathbf{S}$ and $\mathbf{C}$ appropriately. $\mathbf{G}$ and $\mathbf{C}'$ are symmetric $k \times k$ matrices respectively while $\mathbf{H} \in \mathbb{R}^{k\times\ell}$ and $\mathbf{J} \in \mathbb{R}^{\ell\times\ell}$ with $k + \ell = n$ such that $\mathbf{C}'$ has full rank. The solution $\boldsymbol{x}$ for the original general eigenvalue problem is derived from the solution $\boldsymbol{x}'$ of the standard eigenvalue problem:

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}' \\ -(\mathbf{J}^{-1}\mathbf{H}^\mathsf{T})\boldsymbol{x}' \end{pmatrix} \tag{C.27}$$

For further details on linear least squares with absolute quadratic constraint we refer the reader to [SH12].

General quadratic constraint Consider a constrained linear least squares problem of the form:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 \quad \text{subject to } \|\mathbf{C}\boldsymbol{x} - \boldsymbol{d}\|^2 = 1 \tag{C.28}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$ with $m \geqslant n$, $\boldsymbol{b} \in \mathbb{R}^m$, $\mathbf{C} \in \mathbb{R}^{k\times n}$, $\boldsymbol{d} \in \mathbb{R}^k$ and $\boldsymbol{x} \in \mathbb{R}^n$.

Gander [Gan81] describes a method to solve eq. (C.28) using Lagrange multipliers which is similar in spirit to the method described above. The solution is derived from the Lagrange function:

$$f(\boldsymbol{x}, \lambda) = \|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 + \lambda(\|\mathbf{C}\boldsymbol{x} - \boldsymbol{d}\|^2 - 1) \tag{C.29}$$

by solving the following normal equations that result from $\nabla_{\boldsymbol{x}} f = 0$:

$$(\mathbf{A}^\mathsf{T}\mathbf{A} + \lambda\mathbf{C}^\mathsf{T}\mathbf{C})\boldsymbol{x} = \mathbf{A}^\mathsf{T}\boldsymbol{b} + \lambda\mathbf{C}^\mathsf{T}\boldsymbol{d} \tag{C.30}$$

subject to $\|\mathbf{C}\boldsymbol{x} - \boldsymbol{d}\|^2 = 1$. This also involves to solve a general eigenvalue problem of the form $\mathbf{A}^\intercal\mathbf{A}\boldsymbol{x} = \mu\mathbf{C}^\intercal\mathbf{C}\boldsymbol{x}$ similar to eq. (C.22).

C.3.3 Nonlinear Least Squares

Consider an unconstrained nonlinear least squares problem of the form:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{y}\|^2 \tag{C.31}$$

where $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$ is a nonlinear, continuously differentiable function with $m > n$ and $\boldsymbol{y} \in \mathbb{R}^m$ is a vector of observations

Finding a global minimizer of eq. (C.31) is in general very hard [MNT04]. Hence, it is common to reduce the problem to finding a local minimizer in the vicinity of a starting point $\boldsymbol{x}_0 \in \mathbb{R}^n$. In the following we will refer to local minimization only.

A standard tool for the numerical solution of eq. (C.31) is the *Levenberg-Marquardt algorithm*[5] [Mor78]. The Levenberg-Marquardt algorithm is an iterative method to solve eq. (C.31) locally that combines the steepest descent and the Gauss-Newton method. We use the MATLAB function `lsqnonlin` [Mat13] and the C/C++ implementation `levmar` [Lou05] of the Levenberg-Marquardt algorithm based on the lecture notes of Madsen et al. [MNT04] in our work. A brief description of the Levenberg-Marquardt algorithm following these is given in the next section, however we refer the reader to the literature mentioned above for a more comprehensive treatment.

Levenberg-Marquardt algorithm Given a starting point $\boldsymbol{x}_0 \in \mathbb{R}^n$, the nonlinear least squares problem eq. (C.31) is approximated in each iteration step $k \in \mathbb{N}_{\geqslant 0}$ by replacing $\boldsymbol{f}(\boldsymbol{x})$ by its first-order Taylor series expansion

[5] named after the American statisticians Kenneth Levenberg who published the algorithm first in 1944 and Donald W. Marquardt who rediscovered it in 1963 [Mor78]

around $\boldsymbol{x}_k \in \mathbb{R}^n$ and estimating the optimal parameter update $\Delta\boldsymbol{x}$:

$$\min_{\Delta\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{f}_k + \mathbf{J}_k\Delta\boldsymbol{x} - \boldsymbol{y}\|^2 \tag{C.32}$$

where $\boldsymbol{f}_k = \boldsymbol{f}(\boldsymbol{x}_k)$ and $\mathbf{J}_k = \frac{\partial f}{\partial \boldsymbol{x}}(\boldsymbol{x}_k)$ is the Jacobian matrix of $\boldsymbol{f}$ evaluated at the current parameter estimate $\boldsymbol{x}_k$. Equation (C.32) describes a linear least squares problem that can be solved with respect to $\Delta\boldsymbol{x}$ from the normal equations:

$$\mathbf{J}_k^\intercal\mathbf{J}_k\Delta\boldsymbol{x} = \mathbf{J}_k^\intercal\Delta\boldsymbol{y}_k \tag{C.33}$$

where $\Delta\boldsymbol{y}_k = \boldsymbol{y} - \boldsymbol{f}_k$ is the current prediction error. The symmetric, positive semi-definite $n \times n$ matrix $\mathbf{H}_k = \mathbf{J}_k^\intercal\mathbf{J}_k$ is an approximation to the Hessian matrix of the objective function $g(\boldsymbol{x}) = \|\boldsymbol{f}(\boldsymbol{x})\|^2$, i. e., the matrix of second-order derivatives $\frac{\partial g}{\partial x_i \partial x_j}$ near $\boldsymbol{x}_k$. Its inverse $\Sigma_{\boldsymbol{x}_k} = \mathbf{H}_k^{-1}$ can be interpreted as an approximation to the parameter covariance matrix near $\boldsymbol{x}_k$.

The Levenberg-Marquardt algorithm solves a modified version of eq. (C.33), the *augmented normal equations*:

$$(\mathbf{J}_k^\intercal\mathbf{J}_k + \mu\mathbf{D}_k)\Delta\boldsymbol{x} = \mathbf{J}_k^\intercal\Delta\boldsymbol{y}_k \tag{C.34}$$

with $\mathbf{D}_k = \operatorname{diag}(\mathbf{J}_k^\intercal\mathbf{J}_k)$ for a given $\mu \in \mathbb{R}_{>0}$ called the *dampening term*.[6]

In each iteration, eq. (C.34) is solved repeatedly for increasing dampening terms μ until a solution $\Delta\boldsymbol{x}_k$ is found that leads to a reduction of the updated error term, i. e., $g(\boldsymbol{x}_k + \Delta\boldsymbol{x}_k) < g(\boldsymbol{x}_k)$. Afterwards, the dampening term is decreased and $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta\boldsymbol{x}_k$ is used as the starting point for the next iteration step.

