Informed Data Selection For Dynamic Multi-Camera Clusters

by

Arun Das

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2018

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner	Dr. Timothy Bretl Associate Professor
Supervisor(s)	Dr. Steven L. Waslander Associate Professor
Internal Member	Dr. Soo Jeon Associate Professor
Internal Member	Dr. James Tung Assistant Professor
Internal-external	Dr. David Wang Professor

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Chapter 4 of this thesis, "Information Theoretic Next-Best-View", contains content which has been incorporated within a publication that was co-authored by myself, Mr. Jason Rebello, a Ph.D. candidate who is also at the Waterloo Autonomous Vehicles Lab, and our supervisor, Dr. Steven L. Waslander. Specifically, the entropy based next-bext view formulation has been incorporated into the paper,

J. Rebello, **A. Das** and S.L. Waslander, "Autonomous Active Calibration of a Dynamic Camera Cluster using Next-Best-View." 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Both myself and Mr. Rebello contributed equally to the work. I am the sole author of all other thesis contents.

Abstract

Traditional multi-camera systems require a fixed calibration between cameras to provide the solution at the correct scale, which places many limitations on its performance. This thesis investigates the calibration of dynamic camera clusters, or DCCs, where one or more of the cluster cameras is mounted to an actuated mechanism, such as a gimbal or robotic manipulator. Our novel calibration approach parameterizes the actuated mechanism using the Denavit-Hartenberg convention, then determines the calibration parameters which allow for the estimation of the time varying extrinsic transformations between the static and dynamic camera frames. A degeneracy analysis is also presented, which identifies redundant parameters of the DCC calibration system.

In order to automate the calibration process, this thesis also presents two information theoretic methods which selects the optimal calibration viewpoints using a next-best-view strategy. The first strategy looks at minimizing the entropy of the calibration parameters, while the second method selects the viewpoints which maximize the mutual information between the joint angle input and calibration parameters.

Finally, the effective selection of key-frames is also an essential aspect of robust visual navigation algorithms, as it ensures metrically consistent mapping solutions while reducing the computational complexity of the bundle adjustment process. To that end, we propose two entropy based methods which aim to insert key-frames that will directly improve the system's ability to localize. The first approach inserts key-frames based on the cumulative point entropy reduction in the existing map, while the second approach uses the predicted point flow discrepancy to select key-frames which best initialize new features for the camera to track against in the future.

The DCC calibration methods are verified in both simulation and using physical hardware consisting of a 5-DOF Fanuc manipulator and a 3-DOF Aeryon Skyranger gimbal. We demonstrate that the proposed methods are able to achieve high quality calibrations using RMSE pixel error metrics, as well as through analysis of the estimator covariance matrix. The key-frame insertion methods are implemented within the Multi-Camera Parallel Mapping and Tracking (MCPTAM) framework, and we confirm the effectiveness of these approaches using high quality ground truth collected using an indoor positioning system.

Acknowledgements

I would like to thank my supervisor, Dr. Steven L. Waslander for his immense guidance, patience, and support throughout this journey. His constant encouragement and enthusiasm was paramount in helping me to cultivate the knowledge and experience gained over my time spent at the lab.

I would like to thank all my colleagues at the WAVELab, who made my time here both fun and extremely memorable. Thank you to the SLAM team, who were willing to work extra hard when facing tight deadlines. A special thanks to Jason and Nav, who were always up for blowing off some steam and making terrible bets on sports teams.

I would like to especially acknowledge the work of Jason Rebello, who contributed ideas to chapter 4 of this thesis, and Sally Hui, who helped a great deal with the data collection for the experiments conducted in Chapters 3 and 4.

Finally, I would like to immensely thank my parents, whose hard work, dedication, and sacrifice, gave me the opportunity to pursue my goals, and my loving wife, Niyati, and puppy dog, Noodle, for their limitless support.

Dedication

For Niyati,

and my family,

Mom, Dad, Mitun, Luc, Leena, Sonia, and Noodle.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Visual Simultaneous Localization and Mapping (SLAM) algorithms have been proposed for use in a wide array of applications, ranging from inspection and surveillance [1, 2], to outdoor and Martian exploration [3, 4]. Capable visual SLAM solutions are especially important where only intermittent measurements to inertial positioning systems, such as GPS, are available, or the required accuracy of the localization solution is greater than what GPS alone can provide. Although very high accuracy localization and mapping is possible using LIDAR, the high cost of such sensors remains a barrier to wide range deployment. In contrast, the continuous innovation and improvements in cell phone and mobile device technology has made it possible to obtain high quality, low cost, and low power camera sensors off the shelf, which has further made vision-based navigation an appealing approach for the autonomous mobile robotics community.

Monocular visual navigation systems consist of only one camera, and are limited by only being able to observe the environment from a single view-point at a time. As a result, the localization quality can degrade rapidly if parts of the environment are occluded, or rapid motion of the camera causes significant viewpoint changes. Furthermore, without additional sensing, a monocular system is only able to resolve its motion up to a scale

factor [5], limiting its use for navigation and control applications where the true motion at scale is required. To that end, *multi-camera* systems, consisting of two or more cameras, have become a popular approach for visual navigation, as depending on the configuration, they are able to mitigate the shortcomings of monocular systems.

Many different configurations of multi-camera systems are used for localization and mapping, in a broad range of applications. Stereo cameras are a widely used approach, where two forward facing cameras are calibrated and rectified so that epipoloar search can be easily performed between the two images [6]. The large overlapping field-of-view (FOV) allows for the calculation of corresponding feature point depths in every pair of collected images, which has led to the development of many stereo based approaches for visual odometry and SLAM in applications such as planetary exploration [3], autonomous driving [7], and aerial robots [8].

As an alternative, more general multi-camera clusters (MCCs) provide advantages to stereo based configurations, as the addition of an arbitrary number of cameras with multiple viewpoints allows for more robust tracking and mapping operations in three respects. First, the ability of the MCC to take measurements over a wide FOV helps with camera localization robustness by better constraining the motion solution, and preventing feature starvation by consistently tracking features over longer durations and over large viewpoint changes. Second, a wider FOV allows for robust map generation and point triangulation by collecting more feature measurements across the whole environment. Finally, so long as the extrinsic calibration is known, multi-camera systems do not require overlap in the FOV to resolve the scale of the solution [9].

Although capable of performing accurate localization in a variety of environments, a major disadvantage of all multi-camera systems to date is that they require a fixed calibration between cameras to provide the solution at the correct scale. The fixed extrinsic calibration of the cluster places many limitations on MCC performance. First, any camera

cluster must be re-calibrated if a new configuration is required, which is especially tedious and time consuming when the vehicle is deployed in the field. Second, since the MCC is fixed to the vehicle frame, the observation viewpoints of the cameras are highly dependent on the vehicle motion. The coupling of the vehicle motion and camera observation viewpoints is especially problematic if the vehicle undergoes motions which make the vision solution degenerate, or if the vehicle motion results in the camera cluster observing areas of low texture where only poor feature measurements are possible. Finally, many systems, such as UAVs, cannot use the existing gimballed camera payload to assist with the visual navigation. Since current state of the art multi-camera solutions require fixed calibrations between cameras, the gimballed camera is generally only employed for data collection purposes.

Typical vision-based navigation and calibration algorithms are composed of two parts: the *front-end*, and the *back-end*. One of the responsibilities of the front-end is processing the camera stream and extracting features from the images. For the purpose of vision-based navigation and calibration, features are points in the image which can be reliably identified from multiple viewpoints, and may correspond to specific structures in the image such as points, edges or objects. Using the local neighbourhood of pixels surrounding the feature, a *descriptor* for the feature can be generated. The descriptor quantifies the appearance of the feature, and is used to match the feature between multiple viewpoints. Measurements to features in the environment are recorded from multiple view points, known as *key-frames*, and the collection of measurements are passed to the back-end. The back-end performs an optimization procedure called *bundle adjustment*, which is a nonlinear optimization that estimates both the structure of the environment, and the motion of the camera. The robotics and computer vision research community have developed many navigation systems which perform real-time estimation, using vision only techniques [10, 11, 12, 13, 14, 15, 16], as well methods which integrate additional inertial or GNSS sensing [17, 18]. Calibration algorithms operate similarly, except instead optimize over unknown physical

system model parameters of the camera cluster, such as spatial sensor offsets and lens distortion parameters [19, 20]. Successful estimation requires careful attention to both the back-end and the front-end components, as the measurements selected by the front-end directly impact the ability of the back-end to find a valid solution.

In order for any estimation system to operate successfully, the state or parameters of the system must be uniquely recoverable given the measurable outputs. In the context calibration, this means that the data contained in the camera input stream must contain sufficient information in order to fully recover the calibration parameters. Furthermore, the estimation solution should be unique, since convergence to an arbitrary set of calibration parameters, which may also agree with the measurements yet do not reflect the physical state of the system, would likely result in failure of system when used in a visual navigation task. Existing works have investigated the *degeneracy* of similar systems in order to evaluate the limitations of parameter and state estimate recovery [10, 11]. Although it is clear that estimation systems require front-end measurements which contain sufficient information, for state-of-the-art calibration methods, measurement selection remains a widely manual process, thus the true information content of the measurements is unknown.

Manual measurement collection for calibration is limiting in two main aspects. First, providing sufficient information to accurately estimate the calibration parameters requires collecting data over the full measurement space of the system. Defining a set of collection points is not obvious, especially as the system's measurement space increases in complexity, and poor data selection will result in an inaccurate or degenerate calibration. Although the measurement space can be sampled to concurrently select data for batch processing, such an approach becomes cumbersome and impractical as the dimension of the measurement space grows, and further does not guarantee that the selected samples will provide the sufficient excitation required for accurate calibration. Second, manual selection of the

measurements precludes automating the calibration. Automatic calibration is emerging as a crucial functionality for state-of-the-art applications, as robots deployed in the field and those which are mass produced require accurate calibration that can be performed by non-experts without human intervention in order to improve robustness.

The effective selection of key-frames is also an essential aspect of robust visual navigation algorithms, as it ensures metrically consistent mapping solutions while reducing the computational complexity of the bundle adjustment process. Both the accuracy and computational complexity of visual SLAM solutions can be significantly improved by selectively including only the most informative new measurements at each stage of the algorithm. However, determining the information content of measurements before inclusion in the tracking and mapping processes is an open problem.

In this thesis, information theoretic approaches are used to perform *informed data selection* for calibration and navigation of dynamic camera clusters (DCC). This thesis will present and study the time-varying calibration required in order to enable the use of a DCC with existing vision-based tracking and SLAM systems, and will also perform a *degeneracy analysis* of the DCC calibration process, in order to identify the circumstances under which the calibration fails. In order to ensure sufficient measurement excitation of the system, we shall also investigate an *automatic* method of calibration for DCCs, which is autonomous, and works on the principle of next-best-view [21] to select the actuator configurations such that selected measurements locally minimize parameter uncertainty at each iteration. Finally, in this thesis, we shall discuss a method to perform *informed key-frame selection*, which scores potential key-frames based on their predicted information content with respect to the existing map.

The DCC based calibration will allow, for the first time, vehicles such as drones to integrate gimballed cameras into their existing visual navigation pipelines. As the gimballed cameras are typically capable of collecting very high quality imagery, this integration will

greatly improve navigation accuracy, and will facilitate the development of new *active* vision algorithms that perform viewpoint selection of the dynamic camera. The proposed informed key-frame selection methods continually assess the the input image streams and will greatly improve navigation robustness in challenging environments. Advances in informed data selection are critical for robust and reliable navigation, and the work presented in this thesis will further enable the next generation of robots to perform increasingly difficult and demanding missions.

1.1 Related Works

1.1.1 Calibration

Much work has been done on multi-sensor calibration problems for robotics applications. Existing approaches have been able to perform high quality extrinsic calibrations between camera and IMU sensors [19], as well as perform observability analysis to determine when the calibration fails [20]. Precise extrinsic calibration between cameras and 3D LIDAR sensors have also been achieved using both gradient based methods [22], and information theoretic approaches [23]. The camera-to-camera calibration problem is also well studied, as it is essential for MCC based SLAM systems.

Current camera-to-camera calibration approaches typically use fiducial markers to generate common observations between cameras [9, 24, 25], though unsupervised methods which use natural features in the environment from pre-existing maps or on-line SLAM solutions have also provided good results [26]. Although there has been significant work done in the area of camera to camera calibration, we have not found any existing results for camera to camera calibration through an actuated mechanism.

The hand-eye calibration problem, from the field of robotic manipulators, consists of

computing the relative position and orientation between the motion frame of a mechanism, and a sensor which is rigidly mounted to the mechanism. The main focus for the hand-eye problem is simultaneously estimating the relative translation between a camera mounted to a robotic manipulator and the manipulator's end effector frame, as well as the transformation between the manipulator's base frame and the camera's motion base frame [27]. Originally developed for camera to manipulator calibration, the hand-eye problem also describes other calibration tasks, such as camera to odometry calibration [28], and the calibration between a camera cluster and a motion tracking system [9]. Although the hand-eye problem is similar to the dynamic MCC calibration problem, the hand-eye calibration assumes that the parameters of the mechanism's forward kinematics (such as the DH parameters), are known, whereas our dynamic MCC calibration requires estimation of these parameters.

The class of calibration methods related to our problem is known as *kinematic calibration*, and seeks to refine the forward kinematic parameters of robotic manipulators in order to improve overall end effector positioning performance. Generally, the kinematic parameters are optimized by comparing the motion of the end effector to the predicted motion of the mechanism given the forward kinematic parameters and the joint angles. External measurement of the end effector can be performed using a variety of methods, such as using co-ordinate measurement machines (CMM)[29] and externally mounted theodolites [30]. However the cost of such measurement equipment is typically quite high, which has motivated the use of low-cost camera based solutions for kinematic calibration.

Camera based kinematic calibration for manipulators consists of taking relative measurements between a camera mounted on the manipulator and a fiducial target in the environment, or mounting the target on the manipulator and placing a static camera in the environment [31]. For example, kinematic calibration has been performed by using a laser dot target with known scale to estimate the DH-parameters of two pan-tilt units with

attached cameras [32]. The work most similar to ours uses both visual sensors and LIDAR mounted to a robotic manipulator to identify fiducial targets in the environment, and then simultaneously estimate both the DH-parameters of the manipulator the the poses of the targets [33]. Although such approaches use the camera and fiducial marker to perform the calibration, the estimated parameters only describe the forward kinematics of the manipulator with respect to the *robot base*, whereas calibration of the dynamic MCC, for use in a SLAM problem, requires knowledge of the *camera to camera* calibration, which only include the mechanism's kinematic parameters as part of the total transformation between camera co-ordinate frames. Furthermore, existing approaches in the literature do not discuss degeneracy of the calibration procedure

1.1.2 Observability and Degeneracy Analysis

Observability and degeneracy analysis are fundamental to understanding if the desired calibration parameters can be successfully estimated using the observations measured by the system. For nonlinear systems, the *local weak observability* of the system can be determined by checking the observability rank condition at any point in the system's state space [34]. This involves checking the column rank of a matrix containing the partial derivatives with respect to the system states, for increasing orders of Lie derivatives of the measurement model with respect the the system dynamics. When the matrix has full column rank, the system is locally weakly observable about that point. Although primarily used to analyse control systems, local weak observability of nonlinear systems has also successfully been analyzed for visual-inertial odometry Systems [35], as well as sensor-to-sensor self calibration problems [20]. In order to perform observability analysis, we require the system state to have dynamics which evolve with time. For robot systems which do not naturally possess dynamics, constant velocity or constant acceleration 6-DOF motion models are often employed to circumvent this requirement. However, for such cases,

the true characteristics of the robot are ignored, and the analysis only holds for vehicle trajectories which are sufficiently close to the assumed motion model.

The concept of system degeneracy analysis is closely related to observability for non-linear systems. For systems which do not possess any dynamics, only the zeroth-order Lie derivatives are non-zero. Thus, evaluating the observability rank condition is equivalent to checking the rank of the measurement Jacobian matrix. With such an approach, degenerate configurations have successfully been identified for multi-camera systems employing both geometric solution strategies [36, 37, 38, 39, 40] and nonlinear optimization [5]. In this work, we follow an approach similar to that found in [5], in order to identify superfluous calibration parameters that result in the degeneracy of the DCC calibration problem.

1.1.3 Sensitivity Analysis

Observability and degeneracy analysis of systems only provides a binary answer as to whether the system is observable. In many estimation applications, it is also advantageous to understand the *degree of observability* of the system, which attempts to quantify how close a system is to being observable.

To study the degree of observability, existing works discuss a local observability index and a local estimation condition number which operates on the local observability Gramian. The approach has been used to measure the degree of observability or unobservability of a system, and has been successfully applied to determining the optimal locations from which to collect observation for vortex flow strength estimation [41]. The use of the local observability index has also been extended and applied to trajectory optimization of a UAV for sensor self-calibration [42]. The work predicts the quality of state estimation based on the vehicle's ego-motion, and accounts for dynamics, measurements and nonlinear observability of the system.

The approaches which explore the local observability index of a system aim to classify the system on a scale ranging from *poorly* to *strongly* observable [42]. The system is strongly observable if the output changes significantly when the state is marginally perturbed. A state with this property is robust to measurement noise and it is highly distinguishable within some proximity where this property holds. Conversely, a state that leads to a small change in the output, even though the state value was extensively perturbed, is defined as poorly observable. Since the local observability index examines the state-to-measurement relationship through the system measurement model, it is unclear from the existing work as to how the local observability index is related to the state covariance matrix, which is typically used within least squares estimation frameworks to assess the estimation quality.

Other approaches examine the local sensitivity of the estimation error function by analyzing the error function Jacobian with respect to the state and measurement data. Such an approach has been successfully applied to the sensitivity analysis of an electrical power distribution system, in order to identify the parameters and measurements that are most influential to the estimation process [43]. However, the analysis presented requires taking derivatives with respect to individual measurements, which is cumbersome and intractable for typical visual navigation and calibration algorithms, where the process can look at a very large number of visual features as part of the overall estimation task.

Global sensitivity methods looks at performing perturbation experiments which examine how small changes to the system inputs affect its output, over the permissible set of input parameter ranges. These approaches then use scatter plots, regression techniques, or the conditional variance factoring of parameters to determine the strength of correlation between the input and output [44]. Such techniques work well when the nature of the system process does not allow one to take derivatives of the output function with respect to the parameters of interest but may require manual inspection of data plots, making automation infeasible. Furthermore, methods to perform efficient analysis of the coupling

effects between parameters is unclear, as the techniques would require one to perform extensive trials over multiple combinations of parameters.

Generally, state-of-the art sensitivity analysis approaches also only examine deterministic systems. Recently, some researchers have looked at using information theoretic tools to study sensitivity analysis on stochastic systems [45], which requires the characterization of the relationship between the input parameters and the output *distribution*. Specifically, these approaches seek to quantify how much changing one or more of the inputs changes the distribution of the output, or in other words, how sensitive the output distribution is to changes in the inputs. Mutual information can be used to quantify this relationship for estimation purposes, as the mutual information quantifies how much learning the value of a measurement affects the uncertainty in the distribution of estimation parameters [45].

1.1.4 Information Theory Applied to SLAM

Information theoretic data selection techniques have been employed in other areas of the visual SLAM problem. *Active SLAM* algorithms attempt to generate control inputs in order to produce vehicle motions which will best collect informative sensor measurements. LIDAR based approaches have been proposed, which perform active localization through the entropy reduction of discrete occupancy grid maps [46]. The approaches discussed in [47, 48, 49] perform active EKF based SLAM, and formulated information gain metrics based on the innovation covariance matrix maintained by the EKF. The methods proposed in [50, 51, 52, 53] also used information-theoretic formulations, and focused on using mutual information between points to improve feature matching and map partitioning. It is evident that the information-theoretic framework offers valuable methods to quantify information, and can be applied to other visual-navigation components such as active vision and key frame insertion.

1.1.5 Active Vision and Active Calibration

Active vision systems [54] have the advantage of being able to manipulate the viewpoint of a camera in order to obtain maximum information from the environment, and are particularly useful in application such as Visual SLAM, where occlusions and limited field of view are prevailing factors [55]. These systems have also been applied to calibration problems, such as determination of the intrinsic parameters of the camera lens model [56].

Foviated vision, and *gaze selection* systems, selectively view areas of the environment by actively moving cameras which are mounted on actuated mechanisms, or by selecting image stream sub-regions from cameras which are statically mounted. Existing foviated platforms are designed to either individually actuate a pair of forward facing cameras [57], or manipulate the orientation of statically calibrated multi-camera clusters [58, 59], generally to aid with object identification, localization, or precise fixation tasks.

Existing gaze selection approaches analyze the image stream to look for areas of the environment with high saliency [59, 60], or regions which contain the most prominent visual features [61]. However, these approaches do not investigate how salient or prominent image regions affect robot-specific tasks such as localization or path-planning.

Some researchers have investigated information theoretic measures, which seek to maximize the task performance of humanoid robots by actively deriving appropriate view directions for the robot's individual cameras over a planning horizon [58]. However, these approaches mainly focus on the movement of individual cameras based on their image streams, and forgo treatment of the foviated camera set as a single unit which exhibits a time-varying extrinsic calibration between cameras.