The algorithms continues until one of the following conditions is met:

- $\|\Delta\boldsymbol{x}_k\| < \rho_{\Delta x}\|\boldsymbol{x}_k\|$ for a predefined ratio $\rho_{\Delta x} \in \mathbb{R}_{>0}$
- $\|\mathbf{J}_k^\intercal\Delta\boldsymbol{y}_k\| < \varepsilon_{\mathbf{J}}$ for a predefined threshold $\varepsilon_{\mathbf{J}} \in \mathbb{R}_{>0}$
- $g(\boldsymbol{x}_k) < \varepsilon_g$ for a predefined threshold $\varepsilon_g \in \mathbb{R}_{>0}$

[6] In the original algorithm proposed by Levenberg, $\mathbf{D}_k = \mathbf{I}$ was used which is also implemented in `levmar` [Lou05]. Marquardt replaced the dampening term by $\mathbf{D}_k = \operatorname{diag}(\mathbf{J}_k^\intercal\mathbf{J}_k)$ in order to improve convergence [Mar63]. Similar suggestions were made by Moré [Mor78].

$\triangleright$ $k = k_{\max}$ for a predefined maximal iteration number $k_{\max} \in \mathbb{N}_{>0}$

The final solution after k iterations is given by $\boldsymbol{x}^* = \boldsymbol{x}_k$.

Note that the adaptive dampening allows the algorithm to alternate between the Gauss-Newton method for smaller μ when the current solution is close to a local minimum and a gradient descent approach for larger μ when the solution is far from the correct one. This increases the robustness of the algorithm with respect to the Gauss-Newton method while speeding up convergence with respect to the steepest descent approach. For further instructions on the choice of the thresholds $\rho_{\Delta \boldsymbol{x}}$, $\varepsilon_{\mathbf{J}}$, ε_g and the actual update of the dampening term μ, see [MNT04].

Weighted Levenberg-Marquardt algorithm Consider that the objective function in eq. (C.31) has the form:

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{y}\|^2_{\boldsymbol{\Sigma}_y} \tag{C.35}$$

where $\boldsymbol{y} \in \mathbb{R}^m$ is a vector of observations with given covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{y}} \in \mathbb{R}^{m \times m}$ and the Mahalanobis norm $\|\cdot\|_{\boldsymbol{\Sigma}_y}$ is defined as in eq. (C.15).

The Levenberg-Marquardt algorithm can be easily modified in order to solve eq. (C.35) by replacing the normal equations in eq. (C.33) with the following weighted normal equations:

$$\mathbf{J}_k^\intercal \boldsymbol{\Sigma}_y^{-1} \mathbf{J}_k \Delta \boldsymbol{x} = \mathbf{J}_k^\intercal \boldsymbol{\Sigma}_y^{-1} \Delta \boldsymbol{y}_k \tag{C.36}$$

The parameter covariance matrix of the converged solution $\boldsymbol{x}^*$ is estimated from the measurement covariance matrix via uncertainty propagation as described in A.5:

$$\boldsymbol{\Sigma}_{\boldsymbol{x}^*} = (\mathbf{J}_k^\intercal \boldsymbol{\Sigma}_y^{-1} \mathbf{J}_k)^{-1} \tag{C.37}$$

In comparison, the parameter covariance matrix of $\boldsymbol{x}^*$ for the unweighted nonlinear least squares problem (C.32) is given by:

$$\boldsymbol{\Sigma}_{\boldsymbol{x}^*} = (\mathbf{J}_k^\intercal \mathbf{J}_k)^{-1} \mathbf{J}_k^\intercal \boldsymbol{\Sigma}_{\boldsymbol{y}} \mathbf{J}_k (\mathbf{J}_k^\intercal \mathbf{J}_k)^{-1} \tag{C.38}$$

Sparse Levenberg-Marquardt algorithm The solution of the normal equations is in general computationally very demanding when the number of parameters for the objective function $\boldsymbol{f}$ is large. Fortunately, many large-scale problems that arise in practice exhibit a certain lack of interdependence between parameters, leading to Jacobian matrices with a rather sparse structure. For such problems it is advisable to use specific implementations of the Levenberg-Marquardt algorithm that take care of the sparsity pattern of the Jacobian matrix. In our work we use the C/C++ implementation `sparseLM` [Lou10] for this purpose.

C.3.4 Constrained Nonlinear Least Squares

Consider a constrained nonlinear least squares problem of the form:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{y}\|^2 \quad \text{subject to } \boldsymbol{x} \in \Omega_X \tag{C.39}$$

where $\Omega_X \subset \mathbb{R}^n$ is the problem-specific manifold of valid parameters.

Constrained Levenberg-Marquardt algorithm Given that Ω_X can be described by a quadratic parameter constraint

$$\Omega_X = \{\boldsymbol{x} \in \mathbb{R}^n \mid \|\mathbf{C}\boldsymbol{x} - \boldsymbol{d}\|^2 = 1\}$$

where $\mathbf{C} \in \mathbb{R}^{k\times n}$, $\boldsymbol{d} \in \mathbb{R}^k$ as in eq. (C.28), the Levenberg-Marquardt algorithm can be modified to solve eq. (C.39) by replacing the unconstrained linear least squares problem in eq. (C.32) with the following constrained linear least squares problem:

$$\min_{\Delta\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{f}_k + \mathbf{J}_k\Delta\boldsymbol{x} - \boldsymbol{y}\|^2 \quad \text{subject to } \|\mathbf{C}(\boldsymbol{x} + \Delta\boldsymbol{x}) - \boldsymbol{d}\|^2 = 1 \tag{C.40}$$

Equation (C.40) can be solved with the methods described in C.3.2. However, this modification requires to solve a general eigenvalue problem multiple times during each iteration step so the computational effort is increased significantly.

Projected Levenberg-Marquardt algorithm Kanzow [KYF04] describes a similar method based on the Levenberg-Marquardt algorithm for nonlinear least squares with general parameter constraints that also involves the solution of a constrained optimization problem in each iteration step. However, he proposes another more efficient and flexible algorithm denoted as *Projected Levenberg-Marquardt method* in the same paper that is shown to have essentially the same convergence properties.

The major modification of the Projected Levenberg-Marquardt algorithm is to project the updated parameter vector $\boldsymbol{x}_k + \Delta\boldsymbol{x}$ onto Ω_X in each iteration step after eq. (C.34) has been solved for $\Delta\boldsymbol{x}$,

$$\boldsymbol{x}_{k+1} = \text{proj}(\boldsymbol{x}_k + \Delta\boldsymbol{x}_k)$$

where $\text{proj} : \mathbb{R}^n \to \Omega_X$ describes the projection onto Ω_X.

Constrained Nonlinear Programming Furthermore, constrained nonlinear least squares problems can be solved with state-of-the-art methods for *Constrained Nonlinear Programming*, i. e., optimization of a scalar function $g : \mathbb{R}^n \to \mathbb{R}$ subject to general nonlinear equality and inequality constraints $\boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{0}$ and $\boldsymbol{d}(\boldsymbol{x}) \geqslant \boldsymbol{0}$, such as *Sequential Quadratic Programming* or interior point methods with barrier functions [BHN99] for large scale problems.[7]

C.4 Robust Parameter Estimation

In the previous sections we considered the problem of estimating the parameters of a model that optimally fits the given observations under the assumption that the observed data consists of *inliers* only. Inliers are considered as data points that can be described appropriately by some model instance although they might be subject to noise. However, model fitting is heavily impacted by the presence of *outliers*, i. e., data points that do not conform to the model due to gross measurement errors or inapproprioate models. To eliminate the influence of outliers on the

[7] Both algorithms are implemented in the MATLAB function `fmincon` [Mat13].

estimation process, robust estimation techniques must be employed such as the *Random Sample Consensus* algorithm (RANSAC) proposed by Fischler & Bolles [FB81].

Random Sample Consensus RANSAC is a non-deterministic meta algorithm for robust estimation in the presence of outliers that is commonly used in computer vision. Given measurements $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m \in \mathrm{Y}$ containing m_{out} outliers and a model-specific loss function $c : \Theta \times \mathrm{Y} \to \mathbb{R}_{\geqslant 0}$ evaluating the error of an observation $\boldsymbol{y}$ with respect to the model instance given by parameters $\boldsymbol{\theta} \in \Theta$, the RANSAC algorithm proceeds iteratively as follows:

- ▷ Initialize the best solution $\boldsymbol{\theta}^* := \varnothing$ and best inlier count $M^* := 0$.
- ▷ Draw a random subset of k measurements $\boldsymbol{y}_{j_1}, \ldots, \boldsymbol{y}_{j_k}$ from the data. Consider this set as hypothetical inliers.[8]
- ▷ Compute model parameters $\boldsymbol{\theta}$ fitting the hypothetical inliers $\boldsymbol{y}_{j_1}, \ldots, \boldsymbol{y}_{j_k}$.
- ▷ Count all inliers with respect to the estimated model and the loss function c, i. e., $M = \|\{\boldsymbol{y}_j \mid c(\boldsymbol{\theta}, \boldsymbol{y}_j) \leqslant \varepsilon\}\|$ for a predefined error threshold $\varepsilon \in \mathbb{R}_{\geqslant 0}$. The set of inliers for $\boldsymbol{\theta}$ is denoted as the *consensus set*.
- ▷ If $M > M^*$, update the best solution $\boldsymbol{\theta}^* := \boldsymbol{\theta}$ and $M^* := M$.
- ▷ Repeat until $M \geqslant M_{\min}$ for a predefined threshold $M_{\min} \in \mathbb{N}$ or the number of iterations exceeds $N_{\max}$ for a predefined threshold $N_{\max} \in \mathbb{N}$.
- ▷ The final model can be improved by reestimating it from the consensus set for $\boldsymbol{\theta}^*$.

If $M_{\min} < m$, the RANSAC algorithm is also denoted as *greedy* since it terminates as soon as the first valid solution is found.

The minimal number of iterations N needed to find a random sample set consisting of inliers only with some anticipated probability $P \in (0, 1)$ can

[8] The number of samples should be chosen as the minimal number from which a solution for $\boldsymbol{\theta}$ can be retrieved in order to provide the largest probability to draw a subset without outliers.

be derived from the number of samples k to determine a solution and the assumed inlier ratio $\hat{\rho}_{\text{in}} \approx \frac{m-m_{\text{out}}}{m} \in (0,1)$:

$$(1-\hat{\rho}_{\text{in}}^k)^N \leqslant 1-P \quad \Rightarrow \quad N \geqslant \frac{\log(1-P)}{\log(1-\hat{\rho}_{\text{in}}^k)} \tag{C.41}$$

X84 outlier rejection rule In order to increase the robustness of parameter estimation, outlier rejection rules such as *X84* can be applied [Ham+86].

Under the hypothesis that the residuals $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_m)$ of the error function are Gaussian distributed, a robust estimator for the standard deviation $\sigma_{\boldsymbol{\epsilon}}$ is given by the *Median Absolute Deviation (MAD)*:

$$\text{MAD}(\boldsymbol{\epsilon}) = \text{median}_{i=1}^{m}\{|\epsilon_i - \text{median}_{j=1}^{n}\{\epsilon_j\}|\} \tag{C.42}$$

and

$$\sigma_{\boldsymbol{\epsilon}} = \frac{1}{\Phi^{-1}(\frac{3}{4})}\ \text{MAD}(\boldsymbol{\epsilon}) \approx 1.4826\ \text{MAD}(\boldsymbol{\varepsilon})$$

The X84 rule rejects values with $\epsilon > 3.5\,\sigma_{\boldsymbol{\epsilon}}$ since the probability for this case is $< 0.1\%$ assuming Gaussian distribution.

Bibliography

[Aga+11] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Richard Szeliski. "Building rome in a day". In: *Communications of the ACM* 54.10 (2011), pp. 105–112.

[AHE01] Nicolas Andreff, Radu Horaud, and Bernard Espiau. "Robot hand-eye calibration using structure-from-motion". In: *International Journal of Robotics Research* 20.3 (Mar. 2001), pp. 228–248.

[AHE99] Nicolas Andreff, Radu Horaud, and Bernard Espiau. "Online hand-eye calibration". In: *2nd International Conference on 3D Digital Imaging and Modeling*. Ottawa, Canada, Oct. 1999, pp. 430–436.

[Alt86] Simon L. Altmann. Rotations, Quaternions, and Double Groups. Clarendon Press, Oxford, 1986.

[BA00] Patrick T. Baker and Yiannis Aloimonos. "Complete calibration of a multi-camera network". In: *IEEE Workshop on Omnidirectional Vision*. IEEE Computer Society, 2000, pp. 134–141.

[Bai+03] Yohan Baillot, Simon J. Julier, Dennis Brown, and Mark A. Livingston. "A tracker alignment framework for augmented reality". In: *2nd IEEE and ACM International Symposium on Mixed and Augmented Reality*. 2003, pp. 142–150.