Next-Best-View (NBV), in general, is the process of determining the next best camera location from which to collect measurements, in order to maximize an information metric that is specific to the task [21]. NBV approaches have been successfully applied to a va-

riety of applications, such as visual servoing [62], 3D reconstruction [63], and monitoring complex industrial processes [64]. In order to formulate the NBV problem, many existing approaches discretize the configuration space, and select the next-best-view from a finite set of possible configurations. For example, NBV 3D mapping of a cluttered environment has been performed by discretizing the desired space to be mapped into voxels, then sampling a fixed number of locations in order to determine the next best view for accurate reconstruction [65]. Assisted intrinsic camera calibration has also been accomplished by generating a discrete set of fiducial target poses, then suggesting the optimal target poses from the set which result in a high quality calibration [56]. Although promising, the efficacy of these existing approaches is heavily dependent on the discretization strategy.

Using active vision and next-best-view concepts, active calibration of the hand-in-eye problem has been previously attempted. However, the existing approaches use a set of predefined motions for measurement collection, and determine some of hand-in-eye transformations using additional, non-camera-based sensing [66, 67, 68, 69, 70]. Closely related to DCC calibration, an extended kinematic calibration has been proposed, where the intrinsic camera parameters, kinematic parameters, and hand-eye relationship are simultaneously estimated using a point on the manipulator that is tracked by a camera through a set of predefined joint configurations [71]. Existing active calibration approaches use a set of *predefined* movements, while this thesis will propose a method which actively calculates and selects the next best pose of the camera that results in the maximum reduction in parameter uncertainty.

1.1.6 Key-frame Selection

Visual mapping techniques often use key-frames in order to reduce the problem size of the batch optimization. Existing approaches, such as Parallel Tracking and Mapping (PTAM) generally insert key-frames based on point triangulation baselines that vary with the depth

to the feature location being estimated [72]. For example, points with large scene depth are better estimated using wide triangulation baselines. The co-visibility of features can also be used to insert key-frames which maximize the total number of common feature measurements over a specific subset of the map [73]. Another selection heuristic inserts key-frames whenever the number of currently tracked points falls below a user defined threshold [74]. Note that the heuristics can be combined, as is the case with Multiple Camera PTAM (MCPTAM) [10]. However, the overall key-frame insertion strategy remains ad-hoc, and the approach is generally refined through manual tuning. Furthermore, these heuristics attempt to insert key-frames in order to maintain the map integrity, yet do not directly attempt to minimize the uncertainty in the map. The approach detailed in [75] presents a method for variable baseline stereo, where the triangulation baseline is selected to minimize the modeled stereo depth error. Although the approach could be adopted to operate with key-frames, the formulation does not allow one to consider measurements from more than two view-points, nor does it consider the camera motion or initialization of new points in the map. The work discussed in [76] performs key-frame selection to reduce content redundancy in the measurements. However, the approach operates off-line and is not suitable for real-time deployment on a field robot.

1.2 Contributions

This work aims to apply informed data selection techniques to camera based navigation for autonomous robots. The contributions claimed in this thesis are as follows.

DCC Calibration Formulation: Multi-camera clusters used for visual SLAM assume a fixed calibration between the cameras, which places many limitations on its performance, and directly excludes all configurations where a camera in the cluster is mounted to a moving component. In this work, we present a novel calibration method for *dynamic* camera

clusters (DCC), where one or more of the cluster cameras is mounted to an actuated mechanism, such as a gimbal or robotic manipulator. The calibration approach parametrizes the actuated mechanism using the Denavit-Hartenberg convention, then determines the calibration parameters which allow for the estimation of the time varying extrinsic transformations between camera frames.

DCC Calibration Degeneracy Analysis: This thesis will present an analytical degeneracy study of the DCC calibration system, and identify the conditions required for successful parameter estimation. We shall demonstrate that a subset of the estimation parameters are redundant, and their inclusion in the calibration process results in a degenerate system that cannot be calibrated.

DCC Calibration using Next-Best-View: The baseline DCC calibration approach relies on manual selection of the joint angles, and requires sufficient measurement excitation from different actuator configurations to ensure accurate calibration results. To that end, a novel active vision approach for DCC calibration is presented, which directly reduces the parameter uncertainty by selecting calibration measurements using an information theoretic next-best-view policy. The presented system automatically selects the next-best-view for the calibration by determining the optimal actuator inputs which minimize the predicted covariance of the extrinsic calibration parameters.

Key-frame Selection: Two novel entropy based key-frame selection methods which aim to accomplish two different tasks are proposed. First, we formulate an approach which seeks to select key-frames based on the *expected cumulative point entropy reduction (CPER)* in the currently existing map. The CPER method chooses key-frames which are the most likely to provide measurements which maximize the reduction in the uncertainty in the currently existing map points. Second, we propose an approach which seeks to select key-frames which are expected to initialize the most favourable *new* feature points, given predicted camera motion over a finite time horizon. The new points are evaluated using

the *point pixel flow discrepancy (PPFD)* between the currently existing map points and the predicted location of new points triangulated from multiple views.

1.3 Thesis Organization

The remainder of this thesis is organized as follows:

- Chapter 2 presents background information on the co-ordinate frames and notation used in this document, information theoretic tools such as entropy and mutual information, the Denavit-Hartenberg convention which will be used for DCC parameterization, and a summary of other techniques from both the robotics and computer vision communities which provide the fundamentals for nonlinear least squares estimation methods.

- Chapter 3 formulates the calibration process for a dynamic camera clusters, and details the use of the Denavit-Hartenberg convention to parameterize the extrinsic calibration between the optical centers of a static and dynamic camera, as a function of the actuated mechanism input. The chapter also presents an analytical degeneracy study of the DCC calibration system, and identifies the conditions required for successful parameter estimation. Experimental results from both simulation study and data collected using a 3-DOF UAV gimbal and a 5-DOF robot manipulator are presented and demonstrate the success of the proposed calibration approach using a re-projection error metric.

- Chapter 4 formulates the information theoretic next-best-view measurement approach, which seeks to select actuator configurations which directly reduce the uncertainty of the calibration parameters. The novel approach investigates two formulations: one which seeks to reduce the entropy of the estimated parameter covariance

matrix, and another which aims to optimally find the actuator configurations which maximize the mutual information between the extrinsic calibration parameters and mechanism joint angles. The next-best-view approaches are demonstrated using both simulation study and data collected using a 3-DOF UAV gimbal and a 5-DOF robot manipulator.

- Chapter 5 presents two novel approaches for key-frame selection which systematically determine the best key-frames for insertion into a point-feature based map. The first approach, inserts key-frames based on the cumulative point entropy reduction (CPER) in the existing map, while the second approach, uses the predicted point flow discrepancy (PPFD) to select key-frames which best initializes new features for the camera to track against in the future. Both approaches are implemented within the Multi-Camera Parallel Mapping and Tracking framework, and we demonstrate the effectiveness of the proposed methods using ground truth data collected using an indoor positioning system.

- Chapter 6 draws conclusions for the thesis and provides several suggestions for future research directions in this area.

Chapter 2

Background

2.1 Frames and Notation

A co-ordinate frame is defined using the notation $\mathcal{F}_{\mathtt{a}}$, where the provided subscript is the name of the frame. Suppose we are given two frames, $\mathcal{F}_{\mathtt{a}}$ and $\mathcal{F}_{\mathtt{b}}$. Then, a vector quantity is denoted as ${}^{\mathtt{a}}t^{\mathtt{a:b}}$, where the right superscript denotes the tail to head direction of the vector, and the left superscript denotes the frame in which the vector is expressed. Thus, ${}^{\mathtt{a}}t^{\mathtt{a:b}}$ can be described as a vector from the origin of $\mathcal{F}_{\mathtt{a}}$ to the origin of $\mathcal{F}_{\mathtt{b}}$, expressed in $\mathcal{F}_{\mathtt{a}}$.

In order to denote a point in 3D space consistently using our vector notation, we will first define the frame for the point as $\mathcal{F}_{\mathtt{p}}$. The vector describing the position of the point, expressed in frame $\mathcal{F}_{\mathtt{a}}$, is ${}^{\mathtt{a}}t^{\mathtt{a:p}}$. To improve notational clarity, we drop the tail to head relationship for the point vector and introduce the shorthand,

$$ {}^{\mathtt{a}}t^{\mathtt{a:p}} := {}^{\mathtt{a}}\mathbf{p} \in \mathbb{R}^3, \tag{2.1} $$

where the superscript a denotes $\mathcal{F}_{\mathtt{a}}$, the frame in which the point $\mathbf{p}$ is expressed. We will

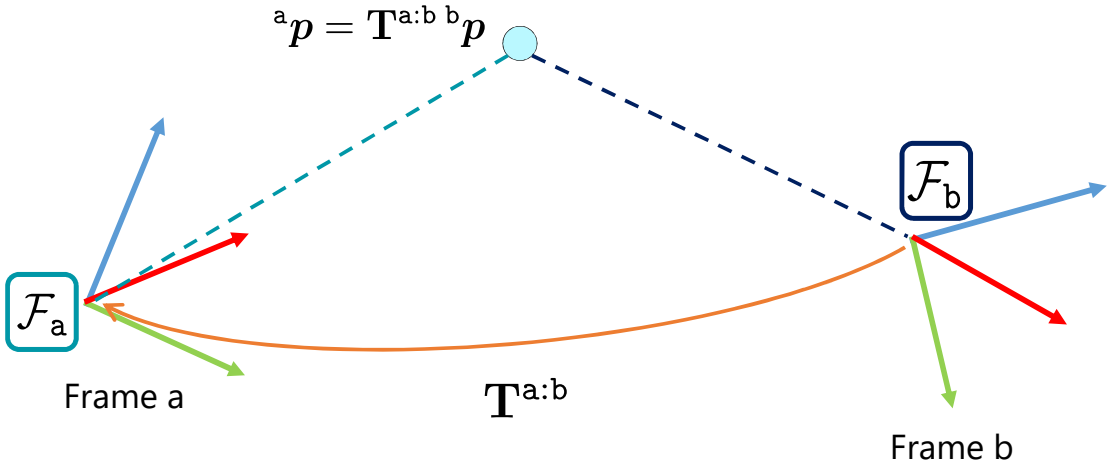


Figure 2.1: Example of two co-ordinates frames, $\mathcal{F}_{\mathtt{a}}$ and $\mathcal{F}_{\mathtt{b}}$, and the transformation between them, $\mathbf{T}^{\mathtt{a:b}}$, which can be read as, "the transformation which maps points from frame $\mathcal{F}_{\mathtt{b}}$, to frame $\mathcal{F}_{\mathtt{a}}$". This notation allows us to ensure the correct frames are being used when transforming points. For example, with the expression ${}^{\mathtt{a}}\boldsymbol{p} = \mathbf{T}^{\mathtt{a:b}}\,{}^{\mathtt{b}}\boldsymbol{p}$, the adjacent superscript $\mathtt{b}$ indicates that the frame which the point ${}^{\mathtt{b}}\boldsymbol{p}$ is expressed in, and the direction of the transformation, $\mathbf{T}^{\mathtt{a:b}}$ are compatible.

also express points in *homogeneous* co-ordinates using italicized script,

$$ {}^{\mathtt{b}}\boldsymbol{p} := \left[\begin{array}{c} {}^{\mathtt{b}}\mathbf{p} \\ \hline 1 \end{array}\right]. \tag{2.2} $$

Occasionally, we will need to extract columns from matrices. Suppose we have a matrix $\mathbf{M}$. Then, the notation $[\mathbf{M}]_i$ denotes the extraction of the i^{th} column from the matrix $\mathbf{M}$, and the notation $[\mathbf{M}]_{i:j}$ denotes the extraction of a sub-matrix which contains the i through j columns of $\mathbf{M}$.

2.2 Rigid Body Transformation

Suppose we are given two frames, $\mathcal{F}_{\mathtt{a}}$ and $\mathcal{F}_{\mathtt{b}}$. Then, the rigid body transformation that maps points from $\mathcal{F}_{\mathtt{b}}$ to $\mathcal{F}_{\mathtt{a}}$ is $\mathbf{T}^{\mathtt{a:b}} \in \mathbb{SE}(3)$, as depicted in Figure 2.1

The transformation, $\mathbf{T}^{\mathtt{a:b}}$, can also be expressed as a matrix,

$$\mathbf{T}^{\mathtt{a:b}} = \left[\begin{array}{c|c} \mathbf{R}^{\mathtt{a:b}} & {}^{\mathtt{a}}t^{\mathtt{a:b}} \\ \hline 0 & 1 \end{array}\right], \tag{2.3}$$

where $\mathbf{R}^{\mathtt{a:b}} \in \mathbb{SO}(3)$ is the rotation matrix from frame $\mathcal{F}_{\mathtt{b}}$ to $\mathcal{F}_{\mathtt{a}}$, and ${}^{\mathtt{a}}t^{\mathtt{a:b}}$ is the translation vector from $\mathcal{F}_{\mathtt{a}}$ and $\mathcal{F}_{\mathtt{b}}$, expressed in $\mathcal{F}_{\mathtt{a}}$. Using this notation, we are able to explicitly check the correctness of transformations by simply inspecting the head-to-tale relationships between vectors. For example, a point expressed in frame $\mathcal{F}_{\mathtt{b}}$, ${}^{\mathtt{b}}\boldsymbol{p}$, is transformed to frame $\mathcal{F}_{\mathtt{a}}$ with the expression ${}^{\mathtt{a}}\boldsymbol{p} = \mathbf{T}^{\mathtt{a:b}}\,{}^{\mathtt{b}}\boldsymbol{p}$, which is equivalent to the sequence,

$$\begin{aligned}
{}^{\mathtt{a}}\boldsymbol{p} &= \mathbf{T}^{\mathtt{a:b}}\,{}^{\mathtt{b}}\boldsymbol{p} && (2.4)\\
\left[\begin{array}{c} {}^{\mathtt{a}}t^{\mathtt{a:p}} \\ \hline 1 \end{array}\right] &= \left[\begin{array}{c|c} \mathbf{R}^{\mathtt{a:b}} & {}^{\mathtt{a}}t^{\mathtt{a:b}} \\ \hline 0 & 1 \end{array}\right] \left[\begin{array}{c} {}^{\mathtt{b}}t^{\mathtt{b:p}} \\ \hline 1 \end{array}\right] && (2.5)\\
&= \left[\begin{array}{c} \mathbf{R}^{\mathtt{a:b}}\,{}^{\mathtt{b}}t^{\mathtt{b:p}} + {}^{\mathtt{a}}t^{\mathtt{a:b}} \\ \hline 1 \end{array}\right] && (2.6)\\
&= \left[\begin{array}{c} {}^{\mathtt{a}}t^{\mathtt{a:b}} + {}^{\mathtt{a}}t^{\mathtt{b:p}} \\ \hline 1 \end{array}\right] && (2.7)\\
&= \left[\begin{array}{c} {}^{\mathtt{a}}t^{\mathtt{a:p}} \\ \hline 1 \end{array}\right]. && (2.8)
\end{aligned}$$

Since the translation vector component for the transformation matrix $\mathbf{T}^{\mathtt{a:b}}$ will always be expressed in frame $\mathcal{F}_{\mathtt{a}}$ in this work, we can exclude the prescript for the translation vector and equivalently express (2.3) as

$$\mathbf{T}^{\mathsf{a:b}} = \left[\begin{array}{c|c} \mathbf{R}^{\mathsf{a:b}} & \mathbf{t}^{\mathsf{a:b}} \\ \hline 0 & 1 \end{array}\right]. \tag{2.9}$$

Since the rotation component of $\mathbf{T}^{\mathsf{a:b}}$, $\mathbf{R}^{\mathsf{a:b}}$, is a member of the *special orthogonal* group, $\mathbb{SO}(3)$, it has an associated Lie algebra, which is denoted $\mathfrak{so}(3)$. A member of the $\mathbb{SO}(3)$ Lie algebra, $\phi^{\mathsf{a:b}} \in \mathfrak{so}(3)$, can be mapped to an element of $\mathbb{SO}(3)$, $\mathbf{R}^{\mathsf{a:b}}$, through the *exponential map*,

$$\exp : \mathfrak{so}(3) \mapsto \mathbb{SO}(3) \tag{2.10}$$

$$\exp(\phi^{\mathsf{a:b}}) \mapsto \mathbf{R}^{\mathsf{a:b}}. \tag{2.11}$$

In order to map elements from the $\mathbb{SO}(3)$ Lie group to the $\mathfrak{so}(3)$ Lie algebra, we shall use the *logarithmic map*,

$$\log : \mathbb{SO}(3) \mapsto \mathfrak{so}(3) \tag{2.12}$$

$$\log(\mathbf{R}^{\mathsf{a:b}}) \mapsto \phi^{\mathsf{a:b}}. \tag{2.13}$$

The lie algebra $\mathfrak{so}(3)$ is the set of all three by three skew-symmetric matrices,

$$\mathfrak{so}(3) = \{A \in \mathbb{R}^{3\times 3} : A^T = -A\}. \tag{2.14}$$

Since a 3×3 skew-symmetric matrix can also be encoded as a three dimensional vector, we will define a *hat* operator $[\cdot]^\wedge : \mathbb{R}^3 \mapsto \mathbb{SO}(3)$. Suppose we are given a vector $a = [a_1\, a_2\, a_3] \in \mathbb{R}^3$. Then,

$$[a]^\wedge = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \tag{2.15}$$

Similarly, we shall define a *vee* operator, $[\cdot]^\vee : \mathfrak{so}(3) \mapsto \mathbb{R}^3$, which provides a reverse mapping for Equation (2.15), and maps a skew-symmetric matrix in $\mathfrak{so}(3)$ to a three-dimensional vector in $\mathbb{R}^3$.

Since both the $\mathbb{SO}(3)$ and $\mathbb{SE}(3)$ groups are not vector spaces, they are not endowed with addition or subtraction operations. To that end, in Section 2.2.1, we shall use the log and exp mapping functions to map small perturbations from the $\mathbb{SO}(3)$ Lie group to the $\mathfrak{so}(3)$ Lie algebra, and vice-versa, to compute differentials. Note that the log and exp mappings from Equations (2.10) and (2.12) can also be defined for the Special Euclidean group, $\mathbb{SE}(3)$, but are not required in this work as we treat the rotation and translation component of (2.9) separately. A more complete description of the usage and properties of Matrix Lie Groups for transformations and orientations is provided by [77, 78].

2.2.1 Taking Derivatives on $\mathbb{SE}(3)$

Performing nonlinear optimization over variables which are members of $\mathbb{SE}(3)$ poses an issue using standard filtering and optimization frameworks, which rely on small differentials in order to compute derivatives. For example, consider the function $f : \mathbb{R}^n \mapsto \mathbb{R}^n$. We can compute the i^{th} Jacobian element of f as

$$\left[\frac{\partial}{\partial x} f(x)\right]_i = \lim_{\epsilon_i \to 0} \frac{f(x + \epsilon_i) - f(x)}{|\epsilon_i|}, \tag{2.16}$$

where $\epsilon_i \in \mathbb{R}^n$ is a perturbation vector which contains an infinitesimal non-zero value at the i^{th} element, and zero for all other elements.