[Bar03] Adrien Bartoli. "Towards gauge invariant bundle adjustment: a solution based on gauge dependent damping". In: *Ninth IEEE International Conference on Computer Vision (ICCV'03)*. Vol. 2. Nice, France, Oct. 2003, pp. 760–765.

[Bay03] Eduardo Bayro-Corrochano. "Modeling the 3d kinematics of the eye in the geometric algebra framework". In: *Pattern Recognition* 36.12 (2003), pp. 2993–3012.

[BBD08] Marcel Brückner, Ferid Bajramovic, and Joachim Denzler. "Experimental evaluation of relative pose estimation algorithms". In: *International Conference on Computer Vision Theory and Applications*. Vol. 2. 2008, pp. 431–438.

[BBD09a] Ferid Bajramovic, Marcel Brückner, and Joachim Denzler. "Using common field of view detection for multi camera calibration". In: *Vision Modeling and Visualization*. 2009, pp. 113–120.

[BBD09b] Marcel Brückner, Ferid Bajramovic, and Joachim Denzler. "Geometric and probabilistic image dissimilarity measures for common field of view detection". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 2052–2057.

[BD08] Ferid Bajramovic and Joachim Denzler. "Global uncertainty-based selection of relative poses for multi camera calibration". In: *British Machine Vision Conference*. Vol. 2. 2008, pp. 745–754.

[BDS97] Eduardo Bayro-Corrochano, Konstantinos Daniilidis, and Gerald Sommer. "Hand-eye calibration in terms of motion of lines using geometric algebra". In: *10th Scandinavian Conference on Image Analysis*. 1997, pp. 397–404.

[BGK06] Kostas E. Bekris, Max Glick, and Lydia E. Kavraki. "Evaluation of algorithms for bearing-only SLAM". In: *IEEE International Conference on Robotics and Automation*. 2006, pp. 1937–1943.

[BHN99] Richard H. Byrd, Mary E. Hribar, and Jorge Nocedal. "An interior point algorithm for large-scale nonlinear programming". In: *SIAM Journal on Optimization* 9.4 (1999), pp. 877–900.

[BKD09] Ferid Bajramovic, Michael Koch, and Joachim Denzler. "Experimental comparison of wide baseline correspondence algorithms for multi camera calibration". In: *International Conference on Computer Vision Theory and Applications*. Vol. 2. 2009, pp. 458–463.

[Bla12] Jose-Luis Blanco. A Tutorial on SE(3) Transformation Parameterizations and On-Manifold Optimization. Tech. rep. #012010. ETS Ingeniería Informática, Universidad de Málaga, Aug. 2012.

[Bra00] Gary Bradski. "The OpenCV library". In: *Dr. Dobb's Journal of Software Tools* (2000). URL: http://opencv.org.

[Bro66] Duane C. Brown. "Decentering distortion of lenses". In: *Photogrammetric Engineering* 32.3 (1966), pp. 444–462.

[BT03] Olivier A. Bauchau and Lorenzo Trainelli. "The vectorial parameterization of rotation". In: *Nonlinear Dynamics* 32.1 (2003), pp. 71–92.

[Che91] Homer H. Chen. "A screw motion approach to uniqueness analysis of head-eye geometry". In: *IEEE Conference on Computer Vision and Pattern Recognition*. Maui, HI, United States, June 1991, pp. 145–151.

[CI02] Yaron Caspi and Michal Irani. "Alignment of non-overlapping sequences". In: *International Journal of Computer Vision* 48.1 (2002), pp. 39–51.

[CK88] Jack C. K. Chou and M. Kamel. "Quaternions approach to solve the kinematic equation of rotation of a sensor-mounted robotic manipulator". In: *IEEE International Conference on Robotics and Automation*. Vol. 2. 1988, pp. 656–662.

[CK91] Jack C. K. Chou and M. Kamel. "Finding the position and orientation of a sensor on a robot manipulator using quaternions". In: *International Journal of Robotics Research* 10.3 (June 1991), pp. 240–254.

[Cli+08] Brian Clipp, Jae-Hak Kim, Jan-Michael Frahm, Marc Pollefeys, and Richard I. Hartley. "Robust 6dof motion estimation for non-overlapping, multi-camera systems". In: *IEEE Workshop on Applications of Computer Vision* (Jan. 2008), pp. 1–8.

[Dai+10] Yuchao Dai, Jochen Trumpf, Hongdong Li, Nick Barnes, and Richard I. Hartley. "Rotation averaging with application to camera-rig calibration". In: *9th Asian Conference on Computer Vision – Volume Part II*. Springer-Verlag, Berlin, Heidelberg, 2010, pp. 335–346.

[Dan99] Konstantinos Daniilidis. "Hand-eye calibration using dual quaternions". In: *International Journal of Robotics Research* 18 (1999), pp. 286–298.

[Dav+07] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse. "MonoSLAM: Real-time single camera SLAM". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.6 (2007).

[DB96] Konstantinos Daniilidis and Eduardo Bayro-Corrochano. "The dual quaternion approach to hand-eye calibration". In: *13th International Conference on Pattern Recognition*. Vol. 1. Aug. 1996, pp. 318–322.

[DC03] Fadi Dornaika and Ronald Chung. "Stereo geometry from 3d ego-motion streams". In: *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics* 33.2 (Apr. 2003), pp. 308–323.

[DEE08] Patrick Denis, James H. Elder, and Francisco J. Estrada. "Efficient edge-based methods for estimating manhattan frames in urban imagery". In: *European Conference on Computer Vision*. 2008, pp. 197–210.

[Del12] David H. Delphenich. "The representation of physical motions by various types of quaternions". In: *ArXiv e-prints* (2012). URL: http://adsabs.harvard.edu/abs/2012arXiv1205.4440D.

[DH98] Fadi Dornaika and Radu Horaud. "Simultaneous robot-world and hand-eye calibration". In: *IEEE Transactions on Robotics and Automation* 14.4 (1998), pp. 617–622.

[EG10] Sandro Esquivel and Stefan Gehrig. "Entwicklung eines Kalibrierverfahrens für fahrzeugmontierte Mehrkamerasysteme". Abschlussbericht im Projekt AKTIV-AS, Teilprojekt KAS, Daimler AG. July 2010.

[EK13] Sandro Esquivel and Reinhard Koch. "Structure from motion using rigidly coupled cameras without overlapping views". In: *35th German Conference on Pattern Recognition*. Vol. 8142. Lecture Notes in Computer Science. Saarbrücken, Germany, Sept. 2013, pp. 11–20.

[EKR08] Sandro Esquivel, Reinhard Koch, and Heino Rehse. “Reconstruction of sewer shaft profiles from fisheye-lens camera images”. In: *31st DAGM Symposium on Pattern Recognition*. Vol. 5748. Lecture Notes in Computer Science. Jena, Germany, Sept. 2008, pp. 332–341.

[EM95] Alan Edelman and H. Murakami. “Polynomial roots from companion matrix eigenvalues”. In: *Mathematics of Computation* 64.210 (1995), pp. 763–776.

[EN05] Christopher Engels and David Nistér. “Global uncertainty in epipolar geometry via fully and partially data-driven sampling”. In: *ISPRS Workshop BenCOS: Towards Benchmarking Automated Calibration, Orientation and Surface Reconstruction from Images*. 2005, pp. 17–22.

[Ern+12] Floris Ernst, Lars Richter, Lars Matthäus, Volker Martens, Ralf Bruder, Alexander Schlaefer, and Achim Schweikard. “Non-orthogonal tool/flange and robot/world calibration for realistic tracking scenarios”. In: *International Journal of Medical Robotics and Computer Assisted Surgery* (2012), pp. 1427–1440.

[Esq07] Sandro Esquivel. “‘Calibration of a Multi-Camera Rig from Non-Overlapping Views’”. Diploma thesis. Technische Fakultät, Christian-Albrechts-Universität zu Kiel, 2007.

[EWK07] Sandro Esquivel, Felix Woelk, and Reinhard Koch. “Calibration of a multi-camera rig from non-overlapping views”. In: *29th DAGM Symposium on Pattern Recognition*. Vol. 4713. Lecture Notes in Computer Science. Heidelberg, Germany, Sept. 2007, pp. 82–91.

[FB81] Martin A. Fischler and Robert C. Bolles. “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography”. In: *Communications of the ACM* 24.6 (June 1981), pp. 381–395.

[FH86] Olivier D. Faugeras and Martial Hébert. “The representation, recognition, and locating of 3-d objects”. In: *The International Journal of Robotics Research* 5.27 (1986), pp. 27–52. URL: http://ijr.sagepub.com/content/5/3/27.

[FKK04] Jan-Michael Frahm, Kevin Köser, and Reinhard Koch. "Pose estimation for multi-camera systems". In: *26th DAGM Symposium on Pattern Recognition*. Vol. 3175. Lecture Notes in Computer Science. Tübingen, Germany, Aug. 2004, pp. 286–293.

[FL05] Irene Fassi and Giovanni Legnani. "Hand to sensor calibration: A geometrical interpretation of the matrix equation AX = XB". In: *Journal on Robotics Systems* 22.9 (2005), pp. 497–506.

[FPF99] Andrew W. Fitzgibbon, Maurizio Pilu, and Robert B. Fisher. "Direct least squares fitting of ellipses". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21.5 (1999), pp. 476–480.

[Gan81] Walter Gander. "Least squares with a quadratic constraint". In: *Numerische Mathematik* 36 (1981), pp. 291–307.

[Gao+03] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. "Complete solution classification for the perspective-three-point problem". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25.8 (2003), pp. 930–943.

[Gol+98] Steven Gold, Anand Rangarajan, Chien-Ping Lu, Pappu Suguna, and Eric Mjolsness. "New algorithms for 2d and 3d point matching: Pose estimation and correspondence". In: *Pattern Recognition* 38.8 (1998), pp. 1019–1031.

[Ham+86] Frank R. Hampel, Elvezio M. Ronchetti, Peter J. Rousseeuw, and Werner A. Stahel. Robust Statistics: The Approach Based on Influence Functions. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, Inc., 1986.

[Ham44] William Rowan Hamilton. "On quaternions, or on a new system of imaginaries in algebra". In: *Philosophical Magazine* 25.3 (1844), pp. 489–495.

[HAT11] Richard I. Hartley, Khurrum Aftab, and Jochen Trumpf. "L_1 rotation averaging using the Weiszfeld algorithm". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2011, pp. 3041–3048.

[HD95] Radu Horaud and Fadi Dornaika. “Hand-eye calibration”. In: *International Journal of Robotics Research* 14.3 (1995), pp. 195–210.

[HDK01] Robert L. Hirsh, Guilherme N. DeSouza, and Avinash C. Kak. “An iterative solution to the hand-eye and base-world calibration problem”. In: *IEEE International Conference on Robotics and Automation*. 2001, pp. 2171–2176.

[Hel+11] Jan Heller, Michal Havlena, Akihiro Sugimoto, and Tomás Pajdla. “Structure-from-motion based hand-eye calibration using L_∞ minimization”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2011, pp. 3497–3503.

[HHP12] Jan Heller, Michal Havlena, and Tomás Pajdla. “A branch-and-bound algorithm for globally optimal hand-eye calibration”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 1608–1615.

[HK08] Kristine Haase and Reinhard Koch. “AR Binocular: Augmented Reality System for nautical navigation”. In: *Workshop on Mobile and Embedded Interactive Systems*. 2008, pp. 295–300.

[HMR08] Joel A. Hesch, Anastasios I. Mourikis, and Stergios I. Roumeliotis. “Mirror-based extrinsic camera calibration”. In: *Workshop on the Algorithmic Foundations of Robotics*. Guanajuato, Mexico, Dec. 2008.

[HO06] Matthew J. Harker and Paul L. O’Leary. “First order geometric distance (The myth of Sampsonus)”. In: *British Machine Vision Conference*. 2006, pp. 10.1–10.10.

[Hor87] Berthold K. P. Horn. “Closed-form solution of absolute orientation using unit quaternions”. In: *Journal of the Optical Society of America A* 4.4 (1987), pp. 629–642.

[HS88] Chris Harris and Mike Stephens. “A combined corner and edge detector”. In: *4th Alvey Vision Conference*. 1988, pp. 147–151.

[HS97] Richard I. Hartley and Peter Sturm. “Triangulation”. In: *Computer Vision and Image Understanding* 68.2 (1997), pp. 146–157.

[Huy09] Du Q. Huynh. "Metrics for 3D rotations: Comparison and analysis". In: *Journal of Mathematical Imaging and Vision* 35.2 (2009), pp. 155–164.

[HZ04] Richard I. Hartley and Andrew Zisserman. Multiple View Geometry in Computer Vision. 2nd Edition. Cambridge University Press, 2004.

[Iki00] Milan Ikits. Coregistration of Pose Measurement Devices Using Nonlinear Least Squares Parameter Estimation. Tech. rep. UUCS-00-018. Virtual Reality Laboratory, University of Utah, 2000.

[Jay04] Christopher Jaynes. "Multi-view calibration from planar motion trajectories". In: *Image and Vision Computing* 22.7 (2004), pp. 535–550.