In order to similarly compute derivatives involving elements of $\mathbb{SE}(3)$, we will use the *box-plus* and *box-minus* operators, which adapt the functions of addition and subtraction on $\mathbb{SE}(3)$ [79]. The box-plus operator, $\boxplus : \mathbb{SE}(3) \times \mathbb{R}^6$, allows us to add a small vector space perturbation,

$$\varphi^{\bar{\mathtt{a}}:\mathtt{a}} = \left[\begin{array}{c} \phi^{\bar{\mathtt{a}}:\mathtt{a}} \\ \hline \rho^{\bar{\mathtt{a}}:\mathtt{a}} \end{array}\right] \in \mathbb{R}^6, \tag{2.17}$$

to an element of $\mathbb{SE}(3)$, $\mathbf{T}^{\mathtt{a}:\mathtt{b}}$. Let $\phi^{\bar{\mathtt{a}}:\mathtt{a}} \in \mathbb{R}^3$ be a small rotation perturbation, and $\rho^{\bar{\mathtt{a}}:\mathtt{a}} \in \mathbb{R}^3$ be a small translation perturbation, such that when $\phi^{\bar{\mathtt{a}}:\mathtt{a}}$, is composed with the rotation, $\mathbf{R}^{\mathtt{a}:\mathtt{b}}$, and $\rho^{\bar{\mathtt{a}}:\mathtt{a}}$ is added to $\mathbf{t}^{\mathtt{a}:\mathtt{b}}$, the result is a transformation, $\mathbf{T}^{\bar{\mathtt{a}}:\mathtt{b}}$, from frame $\mathcal{F}_{\mathtt{b}}$, to a perturbed version of frame $\mathcal{F}_{\mathtt{a}}$, denoted as $\mathcal{F}_{\bar{\mathtt{a}}}$,

$$\left[\begin{array}{c|c} \mathbf{R}^{\mathtt{a}:\mathtt{b}} & \mathbf{t}^{\mathtt{a}:\mathtt{b}} \\ \hline 0 & 1 \end{array}\right] \boxplus \left[\begin{array}{c} \phi^{\bar{\mathtt{a}}:\mathtt{a}} \\ \hline \rho^{\bar{\mathtt{a}}:\mathtt{a}} \end{array}\right] = \left[\begin{array}{c|c} \exp([\phi^{\bar{\mathtt{a}}:\mathtt{a}}]^\wedge)\mathbf{R}^{\mathtt{a}:\mathtt{b}} & \rho^{\bar{\mathtt{a}}:\mathtt{a}} + \mathbf{t}^{\mathtt{a}:\mathtt{b}} \\ \hline 0 & 1 \end{array}\right] \tag{2.18}$$

$$= \left[\begin{array}{c|c} \mathbf{R}^{\bar{\mathtt{a}}:\mathtt{b}} & \mathbf{t}^{\bar{\mathtt{a}}:\mathtt{b}} \\ \hline 0 & 1 \end{array}\right] \tag{2.19}$$

$$= \mathbf{T}^{\bar{\mathtt{a}}:\mathtt{b}}. \tag{2.20}$$

The box-minus operator, $\boxminus : \mathbb{SE}(3) \times \mathbb{SE}(3) \mapsto \mathbb{R}^6$ allows us to determine the difference between two elements of $\mathbb{SE}(3)$ in terms of a vector space perturbation using the mapping

$$\left[\begin{array}{c|c} \mathbf{R}^{\bar{\mathtt{a}}:\mathtt{b}} & \mathbf{t}^{\bar{\mathtt{a}}:\mathtt{b}} \\ \hline 0 & 1 \end{array}\right] \boxminus \left[\begin{array}{c|c} \mathbf{R}^{\mathtt{a}:\mathtt{b}} & \mathbf{t}^{\mathtt{a}:\mathtt{b}} \\ \hline 0 & 1 \end{array}\right] = \left[\begin{array}{c} [\log(\mathbf{R}^{\bar{\mathtt{a}}:\mathtt{b}}(\mathbf{R}^{\mathtt{a}:\mathtt{b}})^T)]^\vee \\ \hline \mathbf{t}^{\bar{\mathtt{a}}:\mathtt{b}} - \mathbf{t}^{\mathtt{a}:\mathtt{b}} \end{array}\right] \tag{2.21}$$

$$= \left[\begin{array}{c} \phi^{\bar{\mathtt{a}}:\mathtt{a}} \\ \hline \rho^{\bar{\mathtt{a}}:\mathtt{a}} \end{array}\right] \tag{2.22}$$

Using the box-plus and box-minus operator, the operations required to calculate deriva-

tives involving elements of $\mathbb{SE}(3)$ can be defined. In this work, to compute derivatives, we treat the rotation and translation perturbations to an element of $\mathbb{SE}(3)$ separately, by extending the approach outlined in [77]. Note that it is also possible to compute derivatives using the exponential map defined for $\mathbb{SE}(3)$ [78], however, the approach discussed in Section 2.2.1 leads to less complex Jacobian expressions which facilitate degeneracy analysis, which is discussed in Chapter 3.

Suppose $g_1 : \mathbb{SE}(3) \mapsto \mathbb{R}^n$, $g_2 : \mathbb{R}^n \mapsto \mathbb{SE}(3)$, and $g_3 : \mathbb{SE}(3) \mapsto \mathbb{SE}(3)$. Then, the i^{th} Jacobian element can be computed as

$$\left[\frac{\partial}{\partial \mathbf{T}^{\mathtt{a:b}}} g_1(\mathbf{T}^{\mathtt{a:b}})\right]_i = \lim_{\varphi_i^{\bar{\mathtt{a}}:\mathtt{a}} \to 0} \frac{g_1(\mathbf{T}^{\mathtt{a:b}} \boxplus \varphi_i^{\bar{\mathtt{a}}:\mathtt{a}}) - g_1(\mathbf{T}^{\mathtt{a:b}})}{|\varphi_i^{\bar{\mathtt{a}}:\mathtt{a}}|}, \tag{2.23}$$

$$\left[\frac{\partial}{\partial x} g_2(x)\right]_i = \lim_{\epsilon_i \to 0} \frac{g_2(x + \epsilon_i) \boxminus g_1(x)}{|\epsilon_i|}, \tag{2.24}$$

$$\left[\frac{\partial}{\partial \mathbf{T}^{\mathtt{a:b}}} g_3(\mathbf{T}^{\mathtt{a:b}})\right]_i = \lim_{\varphi_i^{\bar{\mathtt{a}}:\mathtt{a}} \to 0} \frac{g_3(\mathbf{T}^{\mathtt{a:b}} \boxplus \varphi_i^{\bar{\mathtt{a}}:\mathtt{a}}) \boxminus g_3(\mathbf{T}^{\mathtt{a:b}})}{|\varphi_i^{\bar{\mathtt{a}}:\mathtt{a}}|}, \tag{2.25}$$

where $\varphi_i^{\bar{\mathtt{a}}:\mathtt{a}} \in \mathbb{R}^6$ is the perturbation vector from (2.17), with an infinitesimal value at the i^{th} element of $\varphi_i^{\bar{\mathtt{a}}:\mathtt{a}}$, and zero for all other elements.

2.3 Denavit-Hartenberg Parameterization

The Denavit-Hartenberg (DH) convention is a widely used method to assign co-ordinate frames to the links of a robotic manipulator. Here, we will provide a brief overview of the DH approach for a serial manipulator with rotational joints. For more detailed information, we refer the reader to some of the popular references for manipulator modelling and control [80, 81].

Suppose co-ordinate frame, $\mathcal{F}_{\mathtt{i}}$, is attached to the i^{th} link of a robotic manipulator. The

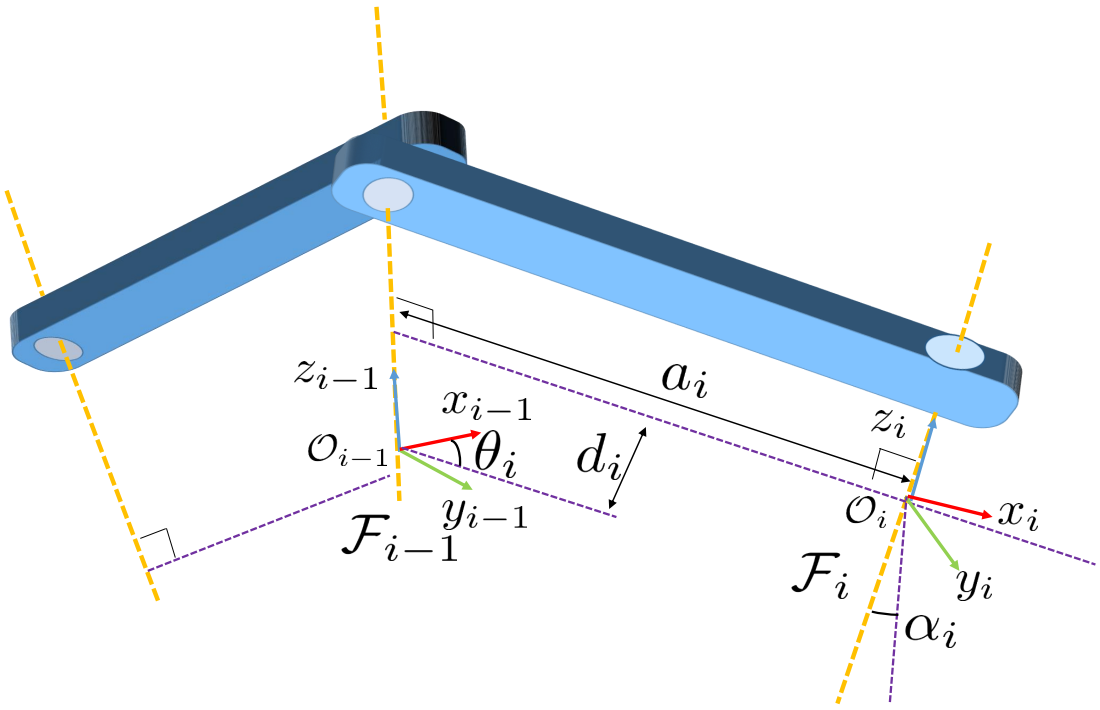


Figure 2.2: Example of DH convention between two rotational joints

DH convention uses four independent parameters to define the transformation between adjacent links, as depicted in Figure 2.2.

Consider the two adjacent co-ordinate frames $\mathcal{F}_{\mathtt{i-1}}$ and $\mathcal{F}_{\mathtt{i}}$ from Figure 2.2. In order to construct co-ordinate frame i using the DH convention, the z axis of the frame is placed co-incident with the joint angle. Then, a common normal direction between z_{i-1} and z_i can be determined as

$$n_i = \frac{z_{i-1} \times z_i}{\|z_{i-1} \times z_i\|}. \tag{2.26}$$

Using the common normal, the x_i axis is placed along n_i and points from z_{i-1} to z_i, and the intersection of the x_i and z_i axes define the origin, $\mathcal{O}_i$, of frame $\mathcal{F}_{\mathtt{i}}$. With the x_i and z_i axes defined, the y_i axis is constructed on the frame according to the right-hand rule. Typically, frames are assigned in a sequential fashion, starting from the end effector frame and ending at the base frame of the mechanism.

With frame $\mathcal{F}_{\mathtt{i}}$ constructed, the transformation between frames $\mathcal{F}_{\mathtt{i-1}}$ and $\mathcal{F}_{\mathtt{i}}$ can be defined using the DH parameters. First, frame $\mathcal{F}_{\mathtt{i-1}}$ is rotated about axis z_{i-1} by the joint rotation parameter θ_i. Second, frame $\mathcal{F}_{\mathtt{i-1}}$ is translated along the z_{i-1} axis by the link offset parameter d_i. Third, $\mathcal{F}_{\mathtt{i-1}}$ is translated along the direction of the x_i axis by the link length parameter, a_i. Finally, the $\mathcal{F}_{\mathtt{i-1}}$ frame is rotated about the x_i axis by the twist angle parameter, α_i. After applying the transformations with the four parameters, frames $\mathcal{F}_{\mathtt{i-1}}$ and $\mathcal{F}_{\mathtt{i}}$ are co-incident.

In this work, we shall denote the DH parameters which describe the transformation between frames $\mathcal{F}_{\mathtt{i-1}}$ and $\mathcal{F}_{\mathtt{i}}$ on an actuated mechanism as $\theta_i \in \mathbb{R}$, which represents the revolute joint angle, and $\omega_i = [d_i, a_i, \alpha_i]^T \in \mathbb{R}^3$, which represent the link length, link offset, and link twist angle, respectively. Using the DH parameters, a homogeneous rigid body transformation, $\mathbf{T}_{\omega_i}^{\mathtt{i:i-1}} \in \mathbb{SE}(3)$, can be computed as

$$\mathbf{T}_{\omega_i,\theta_i}^{\mathtt{i:i-1}} = \mathbf{T}_{\theta_i}^{\mathtt{i:s_i}} \mathbf{T}_{d_i}^{\mathtt{s_i:r_i}} \mathbf{T}_{a_i}^{\mathtt{r_i:q_i}} \mathbf{T}_{\alpha_i}^{\mathtt{q_i:i-1}} \tag{2.27}$$

where the frames $\mathcal{F}_{\mathtt{q_i}}$, $\mathcal{F}_{\mathtt{r_i}}$, $\mathcal{F}_{\mathtt{s_i}}$ are intermediate frames within the compounded transform of Equation (2.27), and the transformations of the individual DH parameters are given as,

$$\mathbf{T}_{\theta_i}^{\mathtt{i:s_i}} = \begin{bmatrix} c(\theta_i) & -s(\theta_i) & 0 & 0 \\ s(\theta_i) & c(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.28}$$

$$\mathbf{T}_{d_i}^{\mathtt{s_i:r_i}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.29}$$

$$\mathbf{T}_{a_i}^{\mathtt{r_i:q_i}} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.30}$$

$$\mathbf{T}_{\alpha_i}^{\mathtt{q_i:i-1}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c(\alpha_i) & -s(\alpha_i) & 0 \\ 0 & s(\alpha_i) & c(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.31}$$

where $c(\cdot)$ and $s(\cdot)$ denote the $\cos(\cdot)$ and $\sin(\cdot)$ of an angle, respectively. Multiplying out the transformation from Equations (2.28) - (2.31), results in the combined DH transformation matrix,

$$\mathbf{T}_{\omega_i}^{\mathtt{i:i-1}} = \begin{bmatrix} c(\theta_i) & -s(\theta_i)c(\alpha_i) & s(\theta_i)s(\alpha_i) & a_i c(\theta_i) \\ s(\theta_i) & c(\theta_i)c(\alpha_1) & -c(\theta_i)s(\alpha_i) & a_i s(\theta_i) \\ 0 & s(\alpha_i) & c(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.32}$$

2.4 Image Feature Tracking and Projections

The projection function, which maps a point from the camera frame, ${}^{c}\boldsymbol{p}_i$, to a pixel location on the 2D image plane is defined as

$$\begin{aligned} \Psi({}^{c}\boldsymbol{p}_i) &: \mathbb{R}^3 \mapsto \mathbb{P}^2 \\ \Psi({}^{c}\boldsymbol{p}_i) &= [u_i \, v_i]^T, \end{aligned} \tag{2.33}$$

where u_i and v_i are the pixel co-ordinates of the projected point along the u and v image directions, respectively. Note that the projection can be performed using a variety of camera models, such as pinhole, Taylor, [82] or omni model [83]. In the standard pinhole camera model, light rays are represented as lines which converge at the center of projection and intersect with the image plane. In order to accommodate the large radial distortion caused by fish-eye lenses, the Taylor camera model, uses a spherical mapping where the elevation and azimuth angles to a 3D point, $\mathbf{s} = [\theta, \phi]^T$, are modelled as half lines which pass through the sphere's centre. The unit vector on the sphere is then mapped to the image plane through a polynomial mapping function. To model the misalignment between the image sensor and the lens axis, the point's final pixel location is adjusted using an affine transformation.

In this work, we do not assume a specific type of camera model, however, it is important to consider the decrease in sensitivity of the measurement Jacobian for wide angle camera models such as the Taylor model [82] and omni model [83]. In such cases, image measurements of the points seen near the boundary of the lens' field of view are less sensitive to perturbations of the point position in 3D, thus degrading the information quality required for precise localization of the camera. The effects of these lens model sensitivities on localization are discussed in detail in Chapter 5.

2.5 Nonlinear Least Squares on Manifolds

When solving estimation problems, a *measurement model*, is used to model observed measurements, given a state estimate. Suppose we have a state or parameter vector we wish to estimate, denoted by $\mathbf{x} \in \mathbb{R}^n$. Then, the nonlinear measurement model is given as,

$$\hat{\mathbf{z}} = h(\mathbf{x}). \tag{2.34}$$

The goal of nonlinear least squares estimation methods is to determine the optimal state vector, $\mathbf{x}^*$, by optimizing the least squares cost function $\Lambda(\mathbf{x}) : \mathbb{R}^n \mapsto \mathbb{R}$,

$$\Lambda(\mathbf{x}) = \frac{1}{2}\mathbf{e}^T \Omega \mathbf{e}, \tag{2.35}$$

where $\mathbf{e} = (\mathbf{z} - h(\mathbf{x}))$ is known as the least squares residual vector, $\Omega = \Sigma^{-1}$ is the measurement noise information matrix, and $\mathbf{z} \in \mathbb{R}^m$ is the vector of observed measurements. By finding the optimal state vector which best explains the observed measurements, optimizing the cost function from (2.35) performs maximum-likelihood estimation, however, the cost can be modified to include prior information about the state, in which case the process becomes maximum a posteriori estimation [84, 78].

The measurements, $\mathbf{z}$, may not always be a vector space quantity, but instead could be a member of a compound manifold, as would be the case if the system was observing direct measurements of poses, which are a member of the special Euclidean group, $\mathbb{SE}(3)$. To generalize the measurement model from (2.34), let us redefine it as a function which maps values from the state space $\mathcal{S}$, to the measurement space $\mathcal{M}$,

$$h(\mathbf{x}) : \mathcal{S} \mapsto \mathcal{M} \tag{2.36}$$

Then, the error term, e, from (2.35), can be rewritten using the box-minus operator described in Section 2.2.1, as

$$\mathbf{e} = (\mathbf{z} \boxminus h(\mathbf{x})). \tag{2.37}$$

To optimize the cost function provided by Equation 2.35 using the error term from Equation (2.37), the process proceeds iteratively, starting with an initial state estimate, $\breve{\mathbf{x}}_0$. Each iteration seeks to update the current state estimate, $\breve{\mathbf{x}}_k$ using a vector space perturbation, $\varphi_k \in \mathbb{R}^n$

$$\breve{\mathbf{x}}_{k+1} = \breve{\mathbf{x}}_k \boxplus \varphi_k, \tag{2.38}$$

such that the sequence $\{\breve{\mathbf{x}}_0, \breve{\mathbf{x}}_1, \breve{\mathbf{x}}_2, \cdots\}$ converges to the optimal solution $\mathbf{x}^*$. Note that the system may converge to a locally optimal solution, based on how the system is initialized.

To find the value of the update perturbation, the cost function from 2.35 is linearized about the estimate $\breve{\mathbf{x}}_k$ using a second-order Taylor-series expansion,

$$\Lambda(\breve{\mathbf{x}}_k \boxplus \varphi_k) \approx \Lambda(\breve{\mathbf{x}}_k) + \left[\frac{\partial \Lambda(\breve{\mathbf{x}}_k \boxplus \varphi_k)}{\partial \varphi_k}\bigg|_{\varphi_k=0}\right]^T \varphi_k + \frac{1}{2}\varphi_k^T \left[\frac{\partial^2 \Lambda(\breve{\mathbf{x}}_k \boxplus \varphi_k)}{\partial \varphi_k^2}\bigg|_{\varphi_k=0}\right] \varphi_k. \tag{2.39}$$

To minimize the Taylor-series expansion of the cost function provided in Equation (2.39), its derivative with respect to the update perturbation, φ_k, is set to zero, resulting in,

$$\left[\frac{\partial^2 \Lambda(\breve{\mathbf{x}}_k \boxplus \varphi_k)}{\partial \varphi_k^2}\bigg|_{\varphi_k=0}\right] \varphi_k = -\left[\frac{\partial \Lambda(\breve{\mathbf{x}}_k \boxplus \varphi_k)}{\partial \varphi_k}\bigg|_{\varphi_k=0}\right]. \tag{2.40}$$

For the quadratic cost function from 2.35, its first partial derivative with respect to the update perturbation evaluates to

$$\frac{\partial \Lambda(\breve{\mathbf{x}}_k \boxplus \varphi_k)}{\partial \varphi_k}\bigg|_{\varphi_k=0} = \mathbf{J}^T \Omega \mathbf{e}_k, \tag{2.41}$$

where

$$\mathbf{J} = \left.\frac{\partial e(\breve{\mathbf{x}}_k \boxplus \varphi_k)}{\partial \varphi_k}\right|_{\varphi_k=0} \tag{2.42}$$

is the Jacobian matrix of the residual term $e(\mathbf{x})$ from Equation (2.37), with respect to the update vector, φ_k [84]. The second derivative of the cost function, or the Hessian matrix, is approximated by

$$\left.\frac{\partial^2 \Lambda(\breve{\mathbf{x}}_k \boxplus \varphi_k)}{\partial \varphi_k^2}\right|_{\varphi_k=0} \approx \mathbf{J}^T \Omega \mathbf{J}. \tag{2.43}$$

Substituting the terms from (2.41) and (2.43) into the update Equation (2.40) results in the nonlinear least squares normal equations,

$$(\mathbf{J}^T \Omega \mathbf{J})\varphi_k = \mathbf{J}^T \Omega \mathbf{e}_k, \tag{2.44}$$

which can be solved for the update perturbation φ_k, so long as the Hessian matrix approximation,

$$\mathbf{J}^T \Omega \mathbf{J} \tag{2.45}$$

is invertible. If the approximated Hessian is singular, then the system is ill-conditioned and cannot be solved. The cases where the DCC calibration system is degenerate are investigated in Chapter 3.

Once the update perturbation is solved, the linearization is updated using Equation (2.38). The optimization proceeds iteratively using the new state estimate, φ_{k+1}, until the termination criteria are fulfilled. These include a maximum iteration limit being reached, the magnitude of the state update falling below a threshold, or the magnitude of the cost reduction becoming less than a selected threshold. Ideally, the solution will converge to the global minimum. However, the system can settle to local minima depending on the shape of the cost function and the initial estimate of the solution. Therefore, it is vital

to supply the optimization algorithm with a reasonably accurate initial state estimate to help with convergence to the correct solution.

2.6 Information Theory

Information Theory is generally considered to initially have been developed in order to facilitate reliable data transmission over noisy communication channels. It has since been established as a means of quantifying information content. This section will outline some fundamental tools from information theory, which are used in this thesis. For a more detailed treatment of the area, the reader is directed to the work of Cover and Thomas [85].

2.6.1 Entropy and Mutual Information for Discrete Random Variables

The Shannon entropy is a measure of the unpredictability or uncertainty of information content. Suppose $X = \{x_1, x_2, \ldots, x_n\}$ is a discrete random variable. The Shannon entropy for X, $H(X)$ is given as

$$H(X) = -\sum_{x_i \in X} P(x_i) \log P(x_i), \tag{2.46}$$

where $P(x_i)$ denotes the probability of event x_i occurring. The Shannon entropy provides a scalar value that quantifies the average variance of the discrete random variable X. The base of the logarithm in Equation (2.46) denotes the units of the entropy. In the case where the base of the logarithm is 2, the units are referred to as *bits*, and when performed using the natural logarithm, the units are referred to as *nats*.