[KB06] Juho Kannala and Sami S. Brandt. "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (2006), pp. 1335–1340.

[KC06] Jae-Hean Kim and Myung Jin Chung. "Absolute motion and structure from stereo image sequences without stereo correspondence and analysis of degenerate cases". In: *Pattern Recognition* 39.9 (2006), pp. 1649–1661.

[Kim+10] Sin-Jung Kim, Mun-Ho Jeong, Joong-Jae Lee, Ji-Yong Lee, Kang-Geon Kim, Bum-Jae You, and Sang-Rok Oh. "Robot head-eye calibration using the minimum variance method". In: *IEEE International Conference on Robotics and Biomimetics*. 2010, pp. 1446–1451.

[KLH10] Jae-Hak Kim, Hongdong Li, and Richard I. Hartley. "Motion estimation for nonoverlapping multicamera rigs: Linear algebraic and L_∞ geometric solutions". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.6 (June 2010), pp. 1044–1059.

[KM07] Georg Klein and David Murray. “Parallel tracking and mapping for small AR workspaces”. In: *6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. Nov. 2007, pp. 225–234.

[KNS13] Moritz Knorr, Wolfgang Niehsen, and Christoph Stiller. “Online extrinsic multi-camera calibration using ground plane induced homographies”. In: *IEEE Intelligent Vehicles Symposium*. June 2013, pp. 236–241.

[Kum+08] Ram Krishan Kumar, Adrian Ilie, Jan-Michael Frahm, and Marc Pollefeys. “Simple calibration of non-overlapping cameras with a mirror”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2008.

[KYF04] Christian Kanzow, Nobuo Yamashita, and Masao Fukushima. “Levenberg-Marquardt methods for constrained nonlinear equations with strong local convergence properties”. In: *Computational and Applied Mathematics* 172 (2004), pp. 375–397.

[LA09] Manolis I. A. Lourakis and Antonis A. Argyros. “SBA: A software package for generic sparse bundle adjustment”. In: *ACM Transactions on Mathematical Software* 36.1 (2009), pp. 1–30. URL: http://www.ics.forth.gr/~lourakis/sba.

[Lam+07] Bernhard Lamprecht, Stefan Rass, Simone Fuchs, and Kyandoghere Kyamakya. “Extrinsic camera calibration for an onboard two-camera system without overlapping field of view”. In: *IEEE Intelligent Transportation Systems Conference*. Sept. 2007, pp. 265–270.

[LC95] Ying-Cherng Lu and Jack C. K. Chou. “Eight-space quaternion approach for robotic hand-eye calibration”. In: *IEEE International Conference on Systems, Man and Cybernetics*. Vol. 4. 1995, pp. 3316–3321.

[Léb+10] Pierre Lébraly, Omar Ait-Aider, Eric Royer, and Michel Dhome. “Calibration of non-overlapping cameras – application to vision-based robotics”. In: *British Machine Vision Conference*. 2010, pp. 10.1–10.12.

[Léb+11] Pierre Lébraly, Eric Royer, Omar Ait-Aider, Clément Deymier, and Michel Dhome. "Fast calibration of embedded non-overlapping cameras". In: *IEEE International Conference on Robotics and Automation*. Shanghai, China, May 2011, pp. 221–227.

[Li+13] Bo Li, Lionel Heng, Kevin Köser, and Marc Pollefeys. "A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2013, pp. 1301–1307.

[Liu+13] Qianzhe Liu, Junhua Sun, Yuntao Zhao, and Zhen Liu. "Calibration method for geometry relationships of nonoverlapping cameras using light planes". In: *Optical Engineering* 52.7 (2013), pages.

[LK81] Bruce D. Lucas and Takeo Kanade. "An iterative image registration technique with an application to stereo vision". In: *7th International Joint Conference on Artificial Intelligence*. Vol. 2. 1981.

[LLZ14] Zhen Liu, Fengjiao Li, and Guangjun Zhang. "An external parameter calibration method for multiple cameras based on laser rangefinder". In: *Measurement* 47 (2014), pp. 954–962.

[LM08] Ronghua Liang and Jianfei Mao. "Hand-eye calibration with a new linear decomposition algorithm". In: *Journal of Zhejiang University - Science A* 9 (2008), pp. 1363–1368.

[LMF09] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. "EPnP: An accurate $O(n)$ solution to the PnP problem". In: *International Journal of Computer Vision* 81.2 (2009).

[LN09] Quoc V. Le and Andrew Y. Ng. "Joint calibration of multiple sensors". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Oct. 2009, pp. 3651–3658.

[Lon81] Hugh Christopher Longuet-Higgins. "A computer algorithm for reconstructing a scene from two projections". In: *Nature* 293.5828 (1981), pp. 133–135.

[Lou05] Manolis I. A. Lourakis. A Brief Description of the Levenberg-Marquardt Algorithm Implemented by `levmar`. Tech. rep. Institute of Computer Science, Foundation for Research and Technology – Hellas (FORTH), Feb. 2005. URL: `http://www.ics.forth.gr/~lourakis/levmar`.

[Lou10] Manolis I. A. Lourakis. "Sparse non-linear least squares optimization for geometric vision". In: *European Conference on Computer Vision*. Vol. 2. 2010, pp. 43–56. URL: `http://www.ics.forth.gr/~lourakis/sparseLM`.

[Low04] David G. Lowe. "Distinctive image features from scale-invariant keypoints". In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110.

[Mar63] Donald W. Marquardt. "An algorithm for least-squares estimation of nonlinear parameters". In: *Journal of the Society for Industrial and Applied Mathematics* 11.2 (1963), pp. 431–441.

[Mat13] MATLAB. Release 2013a. The MathWorks, Inc., Natick, MA, United States, 2013.

[May93] Stephen J. Maybank. Theory of Reconstruction from Image Motion. Springer-Verlag, Berlin, 1993.

[MBF09] Jochen Meidow, Christian Beder, and Wolfgang Förstner. "Reasoning with uncertain points, straight lines, and straight line segments in 2d". In: *Journal of Photogrammetry and Remote Sensing* 64.2 (2009), pp. 125–139.

[MEB04] Dimitrios Makris, Tim Ellis, and James Black. "Bridging the gaps between cameras". In: *IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 2. 2004, pp. 205–210.

[MH00] Henrik Malm and Anders Heyden. "A new approach to hand-eye calibration". In: *15th International Conference on Pattern Recognition*. Vol. 1. 2000, pp. 525–529.

[MHJ10] Jianfei Mao, Xianping Huang, and Li Jiang. "A flexible solution to AX = XB for robot hand-eye calibration". In: *10th WSEAS International Conference on Robotics, Control and Manufacturing Technology*. 2010, pp. 118–122.

[MNT04] K. Madsen, H. B. Nielsen, and O. Tingleff. Methods for Non-Linear Least Squares Problems. 2nd Edition. Informatics and Mathematical Modelling, Technical University of Denmark, 2004.

[Mor78] Jorge J. Moré. "The Levenberg-Marquardt algorithm: Implementation and theory". In: *Lecture Notes in Mathematics* 630 (1978), pp. 105–116.

[MP07] Daniel Martinec and Tomás Pajdla. "Robust rotation and translation estimation in multiview reconstruction". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2007, pp. 1–8.

[Muh11] Daniel Muhle. "'Gegenseitige Orientierung von Mehrkamerasystemen mit nicht überlappendem Sichtfeld'". PhD thesis. Fakultät für Bauingenieurwesen und Geodäsie, Gottfried Wilhelm Leibniz Universität Hannover, 2011.

[Nis04a] David Nistér. "A minimal solution to the generalised 3-point pose problem". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2004.

[Nis04b] David Nistér. "An efficient solution to the five-point relative pose problem". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2004).

[Pag10] Frank Pagel. "Calibration of non-overlapping cameras in vehicles". In: *IEEE Intelligent Vehicles Symposium*. June 2010, pp. 1178–1183.

[Pag12a] Frank Pagel. "'Kalibrierung mobiler Multikamerasysteme mit disjunkten Sichtfeldern'". PhD thesis. Fakultät für Informatik, Karlsruher Institut für Technologie, 2012.

[Pag12b] Frank Pagel. "Motion adjustment for extrinsic calibration of cameras with non-overlapping views". In: *9th Conference on Computer and Robot Vision*. 2012, pp. 94–100.

[Pau+12] Karl Pauwels, Matteo Tomasi, Javier Díaz, Eduardo Ros, and Marc M. Van Hulle. "A comparison of FPGA and GPU for real-time phase-based optical flow, stereo, and local image features". In: *IEEE Transactions on Computers* 61.7 (2012).

[Ple03] Robert Pless. “Using many cameras as one”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2003, pp. 587–593.

[PM94] Frank C. Park and Bryan J. Martin. “Robot sensor calibration: Solving AX = XB on the Euclidean group”. In: *IEEE Transactions on Robotics and Automation* 10.5 (Oct. 1994), pp. 717–721.

[PW10] Frank Pagel and Dieter Willersinn. “Motion-based online calibration for non-overlapping camera views”. In: *13th International IEEE Conference on Intelligent Transportation Systems*. 2010, pp. 843–848.

[PW11] Frank Pagel and Dieter Willersinn. “Extrinsic camera calibration in vehicles with explicit ground estimation”. In: *8th International Workshop on Intelligent Transportation*. 2011, pp. 1–6.

[RDD04] Ali Rahimi, Brian Dunagan, and Trevor Darrell. “Simultaneous calibration and tracking with a network of non-overlapping sensors”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 1. 2004, pp. 187–194.

[Rém+97] Sandrine Rémy, Michel Dhome, Jean-Marc Lavest, and Nadine Daucher. “Hand-eye calibration”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vol. 2. 1997, pp. 1057–1065.

[RHH08] Volker Rodehorst, Matthias Heinrichs, and Olaf Hellwich. “Evaluation of relative pose estimation methods for multi-camera setups”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XXXVII (2008), pp. 135–140.

[RLR11] Antonio L. Rodríguez, Pedro E. López de Teruel, and Alberto Ruiz. “GEA optimization for live structureless motion estimation”. In: *IEEE International Conference on Computer Vision*. 2011, pp. 715–718.

[RPK11] Thomas Ruland, Tomás Pajdla, and Lars Krüger. "Global optimization of extended hand-eye calibration". In: *IEEE Intelligent Vehicles Symposium*. 2011, pp. 740–745.

[RPK12] Thomas Ruland, Tomás Pajdla, and Lars Krüger. "Globally optimal hand-eye calibration". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 1035–1042.

[Rus96] Dave Rusin. *N*-Dim Spherical Random Number Drawing. The Mathematical Atlas. 1996. URL: http://www.math.niu.edu/~rusin/known-math/96/sph.rand.

[SÅ04] Henrik Stewénius and K. Åström. "Hand-eye calibration using multilinear constraints". In: *Asian Conference on Computer Vision*. Jeju, Korea, Jan. 2004.

[SA89] Yiu Cheung Shiu and Shaheen Ahmad. "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX = XB". In: *IEEE Transactions on Robotics and Automation* 5.1 (1989), pp. 16–29.

[SBK08] Ingo Schiller, Christian Beder, and Reinhard Koch. "Calibration of a PMD camera using a planar calibration object together with a multi-camera setup". In: *XXIth International Society for Photogrammetry and Remote Sensing Congress*. Vol. XXXVII. Beijing, China, 2008, pp. 297–302. URL: http://www.mip.informatik.uni-kiel.de/tiki-index.php?page=Calibration.

[SCL09] Yongduek Seo, Young-Ju Choi, and Sang Wook Lee. "A branch-and-bound algorithm for globally optimal calibration of a camera-and-rotation-sensor system". In: *12th IEEE International Conference on Computer Vision*. 2009, pp. 1173–1178.

[SEH12] Mili Shah, Roger D. Eastman, and Tsai Hong. "An overview of robot-sensor calibration methods for evaluation of perception systems". In: *Workshop on Performance Metrics for Intelligent Systems*. ACM Press, Mar. 2012, pp. 15–20.

[SH06] Klaus H. Strobl and Gerd Hirzinger. "Optimal hand-eye calibration". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Beijing, China, Oct. 2006, pp. 4647–4653.

[SH12] René Schöne and Tobias Hanning. “Least squares problems with absolute quadratic constraints”. In: *Journal of Applied Mathematics* (2012), pp. 1–12.

[SMS06] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. “A flexible technique for accurate omnidirectional camera calibration and structure from motion”. In: *IEEE International Conference of Vision Systems*. 2006.

[SN01] Jochen Schmidt and Heinrich Niemann. “Using quaternions for parametrizing 3-D rotations in unconstrained nonlinear optimization”. In: *6th International Fall Workshop of Vision, Modeling, and Visualizations*. 2001, pp. 399–406.

[SSS06] Noah Snavely, Steven M. Seitz, and Richard Szeliski. “Photo Tourism: Exploring photo collections in 3D”. In: *ACM Transactions on Graphics (SIGGRAPH Proceedings)* 25.3 (2006), pp. 835–846.

[ST94] Jianbo Shi and Carlo Tomasi. “Good features to track”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 1994, pp. 593–600.