We can also define the *joint entropy* of two variables, which can be defined as the

entropy over their joint probability distribution,

$$H(X,Y) = -\sum_{x_i \in X, y_i \in Y} P(x_i, y_i) \log P(x_i, y_i). \tag{2.47}$$

When the two variables are independent, it can be shown that their joint entropy is equal to the sum of their marginal entropies,

$$H(X,Y) = H(X) + H(Y) \tag{2.48}$$

In the case where two variables are correlated, then having knowledge about one of the variables can affect the entropy of the other. For example, if the value of the random variable Y is given, then we can compute the *conditional entropy* of X given Y by averaging over all possible outcomes of y_i

$$H(X|Y) = -\sum_{y_i \in Y} P(y_i) \left[\sum_{x_i \in X} P(x_i|y_i) \log P(x_i|y_i) \right] \tag{2.49}$$

$$= -\sum_{x_i \in X, y_i \in Y} P(x_i, y_i) \log P(x_i|y_i). \tag{2.50}$$

Observing Equations (2.46),(2.47) and (2.49), we see that

$$H(X|Y) = H(X,Y) - H(Y). \tag{2.51}$$

Equation (2.51) illustrates the intuitive meaning of conditional entropy, as it quantifies the uncertainty exhibited by a variable, once another variable that is correlated with the first has been observed.

The relative entropy (also called the *information divergence* or *Kullback-Leibler divergence*) provides a measure of the difference between two probability distributions. If we

denote two discrete probability distributions, X and Y, then the relative entropy between the two is given as

$$\mathcal{D}(X\|Y) = \sum_i P(x_i) \log \frac{P(x_i)}{P(y_i)}. \tag{2.52}$$

The relative entropy measures the information lost when Y is used to approximate X, and is zero if and only if $X = Y$. Using the definition of relative entropy, the Kullback-Leibler divergence between a joint probability distribution and its marginal distribution, is given as

$$I(X;Y) = \sum_{x_i \in X, y_i \in Y} P(x_i, y_i) \log \frac{P(x_i, y_i)}{P(x_i)P(y_i)}. \tag{2.53}$$

Since Equation (2.53) measures the difference between a joint probability distribution and its marginal distribution, in effect, it quantifies the correlation or common information between the random variables X and Y. This measure is known as *mutual information*. In the special case where the two variables are independent, the numerator term of (2.53) evaluates to $P(x_i, y_i) = P(x_i)P(y_i)$, resulting in a cancellation and thus zero mutual information between the variables. The mutual information equation can also be expressed as,

$$I(X;Y) = H(X) + H(Y) - H(X,Y), \tag{2.54}$$

which is visualized in Figure 2.3.

2.6.2 Entropy and Mutual Information for Gaussian Random Variables

Using a similar approach to the ones described for discrete random variables, it is possible to derive the information theoretic measures of entropy and mutual information for continuous random variables [85]. In this section, we will look at the specific case where the continuous random variable is modelled as a Gaussian distrubution, as this is the most prevalent case

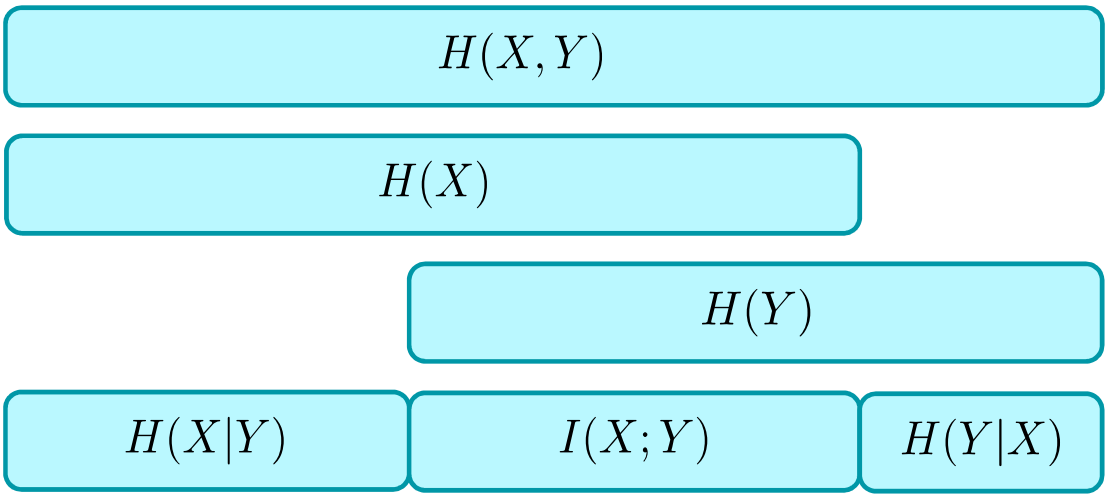


Figure 2.3: Relationship between joint entropy, marginal entropy, conditional entropy, and mutual information [86].

which arises when performing visual navigation and SLAM.

In the case where the probability density function of the continuous random variable, Y, is modelled as a Gaussian distribution, the entropy (sometimes referred to as *differential entropy*) is derived to be

$$h_e(Y) = \ln(\sigma\sqrt{2\pi e}), \tag{2.55}$$

where σ is the variance of the distribution, and $h_e(Y)$ is used to denote that the logarithm was taken with base e, in order to present Equation (2.55) in a simplified form. Similarly, the entropy for a multivariate Gaussian distribution can be computed as

$$h_e(Y) = \frac{1}{2}\ln((2\pi e)^n\,|\Sigma|), \tag{2.56}$$

where Σ is the covariance matrix of the multivariate Gaussian distribution, and $|\cdot|$ denotes the determinant operator. Note that unlike the entropy for discrete random variables, it is possible for the entropy of continuous random variables to be less than zero.

Suppose we have a Gaussian distribution which is defined by its mean vector, x, and

its covariance matrix, Σ. If the distribution is partitioned as

$$x = \left[\begin{array}{c} a \\ \hline b \end{array}\right], \quad \Sigma = \left[\begin{array}{c|c} \Sigma_{aa} & \Sigma_{ab} \\ \hline \Sigma_{ba} & \Sigma_{bb} \end{array}\right]. \tag{2.57}$$

Then, the mutual information between the random variables a, and b is given as [51],

$$I(a;b) = \frac{1}{2}\log\frac{|\Sigma_{aa}||\Sigma_{bb}|}{|\Sigma|}. \tag{2.58}$$

Note that if the variables a and b are uncorrelated then $\Sigma_{ba} = \Sigma_{ab} = 0$, and thus $|\Sigma| = |\Sigma_{aa}||\Sigma_{bb}|$. For this special case, we see that the mutual information measure defined in Equation (2.58) evaluates to zero, exactly as in the general discrete case presented in Equation (2.53).

2.7 Multiple Camera Parallel Tracking and Mapping

MCPTAM is a real-time, feature-based, visual SLAM algorithm which extends Klein and Murray's Parallel Tracking and Mapping (PTAM) [72] in four ways. First it allows multiple, non-overlapping field-of-view (FOV), heterogeneous cameras in any fixed configuration to be successfully combined. The flexible camera configuration enables the use of wide baselines and high visibility, which allows features to be tracked for long periods. Second it extends the PTAM's pinhole camera model to work with fish-eye and omnidirectional lenses through the use of the Taylor camera model [82], which helps mitigate feature starvation due to occlusions and texture-less frames in any single camera. Third, PTAM's back-end has been replaced with the g2o optimizer allowing for faster and more flexible optimization structures [84]. Finally, MCPTAM introduces both an improved update process and a novel feature parameterization using spherical co-ordinates anchored in a base key-frame

co-ordinate system [11].

A brief formulation of MCPTAM is given as follows. Let the point-feature map, $\boldsymbol{P}$, be a set of points defined with respect to an inertial world frame $\mathcal{F}_{\mathtt{w}}$, $\boldsymbol{P} = \{{}^{\mathtt{w}}\mathbf{p}_1, {}^{\mathtt{w}}\mathbf{p}_2, \ldots, {}^{\mathtt{w}}\mathbf{p}_n\}$ for the n map points.

2.7.1 Tracking

We shall denote one of the cameras in the cluster as the *base* camera, and the remaining cameras in the cluster as the *auxiliary* cameras. The co-ordinate frame of the cluster is chosen to be the frame of the base camera, and without loss of generality, the first camera index is chosen to be the base camera. Let us denote the pose of the camera cluster base frame, $\mathcal{F}_{\mathtt{c_b}}$, with respect to the world frame, at time step k, as $\mathbf{T}_k^{\mathtt{w:c_b}}$. In order to track the camera cluster pose, the map points are re-projected into the image frames of the cameras.

Denote a rigid body transformation which transforms a point from the j^{th} auxiliary camera frame $\mathcal{F}_{\mathtt{c_j}}$, to the base camera frame, as $\mathbf{T}^{\mathtt{c_j:c_b}} \in \mathbb{SE}(3)$. Note that in the standard MCPTAM formulation presented in [11], this transformation between an auxiliary camera and the base camera is static, and is computed off-line using a calibration procedure.

A map point, which is expressed in the world frame, ${}^{\mathtt{w}}\boldsymbol{p}$, can be transformed into the frame of the j^{th} camera through the transformation chain ${}^{\mathtt{c_j}}\boldsymbol{p} = \mathbf{T}^{\mathtt{c_j:c_b}}(\mathbf{T}_k^{\mathtt{w:c_b}})^{-1}\,{}^{\mathtt{w}}\boldsymbol{p}$. An example camera cluster is depicted in Figure 2.4.

An image feature detector and descriptor is used to find points in the image which correspond to the existing points maintained in the map. Given a set of corresponding points, the camera cluster pose parameters are found through a manifold based nonlinear least squares optimization process, which seeks to determine the cluster pose such that the re-projection errors between corresponding points is minimized. Note that the MCPTAM method uses a robust cost function to weight the residual terms, which can be computed

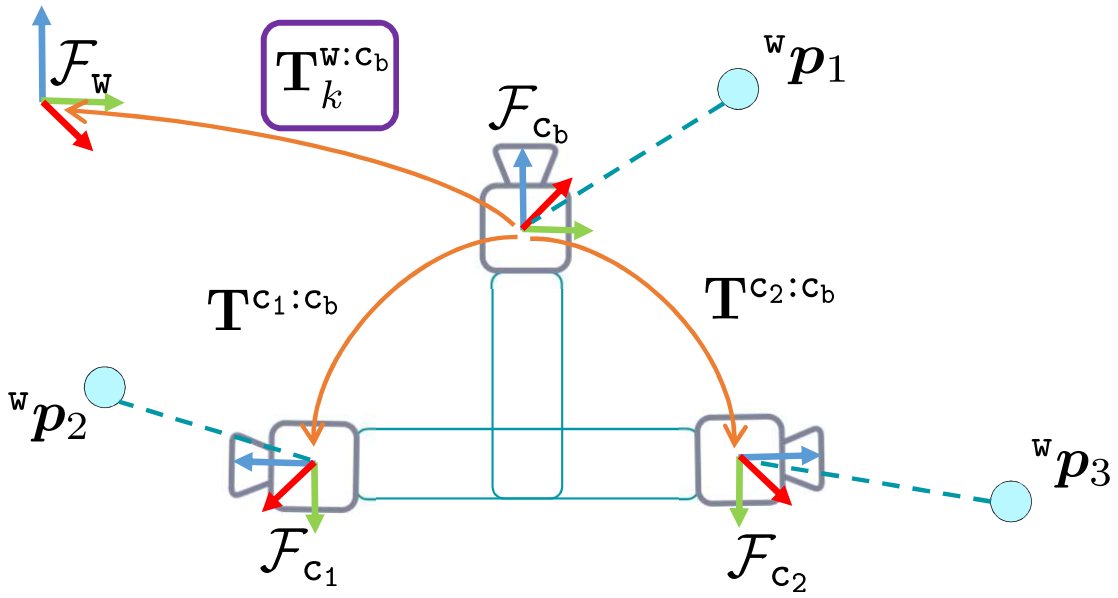


Figure 2.4: An example camera cluster, where $\mathcal{F}_{\mathtt{w}}$ denotes the world frame, and $\mathcal{F}_{\mathtt{c_b}}$, $\mathcal{F}_{\mathtt{c_1}}$, and $\mathcal{F}_{\mathtt{c_2}}$ denote the frames for the base camera, and the first and second auxiliary cameras in the cluster, respectively. Note that estimation of the cluster position, $\mathbf{T}_k^{\mathtt{w}:\mathtt{c_b}}$, is with respect to the base camera, but points can be mapped between frames using the known extrinsic calibrations which exist between the base and auxiliary cameras.

using standard approaches, such as Huber or Tukey weighting [10].

Let the set of map points from $\boldsymbol{P}$ which are visible to camera j at time k, and for which valid image feature correspondences exist, be denoted as $\tilde{\boldsymbol{P}}_j^k \subset \boldsymbol{P}$. Given a point ${}^{\mathtt{w}}\boldsymbol{p}_i \in \tilde{\boldsymbol{P}}_j^k \subseteq \mathbb{R}^3$, the image feature measurement, taken with respect to the j^{th} camera in the cluster, is denoted as $\boldsymbol{z}_i^j \in \mathbb{R}^2$. A measurement of a feature is composed of two elements, corresponding to the u and v pixel co-ordinates, respectively. Thus, a single feature provides two independent measurements per observing camera. The full measurement vector is constructed by stacking all measurements at time k into a single column vector, and is denoted as $\boldsymbol{y}^k \in \mathbb{R}^m$. In a similar fashion, the nonlinear measurement model for the j^{th} camera, $h_j(\mathbf{T}_k^{\mathtt{w}:\mathtt{c}_\mathtt{b}}) : \mathbb{SE}(3) \mapsto \mathbb{R}^{|\tilde{\boldsymbol{P}}_j^k|}$ is given as

$$h_j(\mathbf{T}_k^{\mathtt{w}:\mathtt{c}_\mathtt{b}}) = \begin{bmatrix} \Psi_j(\mathbf{T}^{\mathtt{c}_\mathtt{j}:\mathtt{c}_\mathtt{b}}(\mathbf{T}_k^{\mathtt{w}:\mathtt{c}_\mathtt{b}})^{-1}\,{}^{\mathtt{w}}\boldsymbol{p}_1 \\ \Psi_j(\mathbf{T}^{\mathtt{c}_\mathtt{j}:\mathtt{c}_\mathtt{b}}(\mathbf{T}_k^{\mathtt{w}:\mathtt{c}_\mathtt{b}})^{-1}\,{}^{\mathtt{w}}\boldsymbol{p}_2 \\ \Psi_j(\mathbf{T}^{\mathtt{c}_\mathtt{j}:\mathtt{c}_\mathtt{b}}(\mathbf{T}_k^{\mathtt{w}:\mathtt{c}_\mathtt{b}})^{-1}\,{}^{\mathtt{w}}\boldsymbol{p}_3 \\ \vdots \\ \Psi_j(\mathbf{T}^{\mathtt{c}_\mathtt{j}:\mathtt{c}_\mathtt{b}}(\mathbf{T}_k^{\mathtt{w}:\mathtt{c}_\mathtt{b}})^{-1}\,{}^{\mathtt{w}}\boldsymbol{p}_{|\tilde{\boldsymbol{P}}_j^k|} \end{bmatrix} \tag{2.59}$$

Note that the nonlinear measurement model for the full camera cluster is found by simply stacking the measurement models from Equation (2.59) for all cameras in the cluster.

The re-projection error is calculated by transforming the map points in $\tilde{\boldsymbol{P}}^k$ according to the current camera cluster pose, projecting the points into image co-ordinates using the measurement model, and subtracting from the point's corresponding image measurement. A cost function, $\Lambda_{\tilde{\boldsymbol{P}}^k}(\mathbf{T}_k^{\mathtt{w}:\mathtt{c}_\mathtt{b}}) : \mathbb{SE}(3) \mapsto \mathbb{R}$, which penalizes the re-projection error over the set of corresponding map points, $\tilde{\boldsymbol{P}}^k$, is defined as

$$\Lambda_{\tilde{\boldsymbol{P}}^k}(\mathbf{T}_k^{\mathtt{w}:\mathtt{c}_\mathtt{b}}) = \sum_{j=1}^{C} \sum_{i=1}^{|\tilde{\boldsymbol{P}}_j^k|} \|\boldsymbol{z}_i^j - \Psi_j(\mathbf{T}^{\mathtt{c}_\mathtt{j}:\mathtt{c}_\mathtt{b}}(\mathbf{T}_k^{\mathtt{w}:\mathtt{c}_\mathtt{b}})^{-1}\,{}^{\mathtt{w}}\boldsymbol{p}_i\|, \tag{2.60}$$

where C is the total number of cameras in the cluster. Finally, the pose of the camera cluster at time k is estimated by optimizing the cost given by Equation (2.60), using the manifold optimization framework outlined in Section 2.5. The optimal transformation parameters, $\mathbf{T}_k^{\mathtt{w:c_b}*}$, is given as

$$\mathbf{T}_k^{\mathtt{w:c_b}*} = \underset{\mathbf{T}_k^{\mathtt{w:c_b}} \in \mathbb{SE}(3)}{\operatorname{argmin}} \; \Lambda_{\tilde{\boldsymbol{P}}^k}(\mathbf{T}_k^{\mathtt{w:c_b}}). \tag{2.61}$$

2.7.2 Mapping

By re-observing features as the camera cluster explores the environment, the point locations in the map can be refined using additional measurements. Furthermore, new map points can be inserted into the map in order to provide landmarks to track against. To perform these tasks, MCPTAM uses *key-frames*, which are a snapshot of the images and point measurements taken from a point along the camera cluster's trajectory. Since MCPTAM performs tracking using multiple cameras, it extends the idea of key-frames to *multi-keyframes*, which are simply a collection of the key-frames from the individual cameras at a particular instant in time.

We shall define a multi-keyframe, M, as collection of key-frames, $M = \{K_1, K_2, \ldots, K_C\}$, where each key-frame corresponds to a camera within the cluster. Each multi-keyframe is associated with its pose in $\mathbb{SE}(3)$. In order to insert a new multi-keyframe into the map, the point measurements from each observing key-frame are collected, and the parameters of the point locations, as well as the key-frame poses are optimized using a bundle adjustment procedure [11].

Chapter 3

Dynamic Multi-Camera Cluster Calibration

This chapter describes the calibration process for a dynamic MCC where some or all of the cameras are non-static. First, we formulate the calibration process between a single static camera and a camera mounted to an actuated mechanism, which will be referred to as the *dynamic camera*. Our calibration process requires a region of overlapping FOV between the static and dynamic camera, but only over a subset of all possible configurations of the dynamic camera. Second, we will describe an extension of the static-to-dynamic camera calibration case which will allow for calibration of the dynamic-to-dynamic camera case. Using the static-to-dynamic and dynamic-to-dynamic camera calibration techniques, the extrinsics of any arbitrary dynamic camera cluster can be calibrated in a pair-wise fashion, provided each calibration pair has sufficient field-of-view overlap. Note that simultaneous calibration over multiple camera pairs is a trivial extension to the presented method.

3.1 Static-to-Dynamic Camera Calibration Using Reprojection Error

Suppose we are performing DCC calibration for a mechanism with M links, with link 1 starting at the end effector of the mechanism, and link M ending at the base frame of the mechanism. The transformation between the static camera frame, $\mathcal{F}_{\mathtt{s}}$, and the dynamic camera frame $\mathcal{F}_{\mathtt{d}}$, has the form

$$\mathbf{T}^{\mathtt{s:d}}_{\nu,\bar{\theta}} = \mathbf{T}^{\mathtt{s:b}}\mathbf{T}^{\mathtt{b:e_{M-1}}}_{\omega_M,\theta_M} \cdots \mathbf{T}^{\mathtt{e_2:e_1}}_{\omega_2,\theta_2}\mathbf{T}^{\mathtt{e_1:e}}_{\omega_1,\theta_1}\mathbf{T}^{\mathtt{e:d}}, \tag{3.1}$$

where $\mathbf{T}^{\mathtt{s:b}}$ defines the transformation from the mechanism base frame, $\mathcal{F}_{\mathtt{b}}$, to the static camera frame, $\mathcal{F}_{\mathtt{s}}$, $\mathbf{T}^{\mathtt{e:d}}$ defines the transformation from the dynamic camera frame, $\mathcal{F}_{\mathtt{d}}$, to the mechanism end-effector frame, $\mathcal{F}_{\mathtt{e}}$, and $\mathbf{T}^{\mathtt{e_{i+1}:e_i}}_{\omega_{i+1},\theta_{i+1}}$ defines the transformation from the i^{th} actuated mechanism link, to the $i+1^{\text{th}}$ actuated mechanism link, which is composed of the DH parameters and joint angle inputs of the $i+1^{\text{th}}$ actuated mechanism link, denoted as ω_{i+1} and θ_{i+1}, respectively. In this way, the product

$$\mathbf{T}^{\mathtt{b:e}}_{\omega,\bar{\theta}} = \mathbf{T}^{\mathtt{b:e_{M-1}}}_{\omega_M,\theta_M} \cdots \mathbf{T}^{\mathtt{e_2:e_1}}_{\omega_2,\theta_2}\mathbf{T}^{\mathtt{e_1:e}}_{\omega_1,\theta_1} \tag{3.2}$$

represents the kinematic chain of transformations from the end effector of the mechanism, to its base frame, as a function of its DH parameters and joint angle inputs. Finally, note that using the provided indexing scheme, frame $\mathcal{F}_{\mathtt{e_M}}$ is equivalent to the base frame $\mathcal{F}_{\mathtt{b}}$.