[SVN03] Jochen Schmidt, Florian Vogt, and Heinrich Niemann. “Robust hand-eye calibration of an endoscopic surgery robot using dual quaternions”. In: *25th DAGM Symposium on Pattern Recognition*. Vol. 2781. Lecture Notes in Computer Science. Magdeburg, Germany, 2003, pp. 548–556.

[SVN04] Jochen Schmidt, Florian Vogt, and Heinrich Niemann. “Vector quantization based data selection for hand-eye calibration”. In: *Vision, Modeling, and Visualization*. Ed. by B. Girod, M. Magnor, and H.-P. Seidel. Aka/IOS Press, Berlin, Amsterdam, 2004, pp. 21–28.

[SVN05] Jochen Schmidt, Florian Vogt, and Heinrich Niemann. “Calibration-free hand-eye calibration: A structure-from-motion approach”. In: *27th DAGM Symposium on Pattern Recognition*. Vol. 3663. Lecture Notes in Computer Science. Vienna, Austria, Sept. 2005.

[SWL05] Fanhuai Shi, Jianhua Wang, and Yuncai Liu. "An approach to improve online hand-eye calibration". In: *2nd Iberian Conference on Pattern Recognition and Image Analysis*. 2005, pp. 647–655.

[Sze10] Richard Szeliski. Computer Vision: Algorithms and Applications. Springer-Verlag, New York, 2010.

[TC04] Lorenzo Trainelli and Alessandro Croce. "A comprehensive view of rotation parametrization". In: *European Congress on Computational Methods in Applied Sciences and Engineering*. 2004, pp. 1–17.

[Ter+12] George Terzakis, Phil Culverhouse, Guido Bugmann, Sanjay Sharma, and Robert Sutton. A Recipe on the Parameterization of Rotation Matrices for Non-Linear Optimization Using Quaternions. Tech. rep. MIDAS.SMSE.2012.TR.004. Marine, Industrial Dynamic Analysis School of Marine Science, and Engineering, Plymouth University, 2012.

[TK91] Carlo Tomasi and Takeo Kanade. Detection and Tracking of Point Features. Tech. rep. CMU-CS-91-132. Carnegie Mellon University, Apr. 1991.

[TL89] Roger Y. Tsai and Reimar K. Lenz. "A new technique for fully autonomous and efficient 3d robotics hand/eye calibration". In: *IEEE Transactions on Robotics and Automation* 5.3 (1989), pp. 345–358.

[TM08] Tinne Tuytelaars and Krystian Mikolajczyk. "Local invariant feature detectors: A survey". In: *Foundations and Trends in Computer Graphics and Vision* 3.3 (2008), pp. 177–280.

[Tor97] Philip H. S. Torr. "An assessment of information criteria for motion model selection". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 1997, pp. 47–52.

[Tri+00] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. "Bundle adjustment – A modern synthesis". In: *Vision Algorithms: Theory and Practice*. Vol. 1883. Lecture Notes in Computer Science. 2000, pp. 298–372.

[TŠ13] Radim Tyleček and Radim Šára. "Spatial pattern templates for recognition of objects with regular structure". In: *35th German Conference on Pattern Recognition*. Vol. 8142. Lecture Notes in Computer Science. Saarbrücken, Germany, Sept. 2013, pp. 364–374.

[Tsa87] Roger Y. Tsai. "A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses". In: *International Journal of Robotics and Automation* 3 (1987), pp. 323–344.

[Ull79] Shimon Ullman. The Interpretation of Visual Motion. MIT Press, Cambridge, MA, United States, 1979.

[VMW08] Jaume Vergés-Llahí, Daniel Moldovan, and Toshikazu Wada. "A new reliability measure for essential matrices suitable in multiple view calibration". In: *3rd International Conference on Computer Vision Theory and Applications*. Vol. 1. Madeira, Portugal, 2008, pp. 114–121.

[WAH98] Guo-Qing Wei, Klaus Arbter, and Gerd Hirzinger. "Active self-calibration of robotic eyes and hand-eye relationships with model identication". In: *IEEE International Conference on Robotics and Automation*. Vol. 14. 1998, pp. 158–166.

[Wan92] Ching-Cheng Wang. "Extrinsic calibration of a vision sensor mounted on a robot". In: *IEEE Transactions on Robotics and Automation* 2.8 (1992), pp. 161–175.

[WH92] Juyang Weng and Thomas S. Huang. "Complete structure and motion from two monocular sequences without stereo correspondence". In: *International Conference on Pattern Recognition*. 1992, pp. 651–654.

[Wu11] Changchang Wu. VisualSFM: A Visual Structure from Motion System. 2011. URL: http://ccwu.me/vsfm.

[Zha00] Zhengyou Zhang. "A flexible new technique for camera calibration". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2000), pp. 1330–1334.

[Zha99] Zhengyou Zhang. "Flexible camera calibration by viewing a plane from unknown orientations". In: *7th IEEE International Conference on Computer Vision*. Vol. 1. 1999, pp. 666–673.

[Zhu97] Hanqi Zhuang. "A note on *Hand-Eye Calibration*". In: *International Journal of Robotics Research* 16.5 (1997), pp. 725–727.

[ZL06] Zijian Zhao and Yuncai Liu. "Hand-eye calibration based on screw motions". In: *International Conference on Pattern Recognition* 3 (2006), pp. 1022–1026.

[ZL08] Zijian Zhao and Yuncai Liu. "Integrating camera calibration and hand-eye calibration into robot vision". In: *7th World Congress on Intelligent Control and Automation*. 2008, pp. 5721–5727.

[ZR91] Hanqi Zhuang and Zvi S. Roth. "Comments on *Calibration of Wrist-Mounted Robotic Sensors by Solving Homogeneous Transform Equations of the Form AX = XB*". In: *IEEE Transactions on Robotics and Automation* 7.6 (1991), pp. 877–878.

[ZRS94] Hanqi Zhuang, Zvi S. Roth, and Raghavan Sudhakar. "Simultaneous robot-world and tool-flange calibration by solving homogeneous transformation equations of the form AX = YB". In: *IEEE Transaction on Robotics and Automation* 10.4 (1994).

[ZS93] Hanqi Zhuang and Yiu Cheung Shiu. "A noise-tolerant algorithm for robotic hand-eye calibration with or without sensor orientation measurement". In: *IEEE Transactions on System, Man, and Cybernetics* 23.4 (1993), pp. 1168–1175.

[ZSL05] Jing Zhang, Fanhuai Shi, and Yuncai Liu. "An adaptive selection of motion for online hand-eye calibration". In: *AI 2005: Advances in Artificial Intelligence*. Ed. by S. Zhang and R. Jarvis. Vol. 3809. Lecture Notes in Computer Science. Springer-Verlag, Berlin, Heidelberg, 2005, pp. 520–529.