The aim of the calibration process is to determine the total rigid body transformation $\mathbf{T}^{\mathtt{s:d}}_{\nu,\theta}$, where $\nu \in \mathcal{S}$ is the set of *estimated* parameters which is used to build the transform chain from Equation (3.1),

$$\nu = \{\mathbf{T}^{\mathtt{e:d}}, \omega_1, \omega_2, \cdots, \omega_M, \mathbf{T}^{\mathtt{s:b}}\} \in \mathcal{S}. \tag{3.3}$$

The values $\omega = [\omega_1, \omega_2, \cdots, \omega_M] \in \mathbb{R}^{3M}$ denote the unknown DH parameters of the mechanism which must be estimated, $\mathbf{T}^{\mathtt{e:d}} \in \mathbb{SE}(3)$ is the unknown end-effector to dynamic camera transformation, $\mathbf{T}^{\mathtt{s:b}} \in \mathbb{SE}(3)$ is the unknown base frame to static camera transformation, and $\mathcal{S}$ is the compound state-space for the unknown quantities, consisting of parameters which exist in both the vector space $\mathbb{R}^3$, and the Matrix Lie group $\mathbb{SE}(3)$.

We shall also define $\bar{\theta} = [\theta_1, \theta_2, \cdots, \theta_M] \in \mathbb{R}^M$ as the *measured* parameters used to build the transformation from Equation (3.1). In this work, we assume that the measured parameters are available from either known inputs to the mechanism, or can be measured using sensor feedback. It is possible to also perform the calibration without joint angle input by simultaneously estimating the joint angles as part of the calibration process [87].

In order to perform the calibration between the static camera and the dynamic camera, a fiducial marker is used to collect feature measurements in both cameras. Note that any marker, such as an AprilTag [88] or chess board is suitable, so long as the scale of the points can be determined using a target of known dimension. Measurements of the marker are taken from both cameras simultaneously, which requires that the two cameras share an overlapping field of view. Although it is possible to calibrate a multi-camera cluster with completely disjoint or non-overlapping fields of view, such a calibration requires motion of the camera cluster and tracking of natural feature points from non-fiducial sources [10, 9], which is left as direction of future work for the DCC case.

Using the measurements and known scale of the fiducial marker, it is possible for the observing camera to compute its 3D pose relative to the marker frame, $\mathcal{F}_{\mathtt{t}}$, using well studied techniques such as the perspective-n-point algorithm [89] or a bundle adjustment approach [6]. Given the pose of the observing camera relative to the marker frame, we determine the position of marker points relative to the camera frame. An example image of a DCC set-up and its associated frames and transformations is depicted in Figure 3.1.

For each instance where both the dynamic and static camera capture measurements to

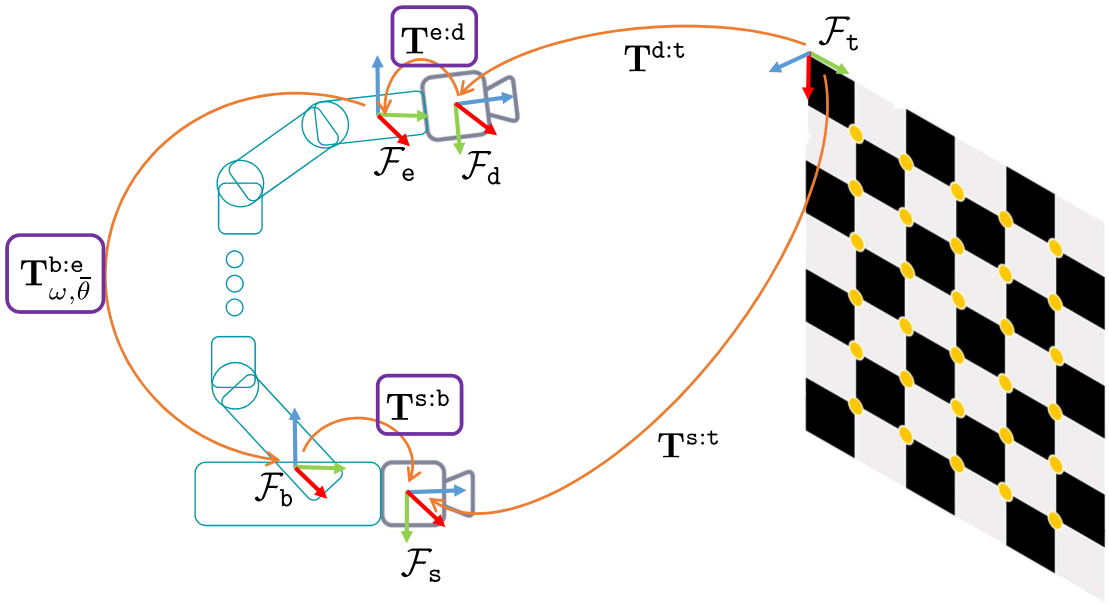
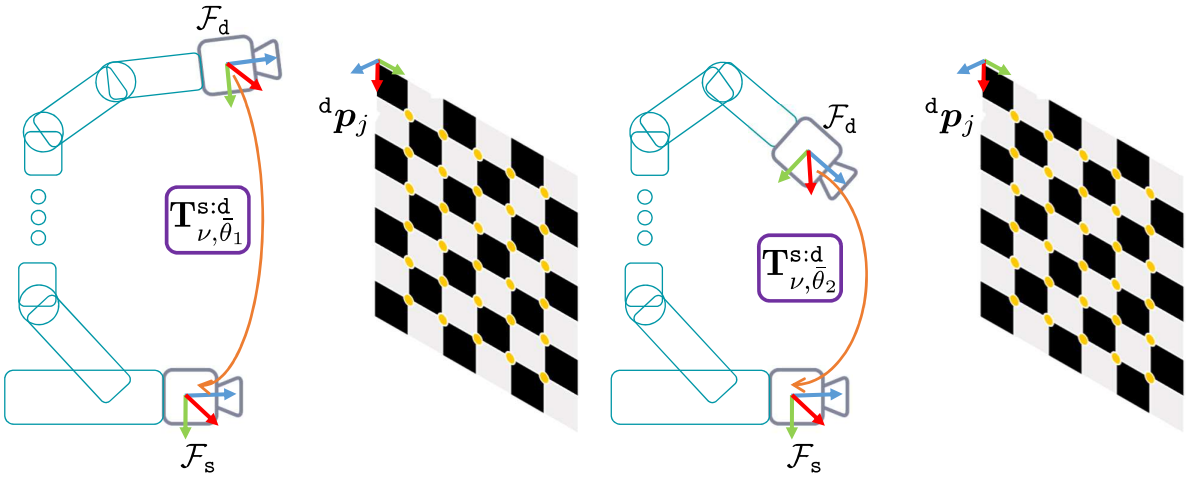


Figure 3.1: Example DCC set-up, where the unknown calibration quantities to be estimated are outlined in purple. The transformations from the target frame, $\mathcal{F}_{\mathrm{t}}$, to the observing cameras are computed using PnP or bundle-adjustment methods.

the fiducial marker, we define the i$^{\text{th}}$ *measurement set* as $Z_i = \{P_i^s, P_i^d, Q_i^s, Q_i^d, \bar{\theta}_i\}$, where P_i^s and P_i^d is the set of marker points defined in the frames of the static and dynamic cameras, respectively, Q_i^s and Q_i^d is the set of measurements to the marker points, as observed by the static and dynamic cameras, respectively, and $\bar{\theta}_i$ is the set of joint inputs for the mechanism at snapshot i. Note that the measurement sets only include corresponding points visible in both cameras, so consequently $|P_i^s| = |P_i^d| = |Q_i^s| = |Q_i^d|$. In order to produce a high quality calibration, multiple measurement sets need to be collected, while ensuring sufficient excitation of the joint inputs by collecting measurements from many different configurations of the dynamic camera. A sequence of measurement sets is visualized in Figure 3.2.

Using the measurement set and the transformation between camera frames, we can now define the reprojection error between the marker point j in the dynamic camera frame and the corresponding measured point in the static camera frame, for measurement set i, as



(a) Measurement Set 1

(b) Measurement Set 2

(c) Measurement Set 3

(d) Measurement Set N

Figure 3.2: An example sequence of DCC measurement configurations. The configuration space of the mechanism is excited in order to collect N different measurement sets used for the calibrations, as seen in figures (a) - (d). Note that for clarity, only the transformation from the dynamic to static camera is drawn.

$$e_j^s(\nu, \bar{\theta}_i) = z_j^s - \Psi^s(\mathbf{T}^{\mathtt{s:d}}_{\nu,\bar{\theta}_i}\, {}^{\mathtt{d}}\mathbf{p}_j) \tag{3.4}$$

where $z_j^s \in Q_i^s$ is the measurement of point j, from measurement set Q_i^s, observed in the static camera, and ${}^{\mathtt{d}}\mathbf{p}_j \in P_i^d$ is the 3D position of point j, from the point set P_i^d, as observed from the dynamic camera. Since both the dynamic and static camera observe the same marker at each snapshot, we can similarly compute the error for points observed in the static frame and projected into the dynamic frame,

$$e_j^d(\nu, \bar{\theta}_i) = z_j^d - \Psi^d((\mathbf{T}^{\mathtt{s:d}}_{\nu,\bar{\theta}_i})^{-1}\, {}^{\mathtt{s}}\mathbf{p}_j) \tag{3.5}$$

where $z_j^d \in Q_i^d$ is the measurement of point j, from measurement set Q_i^d, observed in the dynamic camera, and ${}^{\mathtt{s}}\mathbf{p}_j \in P_i^s$ is the 3D position of point j, from the point set P_i^s, as observed from the static camera. The total squared reprojection error as a function of the estimation parameters, $\Lambda_r(\nu) : \mathcal{S} \mapsto \mathbb{R}$ over all of the collected measurement sets, $\Gamma = \{Z_1, Z_2, \ldots, Z_k\}$, is defined as

$$\Lambda_r(\nu) = \sum_{Z_i \in \Gamma} \left[\sum_{j=1}^{|P_i^s|} e_j^d(\nu, \bar{\theta}_i)^T e_j^d(\nu, \bar{\theta}_i) + \sum_{j=1}^{|P_i^d|} e_j^s(\nu, \bar{\theta}_i)^T e_j^s(\nu, \bar{\theta}_i) \right] \tag{3.6}$$

Finally, Equation (3.6) is optimized in order to find the optimal parameters, ν^*, which minimize

$$\nu^* = \underset{\nu \in \mathcal{S}}{\operatorname{argmin}}\, \Lambda_r(\nu). \tag{3.7}$$

Note that (3.7) describes an unconstrained nonlinear optimization over manifold quantities, and is solved using the nonlinear least squares approach presented in Section 2.5.

3.2 Static-to-Dynamic Camera Calibration Using Pose-Loop Error

Since both the static camera and dynamic camera observe the same fiducial target, it is possible to formulate the calibration problem using a *pose-loop* error function. Let us denote the transformation from the fiducial target frame, $\mathcal{F}_{\mathtt{t}}$, to the static camera frame, $\mathcal{F}_{\mathtt{s}}$, as $\mathbf{T}^{\mathtt{s:t}} \in \mathbb{SE}(3)$. Similar to the reprojection error formation discussed in Section 3.1, the transformation, $\mathbf{T}^{\mathtt{s:t}}$, can be determined using the perspective-n-point algorithm [89] or a bundle adjustment approach [6]. As the dynamic camera also observes the same fiducial target, we can similarly compute the transformation from the target frame, $\mathcal{F}_{\mathtt{t}}$, to the dynamic camera frame, $\mathcal{F}_{\mathtt{d}}$, as $\mathbf{T}^{\mathtt{d:t}}$. Given the two observations of the fiducial target from the static and dynamic camera, we can generate a measurement of the dynamic to static camera transformation, $\tilde{\mathbf{T}}^{\mathtt{s:d}}$, which is given by

$$\tilde{\mathbf{T}}^{\mathtt{s:d}} = \mathbf{T}^{\mathtt{s:t}}(\mathbf{T}^{\mathtt{d:t}})^{-1}. \tag{3.8}$$

For the pose-loop calibration formulation, we shall define the i$^{\text{th}}$ *measurement set* as

$$B_i = \{\tilde{\mathbf{T}}_i^{\mathtt{s:d}}, \bar{\theta}_i\} \tag{3.9}$$

where $\tilde{\mathbf{T}}_i^{\mathtt{s:d}}$ is the measured dynamic to static camera transformation as computed using Equation (3.8) at snapshot i, and $\bar{\theta}_i$ is the set of joint inputs for the mechanism at the i^{th} configuration. Figure 3.3 depicts the pose-loop transformation constraint.

Using the measured transformation from Equation (3.8), we can now formulate an error function, $\varepsilon(\nu, \bar{\theta})$, which directly compares the measured dynamic to static camera transformation for the i^{th} measurement set, and the modeled transformation formulated

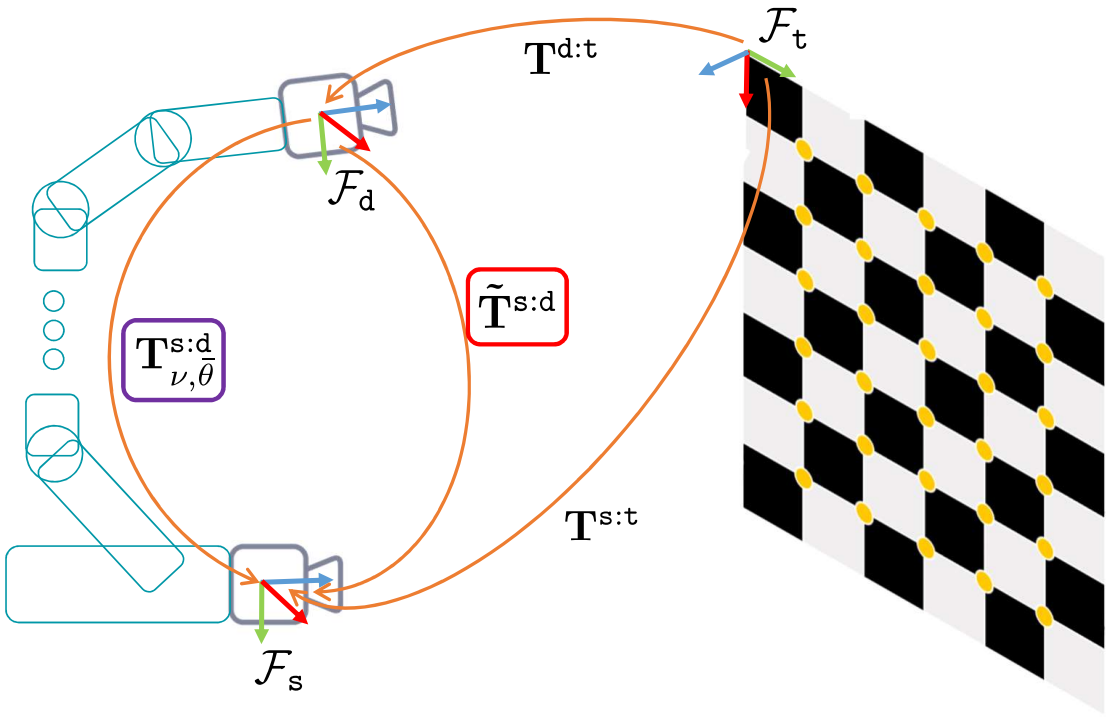


Figure 3.3: Example transformation loop for a measurement set. Since both the moving and static camera make observations to the same target, it is possible to construct the pose-loop measurement, $\tilde{\mathbf{T}}^{s:d}$.

in Equation (3.1), as

$$\varepsilon(\nu, \bar{\theta}_i) = \tilde{\mathbf{T}}_i^{\mathtt{s:d}} \boxminus \mathbf{T}_{\nu,\bar{\theta}_i}^{\mathtt{s:d}}. \tag{3.10}$$

The total squared pose-loop error as a function of the estimation parameters, $\Lambda_p(\nu) : \mathcal{S} \mapsto \mathbb{R}$, over the k collected measurement sets, $\mathit{\Gamma} = \{B_1, B_2, \ldots, B_k\}$, is defined as

$$\Lambda_p(\nu) = \sum_{B_i \in \mathit{\Gamma}} \varepsilon(\nu, \bar{\theta}_i)^T \varepsilon(\nu, \bar{\theta}_i) \tag{3.11}$$

Finally, to perform the calibration and determine the optimal parameters, ν^*, Equation (3.11) is optimized in order to find the parameters which minimize the total pose-loop error,

$$\nu^* = \underset{\nu \in \mathcal{S}}{\operatorname{argmin}} \, \Lambda_p(\nu), \tag{3.12}$$

which is solved using the manifold based nonlinear least squares optimization discussed in 2.5.

A key observation to note is that for the reprojection error approach, an error term is required for each observed point, whereas the pose-loop formulation only requires an error term for each measurement set. In the pose-loop formulation, the point observations are only used to calculate the transformations from the target to the static and dynamic cameras, $\mathbf{T}^{\mathtt{s:t}}$, and $\mathbf{T}^{\mathtt{d:t}}$. In this way, the pose-loop approach abstracts out the fiducial target point observations, and instead operates on the relative pose measurement between the static and dynamic camera. The benefit of this abstraction is the reduced number of row entries in the system Jacobian. The reprojection error formulation will add two row entries to the system Jacobian for each observed point, j, present within each measurement set, i. In contrast, the pose loop formulation will only add six rows per measurement set, i, as the relative pose error term from Equation (3.10) evaluates to a 6×1 vector. When

compared to the reprojection calibration approach, the reduced sized system Jacobian of the pose-loop formulation exhibits a simplified block-structure, which allows for a tractable analysis of calibration degeneracies.

3.3 Actuated-to-Actuated Camera Calibration

Compared to the static-to-dynamic calibration, the calibration of an dynamic-to-dynamic camera pair requires the calculation of an additional transform between the base frames of the each mechanism, $\mathbf{T}^{\mathsf{b_2:b_1}}$, as depicted in Figure 3.4. Suppose the camera pair consists of two cameras, Camera 1 and Camera 2. To determine the unknown transform, the camera pair is calibrated by first holding Camera 1 stationary using a static control input, $\bar{\theta}_1$, and performing the static-to-dynamic calibration by moving Camera 2, which results in the estimation of calibration parameters ν_2. The static-to-dynamic calibration is then performed again, except now holding Camera 2 stationary using a static control input, $\bar{\theta}_2$, and performing the calibration by moving Camera 1, which results in the estimation of calibration parameters ν_1

As illustrated in Figure 3.4, we can now define two equivalent transformation loops using the estimated parameters from the static-to-dynamic calibrations and the static control inputs $\bar{\theta}_1$ and $\bar{\theta}_2$,

$$\begin{aligned} \mathbf{T}^{\mathsf{b2:b2}} &= \mathbf{T}^{\mathsf{b_2:e_2}}_{\omega_2,\bar{\theta}_2} \mathbf{T}^{\mathsf{e_2:d_2}} (\mathbf{T}^{\mathsf{b_1:d_2}})^{-1} (\mathbf{T}^{\mathsf{b_2:b_1}})^{-1} \\ \mathbf{T}^{\mathsf{b1:b1}} &= \mathbf{T}^{\mathsf{b_1:e_1}}_{\omega_1,\bar{\theta}_1} \mathbf{T}^{\mathsf{e_1:d_1}} (\mathbf{T}^{\mathsf{b_2:d_1}})^{-1} \mathbf{T}^{\mathsf{b_2:b_1}} \end{aligned} \tag{3.13}$$

Using the transformation loops from (3.13), we now estimate the parameters of the unknown base to base transformation, $\mathbf{T}^{\mathsf{b_2:b_1}}$, such that $\mathbf{T}^{\mathsf{b2:b2}} = \mathbf{T}^{\mathsf{b1:b1}} = \mathbf{I}$. Let us define

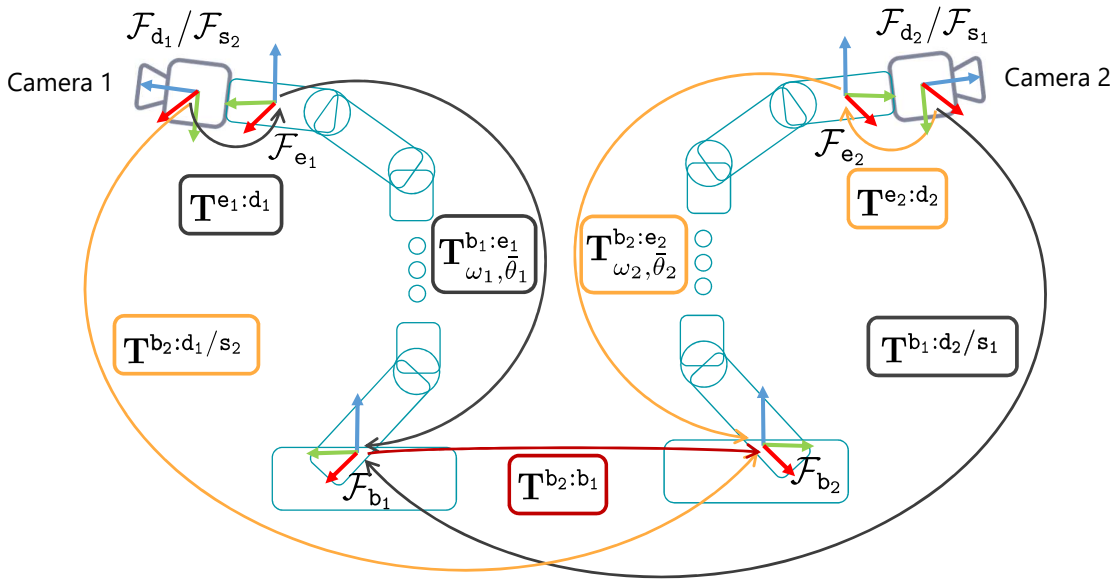


Figure 3.4: Frame transformations for actuated-to-actuated camera calibration case, drawn for the static control input configurations of $\bar{\theta}_1$ and $\bar{\theta}_2$. The grey arrows show the transforms which are calibrated when camera 1 is treated as the dynamic camera, and the orange arrows show the transforms which are calibrated when camera 2 is treated as the dynamic camera. The unknown transformation, $\mathbf{T}^{b_2:b_1}$, is depicted with the red arrow. After performing individual calibrations of the cameras, the unknown transform $\mathbf{T}^{b_2:b_1}$ can be solved by equating pose-loops from camera 1 and camera 2 which contain $\mathbf{T}^{b_2:b_1}$. Note that the notation, $\mathbf{T}^{b_2:d_1/s_2}$ indicates that the transformation can be interpreted as $\mathbf{T}^{b_2:d_1}$ or $\mathbf{T}^{b_2:s_2}$, depending on which camera is being held static.

the error term for the transformation loop for camera 1 and camera 2 as

$$\begin{aligned} \varepsilon_1(\mathbf{T}^{\mathtt{b}_2:\mathtt{b}_1}) &= \mathbf{I} \boxminus \mathbf{T}^{\mathtt{b1:b1}} \\ \varepsilon_2(\mathbf{T}^{\mathtt{b}_2:\mathtt{b}_1}) &= \mathbf{I} \boxminus \mathbf{T}^{\mathtt{b2:b2}} \end{aligned} \tag{3.14}$$

Finally, the cost function, $\Lambda(\mathbf{T}^{\mathtt{b}_2:\mathtt{b}_1}) : \mathbb{SE}(3) \mapsto \mathbb{R}$, which penalizes the error in the loop transformations from (3.13), is given as,

$$\Lambda(\mathbf{T}^{\mathtt{b}_2:\mathtt{b}_1}) = \begin{bmatrix} \varepsilon_1(\mathbf{T}^{\mathtt{b}_2:\mathtt{b}_1}) \\ \varepsilon_2(\mathbf{T}^{\mathtt{b}_2:\mathtt{b}_1}) \end{bmatrix}^T \begin{bmatrix} \varepsilon_1(\mathbf{T}^{\mathtt{b}_2:\mathtt{b}_1}) \\ \varepsilon_2(\mathbf{T}^{\mathtt{b}_2:\mathtt{b}_1}) \end{bmatrix}, \tag{3.15}$$

which can be optimized using the manifold based nonlinear least squares approach outlined in Section 2.5 to find the unknown transformation $\mathbf{T}^{\mathtt{b}_2:\mathtt{b}_1}$. Once the base mechanism to base mechanism transformation is determined, the calibration is complete, as the forward kinematics between both actuated cameras are fully defined.

3.4 Degeneracy Analysis and Minimal Parameterization

In this section, we will present degeneracies that arise from over-parameterization of the calibration problem. In the formulations presented in Section 3.1 and 3.2, we describe the calibration parameters from Equation (3.3) as one six degree of freedom transform from the base link frame of the mechanism, $\mathcal{F}_{\mathtt{b}}$, to the static camera frame, $\mathcal{F}_{\mathtt{s}}$, another six degree of freedom transform from the end effector frame of the mechanism, $\mathcal{F}_{\mathtt{e}}$, to the dynamic camera frame, $\mathcal{F}_{\mathtt{d}}$, and $3M$ DH parameters for a mechanism with M links. Note that only three of the four possible DH parameters are estimated per link, as θ_i parameter is measured using joint or encoder feedback.

As discussed in Section 2.5, the system Hessian matrix is approximated by

$$\left.\frac{\partial^2 \Lambda(\breve{\mathbf{x}}_k \boxplus \varphi_k)}{\partial \varphi_k^2}\right|_{\varphi_k=0} \approx \mathbf{J}^T \Omega \mathbf{J}. \tag{3.16}$$

In order for the system to possess a unique solution, the Hessian, $\mathbf{J}^T\Omega\mathbf{J}$ must be invertible, which is equivalent to the Jacobian matrix, $\mathbf{J}$, having full rank [5]. We will now show that for the DCC calibration problem, a subset of the estimation parameters are redundant, and their inclusion in the calibration process results in a degenerate system that cannot be solved. Our approach will be to demonstrate that the system Jacobian column corresponding to the redundant parameters is in fact, a linear combination of other columns which exist in the Jacobian matrix, causing the system Jacobian matrix be rank deficient. A example of the structure of the DCC system Jacobian is depicted in Figure 3.5

We will first briefly present two derivative functions required to perform the degeneracy analysis. Then, we will present the degeneracies resulting from redundant parameters in the system. Note that the degeneracy analysis will be performed using the pose-loop formulation of the calibration problem, as the pose-loop formulation is functionally equivalent to the reprojection formulation, but benefits from a Jacobian structure with fewer rows, which facilitates degeneracy analysis.

3.4.1 Jacobian of Transform Function and DH Matrix

The following Jacobians of the transformation composition function and box-minus function will be required to perform the degeneracy analysis for the calibration system. To that end, we will also present the derivatives for the DH transformation matrix with respect to the d_i, a_i, and α_i parameters. Jacobians that are required, but not specific to the development of the degeneracy analysis, are derived in Appendix B.

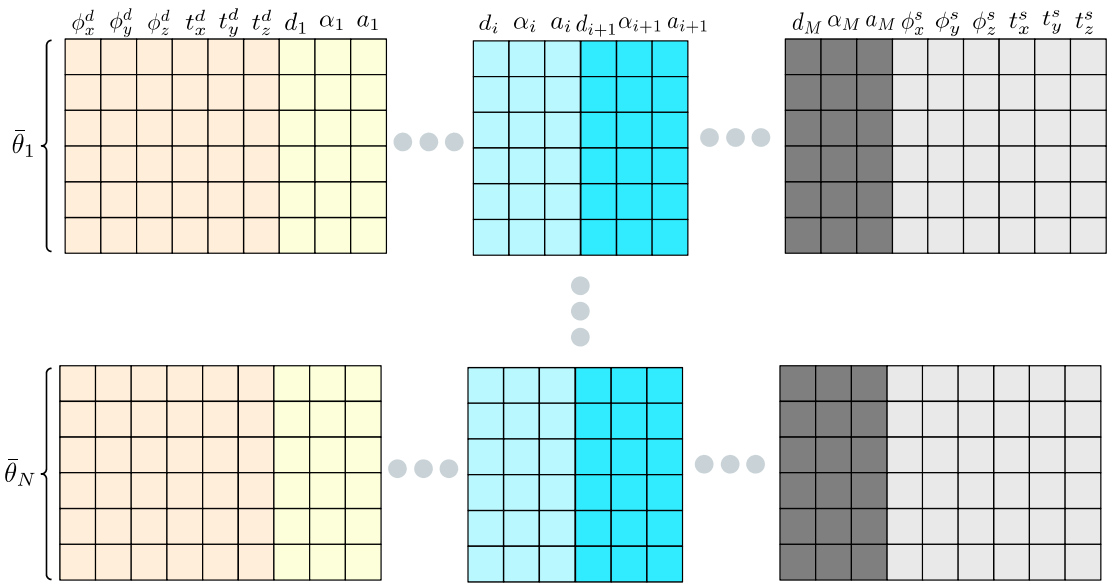


Figure 3.5: A visualization of the general system Jacobian for the DCC calibration problem. The figure depicts the Jacobian of the error function (3.10), with respect to the calibration parameters, ν, stacked over N measurement sets. Each column in the matrix corresponds to a different calibration parameter. Our strategy for degeneracy analysis will be to show that some of the Jacobian columns corresponding to a degenerate parameter can be expressed as linear combination of other parameter columns. Note that the contribution from each measurement set stacks along the rows blocks of this Jacobian, and thus our degeneracy analysis must be invariant to the selected measurement configuration.

Jacobian of composition function: Suppose we have two transformation matrices, $\mathbf{T}^{\mathsf{b:c}} \in \mathbb{SE}(3)$ and $\mathbf{T}^{\mathsf{a:b}} \in \mathbb{SE}(3)$, with their resulting composition

$$\mathbf{T}^{\mathsf{a:c}} = \mathbf{T}^{\mathsf{a:b}}\mathbf{T}^{\mathsf{b:c}}. \tag{3.17}$$

Then, the 6×6 Jacobian matrices, representing the derivatives of Equation (3.17), with respect to the left and right entities are computed as,

$$\frac{\partial \mathbf{T}^{\mathsf{a:c}}}{\mathbf{T}^{\mathsf{a:b}}} = \left[\begin{array}{c|c} I & 0 \\ \hline -[\mathbf{R}^{\mathsf{a:b}}\mathbf{t}^{\mathsf{b:c}}]^{\wedge} & I \end{array}\right] \tag{3.18}$$

$$\frac{\partial \mathbf{T}^{\mathsf{a:c}}}{\mathbf{T}^{\mathsf{b:c}}} = \left[\begin{array}{c|c} \mathbf{R}^{\mathsf{a:b}} & 0 \\ \hline 0 & \mathbf{R}^{\mathsf{a:b}} \end{array}\right] \tag{3.19}$$

Jacobian of box-minus function: Suppose we have two transformation matrices, $\mathbf{T}^{\mathsf{a:c}} \in \mathbb{SE}(3)$ and $\mathbf{T}^{\mathsf{b:c}} \in \mathbb{SE}(3)$, with their resulting box-minus difference

$$\varphi^{\mathsf{a:b}} = \mathbf{T}^{\mathsf{a:c}} \boxminus \mathbf{T}^{\mathsf{b:c}} \tag{3.20}$$

Then, the 6×6 Jacobian matrices, representing the derivatives of Equation (3.20), with respect to the left and right entities are computed as,

$$\frac{\partial \varphi^{\mathsf{a:b}}}{\mathbf{T}^{\mathsf{a:c}}} = \left[\begin{array}{c|c} \mathcal{J}_{\boxminus}^{\mathsf{a:c}} & 0 \\ \hline 0 & I \end{array}\right] \tag{3.21}$$

$$\frac{\partial \varphi^{\mathsf{a:b}}}{\mathbf{T}^{\mathsf{b:c}}} = \left[\begin{array}{c|c} \mathcal{J}_{\boxminus}^{\mathsf{b:c}} & 0 \\ \hline 0 & -I \end{array}\right] \tag{3.22}$$

where $\mathcal{J}_{\boxminus}^{\mathsf{a:c}}$ and $\mathcal{J}_{\boxminus}^{\mathsf{b:c}}$ are the derivatives with respect to the left and right entities of the

box-minus operation on rotation matrices, as presented in B.

Derivatives of the DH transformation matrix: In this section, we shall derive the derivative of DH parameter matrix from Equation (2.32), with respect to the d_i , a_i, and α_i parameters. Note that the derivative with respect to θ_i is not required, as we assume that the joint angles are measured quantities and thus do not need to be estimated.

To determine the derivative of $\mathbf{T}_{\omega_i}^{\mathtt{i:i-1}}$ with respect to the d_i parameter, we first express the DH matrix transform chain from Equation (2.27) as

$$\mathbf{T}_{\omega_i,\theta_i}^{\mathtt{i:i-1}} = \mathbf{T}_{\theta_i}^{\mathtt{i:s_i}} \mathbf{T}_{d_i,a_i,\alpha_i}^{\mathtt{s_i:i-1}}. \tag{3.23}$$

Recall that using the definition of derivative with the box-plus and box-minus operator from Equation (2.25), we can compute the derivative of Equation (3.23) with respect to the right side transformation as,

$$\left[\frac{\partial \mathbf{T}_{\omega_i,\theta_i}^{\mathtt{i:i-1}}}{\partial \mathbf{T}_{d_i,a_i,\alpha_i}^{\mathtt{s_i:i-1}}}\right]_i = \lim_{\varphi_j^{\bar{\mathtt{s}}_\mathtt{i}:\mathtt{s_i}} \to 0} \frac{\mathbf{T}_{\theta_i}^{\mathtt{i:s_i}} (\mathbf{T}_{d_i,a_i,\alpha_i}^{\mathtt{s_i:i-1}} \boxplus \varphi_j^{\bar{\mathtt{s}}_\mathtt{i}:\mathtt{s_i}}) \boxminus \mathbf{T}_{\theta_i}^{\mathtt{i:s_i}} \mathbf{T}_{d_i,a_i,\alpha_i}^{\mathtt{s_i:i-1}}}{|\varphi_j^{\bar{\mathtt{s}}_\mathtt{i}:\mathtt{s_i}}|}. \tag{3.24}$$

If we inspect the sixth element of the perturbation vector

$$\varphi_6^{\bar{\mathtt{s}}_\mathtt{i}:\mathtt{s_i}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \delta_d \end{bmatrix}, \tag{3.25}$$

and apply it to the term $\mathbf{T}_{d_i,a_i,\alpha_i}^{\mathtt{s_i:i-1}} \boxplus \varphi_6^{\bar{\mathtt{s}}_\mathtt{i}:\mathtt{s_i}}$ from Equation (3.24), we get,

$$\mathbf{T}^{\mathtt{s_i:i}}_{d_i,a_i,\alpha_i} \boxplus \varphi_6^{\mathtt{\bar{s}_i:s_i}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \delta_d \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{T}^{\mathtt{s_i:i}}_{d_i,a_i,\alpha_i} \tag{3.26}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \delta_d \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{T}^{\mathtt{r_i:i}}_{a_i,\alpha_i} \tag{3.27}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i + \delta_d \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{T}^{\mathtt{r_i:i}}_{a_i,\alpha_i} \tag{3.28}$$

The key observation from Equation (3.28) is that the perturbation from the sixth element of the perturbation vector from Equation (3.24), exactly maps to a perturbation of the d_i parameter in the DH matrix $\mathbf{T}^{\mathtt{i:i-1}}_{\omega_i}$. Thus the derivative of $\mathbf{T}^{\mathtt{i:i-1}}_{\omega_i}$ with respect to the d_i parameter is simply the sixth column of the composition derivative with respect to the right side element, as presented in Equation (3.19),

$$\frac{\partial \mathbf{T}^{\mathtt{i:i-1}}_{\omega_i,\theta_i}}{\partial d_i} = \left[\frac{\partial \mathbf{T}^{\mathtt{i:i-1}}_{\omega_i,\theta_i}}{\partial \mathbf{T}^{\mathtt{s_i:i}}_{d_i,a_i,\alpha_i}} \right]_6 \tag{3.29}$$

$$= \left[\begin{array}{c|c} \mathbf{R}^{\mathtt{i:s_i}}_{\theta_i} & 0 \\ \hline 0 & \mathbf{R}^{\mathtt{i:s_i}}_{\theta_i} \end{array} \right]_6 \tag{3.30}$$

where the rotation matrix $\mathbf{R}_{\theta_i}^{\mathtt{i:s_i}}$ is the rotation component of $\mathbf{T}_{\theta_i}^{\mathtt{i:s_i}}$, and is simply a function of the joint angle parameter, θ_i,

$$\mathbf{R}_{\theta_i}^{\mathtt{i:s_i}} = \begin{bmatrix} c(\theta_i) & -s(\theta_i) & 0 \\ s(\theta_i) & c(\theta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.31}$$

and the sixth column of Equation (3.30) evaluates to

$$\frac{\partial \mathbf{T}_{\omega_i,\theta_i}^{\mathtt{i:i-1}}}{\partial d_i} = \left[\begin{array}{c|c} \mathbf{R}_{\theta_i}^{\mathtt{i:s_i}} & 0 \\ \hline 0 & \mathbf{R}_{\theta_i}^{\mathtt{i:s_i}} \end{array}\right]_6 \tag{3.32}$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{3.33}$$

Using a similar strategy, it is possible to show that the derivative of the DH matrix with respect to the a_i parameter is the fourth column of the composition derivative with respect to the right side transformation of

$$\mathbf{T}_{\omega_i,\theta_i}^{\mathtt{i:i-1}} = \mathbf{T}_{\theta_i,d_i}^{\mathtt{i:r_i}} \mathbf{T}_{a_i,\alpha_i}^{\mathtt{r_i:i-1}}, \tag{3.34}$$

and that the derivative of the DH matrix with respect to the α_i parameter is the first

column of the composition derivative with respect to the right side transformation of

$$\mathbf{T}^{\mathtt{i:i-1}}_{\omega_i,\theta_i} = \mathbf{T}^{\mathtt{i:q_i}}_{\theta_i,d_i,a_i}\mathbf{T}^{\mathtt{q_i:i-1}}_{\alpha_i} \tag{3.35}$$

The derivatives are given as

$$\frac{\partial \mathbf{T}^{\mathtt{i:i-1}}_{\omega_i,\theta_i}}{\partial a_i} = \left[\frac{\partial \mathbf{T}^{\mathtt{i:i-1}}_{\omega_i,\theta_i}}{\partial \mathbf{T}^{\mathtt{r_i:i-1}}_{a_i,\alpha_i}}\right]_4 \tag{3.36}$$

$$= \left[\begin{array}{c|c} \mathbf{R}^{\mathtt{i:r_i}}_{\theta_i} & 0 \\ \hline 0 & \mathbf{R}^{\mathtt{i:r_i}}_{\theta_i} \end{array}\right]_4 \tag{3.37}$$

$$= \left[\begin{array}{c} 0 \\ \hline \left[\mathbf{R}^{\mathtt{i:r_i}}_{\theta_i}\right]_1 \end{array}\right] \tag{3.38}$$

$$\frac{\partial \mathbf{T}^{\mathtt{i:i-1}}_{\omega_i,\theta_i}}{\partial \alpha_i} = \left[\frac{\partial \mathbf{T}^{\mathtt{i:i-1}}_{\omega_i,\theta_i}}{\partial \mathbf{T}^{\mathtt{q_i:i-1}}_{\alpha_i}}\right]_1 \tag{3.39}$$

$$= \left[\begin{array}{c|c} \mathbf{R}^{\mathtt{i:q_i}}_{\theta_i} & 0 \\ \hline 0 & \mathbf{R}^{\mathtt{i:q_i}}_{\theta_i} \end{array}\right]_1 \tag{3.40}$$

$$= \left[\begin{array}{c} \left[\mathbf{R}^{\mathtt{i:q_i}}_{\theta_i}\right]_1 \\ \hline 0 \end{array}\right] \tag{3.41}$$

3.4.2 DCC Error Term Jacobian:

Recall that the transformation chain for the DCC calibration problem has the form

$$\mathbf{T}^{\mathtt{s:d}}_{\nu,\bar{\theta}} = \mathbf{T}^{\mathtt{s:b}}\mathbf{T}^{\mathtt{b:e_{M-1}}}_{\omega_M,\theta_M}\cdots\mathbf{T}^{\mathtt{e_2:e_1}}_{\omega_2,\theta_2}\mathbf{T}^{\mathtt{e_1:e}}_{\omega_1,\theta_1}\mathbf{T}^{\mathtt{e:d}}, \tag{3.42}$$

and that for each measurement set i, we construct the pose-loop error equation

$$\varepsilon(\nu, \bar{\theta}_i) = \tilde{\mathbf{T}}_i^{\mathtt{s:d}} \boxminus \mathbf{T}_{\nu,\bar{\theta}_i}^{\mathtt{s:d}}, \tag{3.43}$$

and the inclusion of each measurement to the calibration problem adds a block row to the system Jacobian matrix, as seen in Figure 3.5.

Let us inspect the derivative of the error function (3.43) with respect to the unknown calibration transformation, $\mathbf{T}_{\nu,\bar{\theta}_i}^{\mathtt{s:d}}$. Using the derivative function for box-minus with respect to the right-side transforms from Equation (3.22), we get,

$$\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}_{\nu,\bar{\theta}}^{\mathtt{s:d}}} = \left[\begin{array}{c|c} \mathcal{J}_{\boxminus}^{\mathtt{s:d}} & 0 \\ \hline 0 & -I \end{array}\right]. \tag{3.44}$$

Equation 3.44 will be used repeatedly in the latter sections to evaluate the Jacobian with respect to the parameters used to build the calibration transform, $\mathbf{T}_{\nu,\bar{\theta}_i}^{\mathtt{s:d}}$.

3.4.3 Degeneracies related to static-camera-to-base transform

First, let us inspect the derivative of the error function (3.43) with respect to the base frame to static frame transformation. Using chain rule,

$$\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathtt{s:b}}} = \frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}_{\nu,\bar{\theta}}^{\mathtt{s:d}}} \frac{\partial \mathbf{T}_{\nu,\bar{\theta}}^{\mathtt{s:d}}}{\partial \mathbf{T}^{\mathtt{s:b}}}. \tag{3.45}$$

Using the derivative for box-minus with respect to the right-side of the error equation, as given by Equation (3.44), and the derivative of the transformation composition with

respect to the left side transform (3.18), we get,

$$\frac{\partial \mathbf{T}^{\mathtt{s:d}}_{\nu,\bar{\theta}}}{\partial \mathbf{T}^{\mathtt{s:b}}} = \left[\begin{array}{c|c} I & 0 \\ \hline -[\mathbf{R}^{\mathtt{s:b}}\mathbf{t}^{\mathtt{b:d}}]^{\wedge} & I \end{array}\right], \tag{3.46}$$

resulting in,

$$\frac{\partial \varepsilon(\nu,\bar{\theta}_i)}{\partial \mathbf{T}^{\mathtt{s:b}}} = \left[\begin{array}{c|c} \mathcal{J}^{\mathtt{s:d}}_{\boxminus} & 0 \\ \hline 0 & -I \end{array}\right] \left[\begin{array}{c|c} I & 0 \\ \hline -[\mathbf{R}^{\mathtt{s:b}}\mathbf{t}^{\mathtt{b:d}}]^{\wedge} & I \end{array}\right] \tag{3.47}$$

$$= \left[\begin{array}{c|c} \mathcal{J}^{\mathtt{s:d}}_{\boxminus} & 0 \\ \hline -[\mathbf{R}^{\mathtt{s:b}}\mathbf{t}^{\mathtt{b:d}}]^{\wedge} & -I \end{array}\right]. \tag{3.48}$$

Degeneracy 1: Inclusion of the d_M parameter: We will now show that the inclusion of the offset parameter, d_M, which is used to construct the transformation $\mathbf{T}^{\mathtt{b:e_{M-1}}}_{\omega_M,\theta_M}$, results in a degeneracy of the calibration system.

Let us inspect the derivative of the loop-error Equation (3.43) with respect to the offset parameter d_M,

$$\frac{\partial \varepsilon(\nu,\bar{\theta}_i)}{\partial d_M} = \frac{\partial \varepsilon(\nu,\bar{\theta}_i)}{\partial \mathbf{T}^{\mathtt{s:d}}_{\nu,\bar{\theta}}} \frac{\partial \mathbf{T}^{\mathtt{s:d}}_{\nu,\bar{\theta}}}{\partial \mathbf{T}^{\mathtt{b:d}}} \frac{\partial \mathbf{T}^{\mathtt{b:d}}}{\partial \mathbf{T}^{\mathtt{b:e_{M-1}}}_{\omega_M,\theta_M}} \frac{\partial \mathbf{T}^{\mathtt{b:e_{M-1}}}_{\omega_M,\theta_M}}{\partial d_M}. \tag{3.49}$$

Using the derivative functions from Equations (3.44), (3.18), and the derivative of the

DH matrix with respect to the d_i parameter, given in Equation (3.32), we can compute

$$\frac{\partial \mathbf{T}^{\mathtt{s:d}}_{\nu,\bar{\theta}}}{\partial \mathbf{T}^{\mathtt{b:d}}} = \left[\begin{array}{c|c} \mathbf{R}^{\mathtt{s:b}} & 0 \\ \hline 0 & \mathbf{R}^{\mathtt{s:b}} \end{array}\right] \tag{3.50}$$

$$\frac{\partial \mathbf{T}^{\mathtt{b:d}}}{\partial \mathbf{T}^{\mathtt{b:e_{M-1}}}_{\omega_M,\theta_M}} = \left[\begin{array}{c|c} I & 0 \\ \hline -[\mathbf{R}^{\mathtt{b:e_{M-1}}}\mathbf{t}^{\mathtt{e_{M-1}:d}}]^{\wedge} & I \end{array}\right] \tag{3.51}$$

$$\frac{\partial \mathbf{T}^{\mathtt{b:e_{M-1}}}_{\omega_M,\theta_M}}{\partial d_M} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{3.52}$$

we get,

$$\frac{\partial \varepsilon(\nu,\bar{\theta}_i)}{\partial d_M} = \left[\begin{array}{c} 0 \\ \hline -\mathbf{R}^{\mathtt{s:b}} \end{array}\right]_3 \tag{3.53}$$

Inspecting the last three columns of the Jacobian $\frac{\partial \varepsilon(\nu,\bar{\theta}_i)}{\partial \mathbf{T}^{\mathtt{s:b}}}$ from Equation (3.48), observe that it is possible to write $\frac{\partial \varepsilon(\nu,\bar{\theta}_i)}{\partial \mathbf{T}^{\mathtt{s:b}}}$ as a linear combination of the elements of $\frac{\partial \varepsilon(\nu,\bar{\theta}_i)}{\partial d_M}$,

$$\left[\begin{array}{c} 0 \\ \hline -I \end{array}\right] \lambda = \left[\begin{array}{c} 0 \\ \hline -\mathbf{R}^{\mathtt{s:b}} \end{array}\right]_3, \tag{3.54}$$

where λ is a 3×1 column vector. By setting λ as

$$\lambda = \left[\begin{array}{c} 0 \\ \hline \mathbf{R}^{\mathtt{s:b}} \end{array}\right]_3, \tag{3.55}$$

which demonstrates that the inclusion of the d_M parameter in the calibration formulation results in a degeneracy. Intuitively, this degeneracy is caused by the freedom with which the base frame of the mechanism, $\mathcal{F}_{\mathtt{b}}$, can be placed. In accordance with the DH-convention described in Section 2, the only constraint on the base frame is that its z-axis must coincide with the rotation axis of the M^{th} joint. Thus, the origin of $\mathcal{F}_{\mathtt{b}}$ can be placed anywhere along the M^{th} joint rotation axis. This degree of freedom for the movement of $\mathcal{F}_{\mathtt{b}}$ along the rotation axis is precisely the same as the degree-of-freedom offered by the d_M parameter, and thus the d_M parameter and the translation component of $\mathbf{T}^{\mathtt{s:b}}$ are ambiguous.

It is important to note that the rotation $\mathbf{R}^{\mathtt{s:b}}$ remains constant as additional measurement sets are added, as it does not depend on any of the joint angle inputs. Thus, even as multiple row-blocks are added to the system Jacobian due to measurements from multiple mechanism configurations, the Jacobian column associated with the d_M parameter is comprised of a stack of repeated vectors corresponding to Equation (3.53). This result implies that the degeneracy is invariant to the measurement configurations collected, and is fundamental to the problem formulation. The system Jacobian as it pertains to this degeneracy is visualized in Figure 3.6.

3.4.4 Degeneracies related to end-effector-to-dynamic-camera transform

Let us first inspect the derivative of the transformation chain of Equation (3.1) with respect to the transformation from the dynamic camera frame, $\mathcal{F}_{\mathtt{d}}$, to the end-effector frame $\mathcal{F}_{\mathtt{e}}$,

$$\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathtt{e:d}}} = \frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathtt{s:d}}_{\nu,\bar{\theta}}} \frac{\partial \mathbf{T}^{\mathtt{s:d}}_{\nu,\bar{\theta}}}{\partial \mathbf{T}^{\mathtt{e:d}}}. \tag{3.56}$$

Using the derivative functions for box-minus with respect to the right-side transforms Equation (3.44), and transformation composition with respect to the right side transform

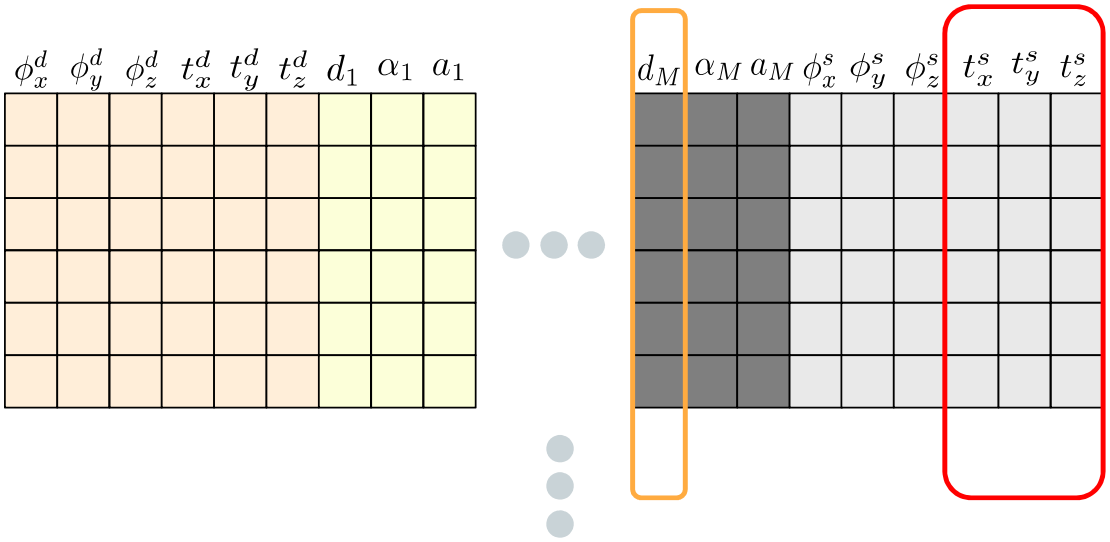


Figure 3.6: A visualization of the system Jacobian for the DCC calibration problem and the degeneracy caused by the inclusion of the d_M parameter. The Jacobian column of the d_M parameter can be expressed as a linear combination of the static-camera-to-base translation parameters.

from Equation (3.19), we get,

$$\frac{\partial \mathbf{T}_{\nu,\bar{\theta}}^{\mathsf{s:d}}}{\partial \mathbf{T}^{\mathsf{e:d}}} = \left[\begin{array}{c|c} \mathbf{R}^{\mathsf{s:e}} & 0 \\ \hline 0 & \mathbf{R}^{\mathsf{s:e}} \end{array}\right] \tag{3.57}$$

resulting in,

$$\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{e:d}}} = \left[\begin{array}{c|c} \mathcal{J}_{\boxminus}^{\mathsf{s:d}} \mathbf{R}^{\mathsf{s:e}} & 0 \\ \hline 0 & -\mathbf{R}^{\mathsf{s:e}} \end{array}\right]. \tag{3.58}$$

Degeneracy 2: Inclusion of the α_1 parameter: The derivative of the transformation chain, given in Equation (3.1), with respect to the α_1 parameter can be expressed as,

$$\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \alpha_1} = \frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}_{\nu,\bar{\theta}}^{\mathsf{s:d}}} \frac{\partial \mathbf{T}_{\nu,\bar{\theta}}^{\mathsf{s:d}}}{\partial \mathbf{T}^{\mathsf{e_1:d}}} \frac{\partial \mathbf{T}^{\mathsf{e_1:d}}}{\partial \mathbf{T}_{\omega_1,\theta_1}^{\mathsf{e_1:e}}} \frac{\partial \mathbf{T}_{\omega_1,\theta_1}^{\mathsf{e_1:e}}}{\partial \alpha_1}. \tag{3.59}$$

Using the derivative functions given by Equations (3.44), (3.18), (3.19), and the derivative of the DH matrix with respect to the α_1 parameter, given by Equation (3.39), we can compute

$$\frac{\partial \mathbf{T}^{\mathsf{s:d}}_{\nu,\bar{\theta}}}{\partial \mathbf{T}^{\mathsf{e_1:d}}} = \left[\begin{array}{c|c} \mathbf{R}^{\mathsf{s:e_1}} & 0 \\ \hline 0 & \mathbf{R}^{\mathsf{s:e_1}} \end{array}\right] \tag{3.60}$$

$$\frac{\partial \mathbf{T}^{\mathsf{e_1:d}}}{\partial \mathbf{T}^{\mathsf{e_1:e}}_{\omega_1,\theta_1}} = \left[\begin{array}{c|c} I & 0 \\ \hline -[\mathbf{R}^{\mathsf{e_1:e}}\mathbf{t}^{\mathsf{e:d}}]^\wedge & I \end{array}\right] \tag{3.61}$$

$$\frac{\partial \mathbf{T}^{\mathsf{e_1:e}}_{\omega_1,\theta_1}}{\partial \alpha_1} = \left[\begin{array}{c} \left[\mathbf{R}^{\mathsf{e_1:e}}_{\alpha_1,\theta_1}\right]_1 \\ \hline 0 \end{array}\right], \tag{3.62}$$

which evaluates to

$$\frac{\partial \varepsilon(\nu,\bar{\theta}_i)}{\partial \alpha_1} = \left[\begin{array}{c} \mathcal{J}^{\mathsf{s:d}}_{\boxminus}\mathbf{R}^{\mathsf{s:e_1}}\left[\mathbf{R}^{\mathsf{e_1:e}}_{\alpha_1,\theta_1}\right]_1 \\ \hline -\mathbf{R}^{\mathsf{s:e_1}}[\mathbf{R}^{\mathsf{e_1:e}}\mathbf{t}^{\mathsf{e:d}}]^\wedge\left[\mathbf{R}^{\mathsf{e_1:e}}_{\alpha_1,\theta_1}\right]_1 \end{array}\right] \tag{3.63}$$

$$= \left[\begin{array}{c} \mathcal{J}^{\mathsf{s:d}}_{\boxminus}\left[\mathbf{R}^{\mathsf{s:e}}\right]_1 \\ \hline -\mathbf{R}^{\mathsf{s:e_1}}[\mathbf{R}^{\mathsf{e_1:e}}\mathbf{t}^{\mathsf{e:d}}]^\wedge\left[\mathbf{R}^{\mathsf{e_1:e}}_{\alpha_1,\theta_1}\right]_1 \end{array}\right]. \tag{3.64}$$

Let us denote the bottom partition of $\frac{\partial \varepsilon(\nu,\bar{\theta}_i)}{\partial \alpha_1}$ as $\left(\frac{\partial \varepsilon(\nu,\bar{\theta}_i)}{\partial \alpha_1}\right)_b = -\mathbf{R}^{\mathsf{s:e_1}}[\mathbf{R}^{\mathsf{e_1:e}}\mathbf{t}^{\mathsf{e:d}}]^\wedge\left[\mathbf{R}^{\mathsf{e_1:e}}_{\alpha_1,\theta_1}\right]_1$, which can be simplified as,

$$\left(\frac{\partial \varepsilon(\nu,\bar{\theta}_i)}{\partial \alpha_1}\right)_b = -[\mathbf{R}^{\mathsf{s:e}}\mathbf{t}^{\mathsf{e:d}}]^\wedge\mathbf{R}^{\mathsf{s:e_1}}\left[\mathbf{R}^{\mathsf{e_1:e}}_{\alpha_1,\theta_1}\right]_1 \tag{3.65}$$

$$= -[\mathbf{R}^{\mathsf{s:e}}\mathbf{t}^{\mathsf{e:d}}]^\wedge\left[\mathbf{R}^{\mathsf{s:e}}\right]_1, \tag{3.66}$$

by identity (A.3), and can be further simplified to

$$-\mathbf{R}^{\mathsf{s:e}}[\mathbf{t}^{\mathsf{e:d}}]^\wedge(\mathbf{R}^{\mathsf{s:e}})^T\left[\mathbf{R}^{\mathsf{s:e}}\right]_1 \tag{3.67}$$

by identity (A.2), resulting in

$$= -\mathbf{R}^{\mathtt{s:e}}[\mathbf{t}^{\mathtt{e:d}}]^{\wedge} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{3.68}$$

$$= -\mathbf{R}^{\mathtt{s:e}} \begin{bmatrix} 0 \\ \mathbf{t}_z^{\mathtt{e:d}} \\ \mathbf{t}_y^{\mathtt{e:d}} \end{bmatrix}. \tag{3.69}$$

Finally, through comparison of (3.69) and (3.58) we see that

$$\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \alpha_1} = \left[\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathtt{e:d}}}\right]_1 + \left[\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathtt{e:d}}}\right]_{3:6} \begin{bmatrix} 0 \\ \mathbf{t}_z^{\mathtt{e:d}} \\ \mathbf{t}_y^{\mathtt{e:d}} \end{bmatrix}. \tag{3.70}$$

Notice that the vector,

$$\begin{bmatrix} 0 \\ \mathbf{t}_z^{\mathtt{e:d}} \\ \mathbf{t}_y^{\mathtt{e:d}} \end{bmatrix}, \tag{3.71}$$

is composed of translation elements of the transformation from the dynamic camera to the end effector frame, which do not change with measurement configurations. Thus the linear combination of (3.70) is present for the row-blocks added to the system Jacobian by all the measurement sets, and the system experiences a degeneracy if the α_1 parameter is included as part of the calibration state. The system Jacobian as it pertains to this degeneracy is visualized in Figure 3.7

Degeneracy 3: Inclusion of the a_1 parameter: Let us compute the derivative of

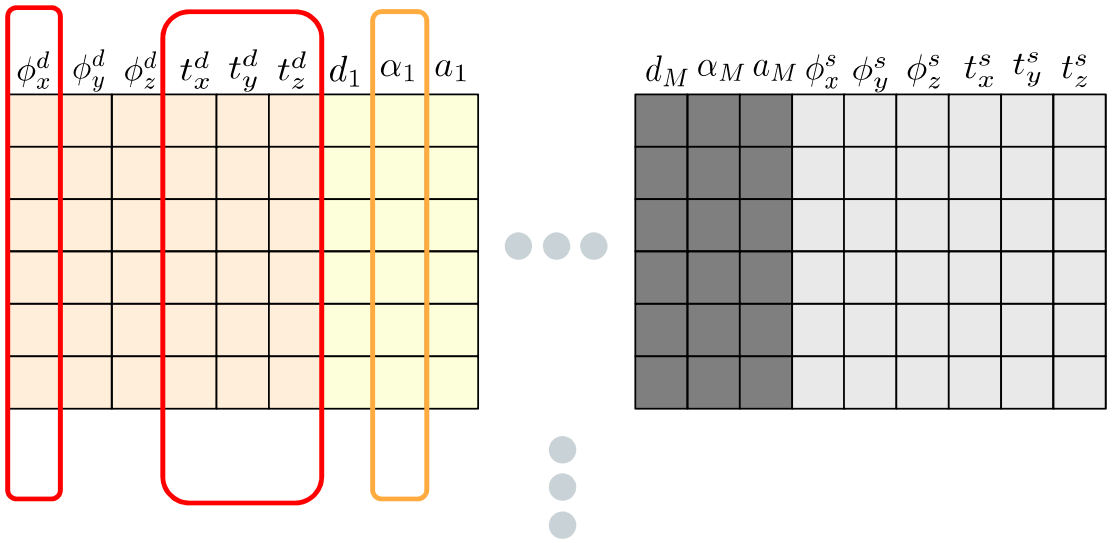


Figure 3.7: A visualization of the system Jacobian for the DCC calibration problem and the degeneracy caused by the inclusion of the α_1 parameter. The Jacobian column of the α_1 parameter can be expressed as a linear combination of a subset of the dynamic-camera-to-end-effector transformation parameters.

the transformation chain Equation (3.1) with respect to the a_1 parameter,

$$\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial a_1} = \frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathtt{s:d}}_{\nu,\bar{\theta}}} \frac{\partial \mathbf{T}^{\mathtt{s:d}}_{\nu,\bar{\theta}}}{\partial \mathbf{T}^{\mathtt{e_1:d}}} \frac{\partial \mathbf{T}^{\mathtt{e_1:d}}}{\partial \mathbf{T}^{\mathtt{e_1:e}}_{\omega_1,\theta_1}} \frac{\partial \mathbf{T}^{\mathtt{e_1:e}}_{\omega_1,\theta_1}}{\partial a_1}. \tag{3.72}$$

Using the derivative functions from Equations (3.44), (3.60), (3.61), and the derivative of the DH matrix with respect to the a_1 parameter, given by (3.36), we can compute

$$\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial a_1} = \begin{bmatrix} 0 \\ \hline -\mathbf{R}^{\mathtt{s:e_1}} \left[\mathbf{R}^{\mathtt{e_1:e}}_{\alpha_1,\theta_1}\right]_1 \end{bmatrix} \tag{3.73}$$

$$= \begin{bmatrix} 0 \\ \hline -\left[\mathbf{R}^{\mathtt{s:e}}\right]_1 \end{bmatrix}. \tag{3.74}$$

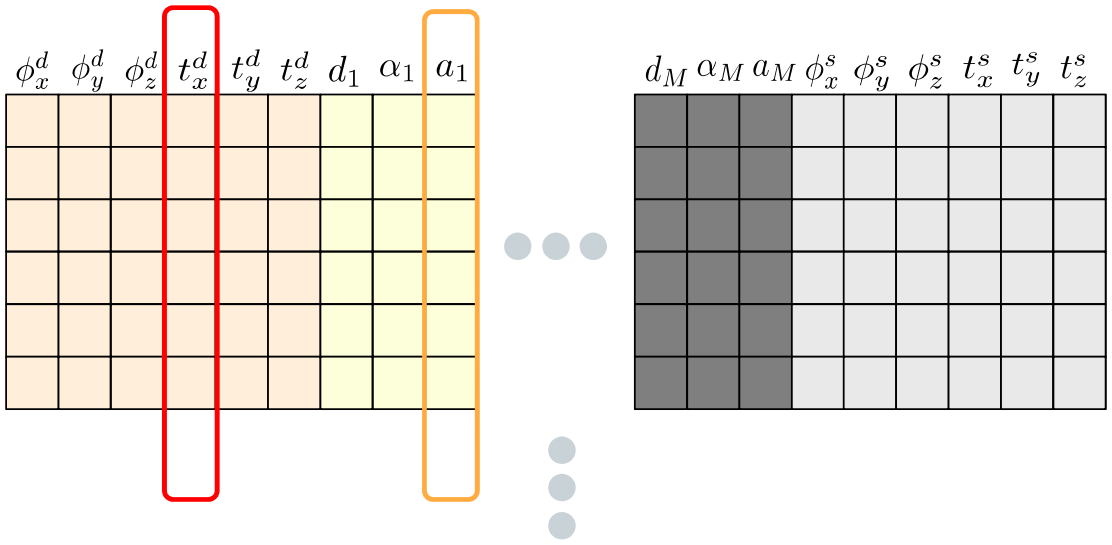


Figure 3.8: A visualization of the system Jacobian for the DCC calibration problem and the degeneracy caused by the inclusion of the a_1 parameter. The Jacobian column of the a_1 parameter is equivalent to the dynamic-camera-to-end-effector transformation $\mathbf{t_x}$ parameter.

Finally, through comparison of (3.74) and (3.58) we see that,

$$\left[\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{e:d}}}\right]_4 = \frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial a_1}. \tag{3.75}$$

Since these columns are identical for all row-blocks added to the system Jacobian through multiple measurement sets, the system experiences a degeneracy if the a_1 parameter is included as part of the calibration state, as the contribution of the a_1 parameter to the system Jacobian, is encompassed by the contribution of the transformation $\mathbf{T}^{\mathsf{e:d}}$ to the system Jacobian. The system Jacobian as it pertains to this degeneracy is visualized in Figure 3.8

Degeneracy 4: Inclusion of the d_1 parameter: Let us compute the derivative of the transformation chain Equation (3.1) with respect to the d_1 parameter,

$$\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial d_1} = \frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{s:d}}_{\nu,\bar{\theta}}} \frac{\partial \mathbf{T}^{\mathsf{s:d}}_{\nu,\bar{\theta}}}{\partial \mathbf{T}^{\mathsf{e_1:d}}} \frac{\partial \mathbf{T}^{\mathsf{e_1:d}}}{\partial \mathbf{T}^{\mathsf{e_1:e}}_{\omega_1,\theta_1}} \frac{\partial \mathbf{T}^{\mathsf{e_1:e}}_{\omega_1,\theta_1}}{\partial d_1}. \tag{3.76}$$

Using the derivative functions from Equations (3.44), (3.60), (3.61), and the derivative of the DH matrix with respect to the d_1 parameter, which is given by Equation (3.32), we can compute

$$\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial d_1} = \begin{bmatrix} 0 \\ \hline -\left[\mathbf{R}^{\mathsf{s:e_1}}\right]_3 \end{bmatrix}, \tag{3.77}$$

We shall now expand the last three columns of the derivative from Equation (3.58) as

$$\left[\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{e:d}}}\right]_{3:6} = \begin{bmatrix} 0 \\ \hline -\mathbf{R}^{\mathsf{s:e}} \end{bmatrix} \tag{3.78}$$

$$= \begin{bmatrix} 0 \\ \hline -\mathbf{R}^{\mathsf{s:e_1}}\mathbf{R}^{\mathsf{e_1:e}} \end{bmatrix}. \tag{3.79}$$

In order to express Equation (3.77) as a linear combination of Equation (3.79), we multiply Equation (3.79) by the column vector $[\mathbf{R}^{\mathsf{e_1:e}}]_3^T$,

$$\left[\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{e:d}}}\right]_{3:6} [\mathbf{R}^{\mathsf{e_1:e}}]_3^T = \begin{bmatrix} 0 \\ \hline -\mathbf{R}^{\mathsf{s:e_1}}\mathbf{R}^{\mathsf{e_1:e}}\,[\mathbf{R}^{\mathsf{e_1:e}}]_3^T \end{bmatrix} \tag{3.80}$$

$$= \begin{bmatrix} 0 \\ \hline -\mathbf{R}^{\mathsf{s:e_1}} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} \tag{3.81}$$

$$= \begin{bmatrix} 0 \\ \hline -\left[\mathbf{R}^{\mathsf{s:e_1}}\right]_3 \end{bmatrix}. \tag{3.82}$$

Recall that the rotation matrix $\mathbf{R}^{\mathsf{e_1:e}}$ is the rotation component of the DH transforma-

tion matrix, $\mathbf{T}^{\mathsf{e_1:e}}_{\omega_1,\theta_1}$, thus, the rotation matrix $\mathbf{R}^{\mathsf{e_1:e}}$ has the form

$$\mathbf{R}^{\mathsf{e_1:e}} = \begin{bmatrix} c(\theta_1) & -s(\theta_1)c(\alpha_1) & s(\theta_1)s(\alpha_i) \\ s(\theta_1) & c(\theta_1)c(\alpha_1) & -c(\theta_1)s(\alpha_i) \\ 0 & s(\alpha_1) & c(\alpha_1) \end{bmatrix}. \tag{3.83}$$

Then, the column vector $[\mathbf{R}^{\mathsf{e_1:e}}]^T_3$ is the third column of the transposed matrix from (3.83),

$$[\mathbf{R}^{\mathsf{e_1:e}}]^T_3 = \begin{bmatrix} 0 \\ s(\alpha_1) \\ c(\alpha_1) \end{bmatrix}. \tag{3.84}$$

The value of α_1 is constant with respect to the measurement configuration. Thus, for any measurement set, we can express the derivative of the DH chain with respect to the d_1 parameter as the linear combination

$$\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial d_1} = \left[\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{e:d}}}\right]_{3:6} \begin{bmatrix} 0 \\ s(\alpha_1) \\ c(\alpha_1), \end{bmatrix} \tag{3.85}$$

and therefore including the d_1 parameter as part of the calibration process results in a system degeneracy. The system Jacobian as it pertains to this degeneracy is visualized in Figure 3.9. Analogous to the degeneracy caused by the d_M parameter, we see that d_1 parameter can be expressed using the translation components of the subsequent 6-DOF transform.

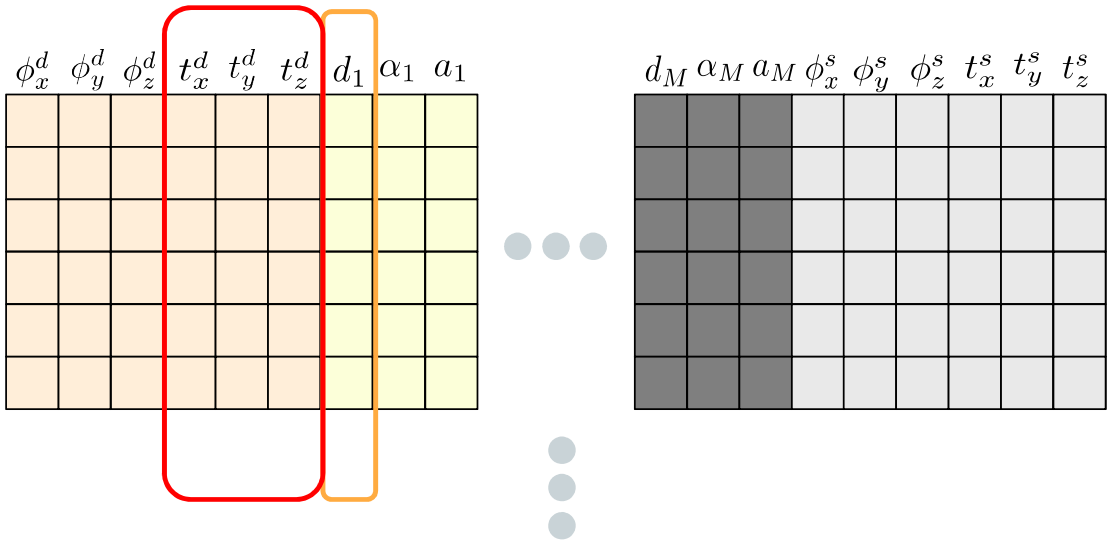


Figure 3.9: A visualization of the system Jacobian for the DCC calibration problem and the degeneracy caused by the inclusion of the d_1 parameter. The Jacobian column of the d_1 parameter can be expressed as a linear combination of the dynamic-camera-to-end-effector translation parameters.

3.4.5 Degeneracies caused by Parallel Joint Axis

Recall from the DH convention discussed in Section 2, that the d_i parameter is the i^{th} link offset, and that the direction of d_i is along the i^{th} link joint axis. We will now show that if the direction of two subsequent joint axes, the directions of d_i and d_{i+1} are parallel, then the system is degenerate. Note that if the two subsequent joint axes are parallel, then the twist angle parameter between them, α_i is equal to zero. An example manipulator with parallel sequential joint axes is depicted in Figure 3.10.

We begin by computing the derivative of the DH chain function from (3.1) with respect to an arbitrary d_i parameter,

$$\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial d_i} = \frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathtt{s:d}}_{\nu,\bar{\theta}}} \frac{\partial \mathbf{T}^{\mathtt{s:d}}_{\nu,\bar{\theta}}}{\partial \mathbf{T}^{\mathtt{e_i:d}}} \frac{\partial \mathbf{T}^{\mathtt{e_i:d}}}{\partial \mathbf{T}^{\mathtt{e_i:e_{i-1}}}_{\omega_1,\theta_1}} \frac{\partial \mathbf{T}^{\mathtt{e_i:e_{i-1}}}_{\omega_i,\theta_i}}{\partial d_i}. \tag{3.86}$$

Using the derivative functions from Equations (3.44), (3.60), (3.19), and the derivative of

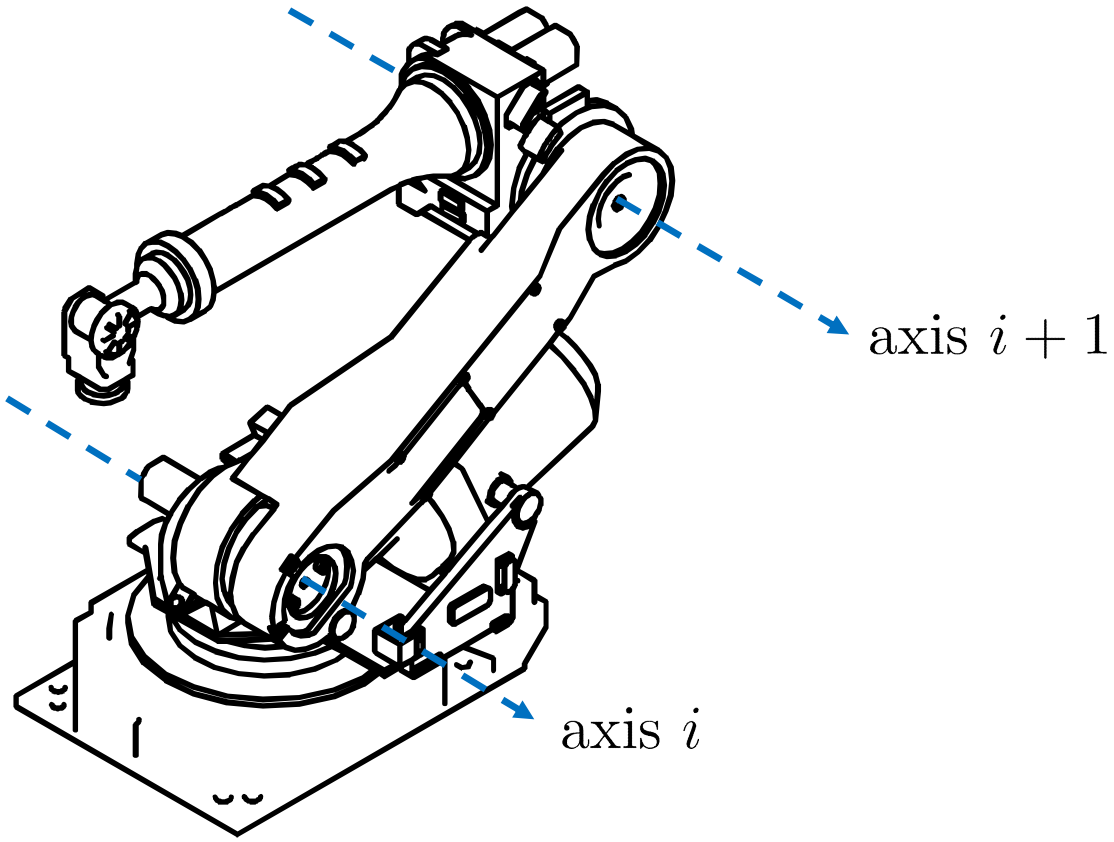


Figure 3.10: An example mechanism where subsequent joint axes are parallel

the DH matrix with respect to the d_i parameter, (3.32), we can compute

$$\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial d_i} = \begin{bmatrix} 0 \\ \hline -\left[\mathbf{R}^{\mathtt{s:e_i}}\right]_3 \end{bmatrix}, \tag{3.87}$$

Using a similar process, we can compute the derivative of the DH chain function from Equation (3.1) with respect to the subsequent d_{i+1} parameter as,

$$\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial d_{i+1}} = \begin{bmatrix} 0 \\ \hline -\left[\mathbf{R}^{\mathtt{s:e_{i+1}}}\right]_3 \end{bmatrix}. \tag{3.88}$$

Expanding the rotation chain from Equation (3.88), we get

$$\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial d_{i+1}} = \begin{bmatrix} 0 \\ \hline -\left[\mathbf{R}^{\mathtt{s:e_{i+1}}}\right]_3 \end{bmatrix} \tag{3.89}$$

$$= \begin{bmatrix} 0 \\ \hline -\mathbf{R}^{\mathtt{s:e_i}} \left[\mathbf{R}^{\mathtt{e_i:e_{i+1}}}\right]_3 \end{bmatrix}. \tag{3.90}$$

If we set $\left[\mathbf{R}^{\mathtt{e_i:e_{i+1}}}\right]_3$ to $[0\,0\,1]^T$, we get that

$$\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial d_{i+1}} = \begin{bmatrix} 0 \\ \hline -\mathbf{R}^{\mathtt{s:e_i}} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} \tag{3.91}$$

$$= \begin{bmatrix} 0 \\ \hline -\left[\mathbf{R}^{\mathtt{s:e_i}}\right]_3 \end{bmatrix} \tag{3.92}$$

$$= \frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial d_i} \tag{3.93}$$

Thus, when $[\mathbf{R}^{\mathbf{e_i}:\mathbf{e_{i+1}}}]_3 = [0\,0\,1]^T$, the derivatives of the error function with respect to d_i, given by Equation (3.87), and with respect to d_{i+1}, given by Equation (3.88), are equal. Since $\mathbf{R}^{\mathbf{e_i}:\mathbf{e_{i+1}}}$ is the rotation component for a DH matrix, its transpose, $\mathbf{R}^{\mathbf{e_{i+1}}:\mathbf{e_i}}$ has the form shown in Equation (3.83). Taking the third column of $\mathbf{R}^{\mathbf{e_i}:\mathbf{e_{i+1}}}$ is equivalent to taking the third row of $\mathbf{R}^{\mathbf{e_{i+1}}:\mathbf{e_i}}$, and thus $[\mathbf{R}^{\mathbf{e_i}:\mathbf{e_{i+1}}}]_3$ is given as

$$[\mathbf{R}^{\mathbf{e_i}:\mathbf{e_{i+1}}}]_3 = \begin{bmatrix} 0 \\ s(\alpha_i) \\ c(\alpha_i) \end{bmatrix} \tag{3.94}$$

Finally, when the sequential joint axes for d_i and d_{i+1} are parallel, the twist angle between the axis are zero, $\alpha = 0$, and (3.94) becomes

$$[\mathbf{R}^{\mathbf{e_i}:\mathbf{e_{i+1}}}]_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{3.95}$$

Thus, when the twist angle, α_i is zero, the two subsequent joint axes are parallel, and

$$\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial d_i} = \frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial d_{i+1}}, \tag{3.96}$$

resulting in a system degeneracy. Similar to the previously discussed degeneracies, the presented result does not depend on the measurement configuration, and therefore, the degeneracy applies to all block-rows added to the system Jacobian from the collected measurement sets. The system Jacobian as it pertains to this degeneracy is visualized in Figure 3.11.

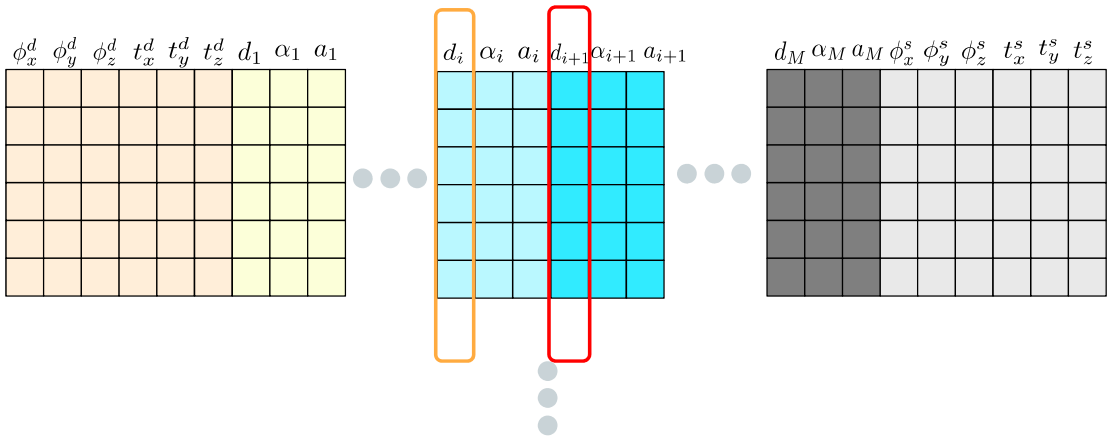


Figure 3.11: A visualization of the system Jacobian for the DCC calibration problem and the degeneracy caused by parallel subsequent joint axes. In this case, the Jacobian column of the d_i and d_{i+1} parameters are equivalent.

3.4.6 Degeneracies Specific to the One Joint Mechanism

In this section, we will discuss the system degeneracies of the DCC calibration problem when there is only one joint axis. In this case, the general chain from Equation (3.42) can be expressed as

$$\mathbf{T}_{\nu,\bar{\theta}}^{\mathsf{s:d}} = \mathbf{T}^{\mathsf{s:b}} \mathbf{T}_{\omega_1,\theta_1}^{\mathsf{e_1:e}} \mathbf{T}^{\mathsf{e:d}} \tag{3.97}$$

where, for this special case, $\bar{\theta} = [\theta_1]$, as there is only one degree of freedom in the system, so the set of estimation parameters becomes

$$\nu = \{\mathbf{T}^{\mathsf{s:b}}, a_1, d_1, \alpha_1, \mathbf{T}^{\mathsf{e:d}}\}. \tag{3.98}$$

However, as discussed in Section 3.4, the columns for the a_1, d_1, and α_1 parameters must be removed from the system Jacobian in order to prevent degeneracy. Thus, for this special case, the set of estimation parameters becomes $\nu = \{\mathbf{T}^{\mathsf{s:b}}, \mathbf{T}^{\mathsf{e:d}}\}$, and we need only analyze the Jacobian of the error function from Equation (3.43) with respect to $\mathbf{T}^{\mathsf{s:b}}$ and $\mathbf{T}^{\mathsf{e:d}}$.

First One Joint Mechanism Degeneracy: First, let us isolate the 5th and 6th

column of the derivative of $\mathbf{T}^{\mathsf{e:d}}$ from Equation (3.58),

$$\left[\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{e:d}}}\right]_5 = \left[\begin{array}{c} 0 \\ \hline -\left[\mathbf{R}^{\mathsf{s:e}}\right]_2 \end{array}\right] \tag{3.99}$$

$$= \left[\begin{array}{c} 0 \\ \hline -\mathbf{R}^{\mathsf{s:b}}\left[\mathbf{R}^{\mathsf{b:e}}\right]_2 \end{array}\right] \tag{3.100}$$

$$\left[\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{e:d}}}\right]_6 = \left[\begin{array}{c} 0 \\ \hline -\left[\mathbf{R}^{\mathsf{s:e}}\right]_3 \end{array}\right] \tag{3.101}$$

$$= \left[\begin{array}{c} 0 \\ \hline -\mathbf{R}^{\mathsf{s:b}}\left[\mathbf{R}^{\mathsf{b:e}}\right]_3 \end{array}\right]. \tag{3.102}$$

Next, let us isolate columns 3 to 6 of the derivative of $\mathbf{T}^{\mathsf{s:b}}$ from Equation (3.48),

$$\left[\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{s:b}}}\right]_{3:6} = \left[\begin{array}{c} 0 \\ \hline -I \end{array}\right] \tag{3.103}$$

We can now develop an expression using the bottom partitions of Equations (3.100), (3.102), and (3.103). Suppose the expression

$$\mathbf{R}^{\mathsf{s:b}}\left[\mathbf{R}^{\mathsf{b:e}}\right]_2 \lambda_1 + \mathbf{I}\lambda_2 = \mathbf{R}^{\mathsf{s:b}}\left[\mathbf{R}^{\mathsf{b:e}}\right]_3 \tag{3.104}$$

holds, where λ_1 and λ_2 are vectors of the appropriate dimensions. Equation (3.104) can be further manipulated as

$$\left[\mathbf{R}^{\mathsf{b:e}}\right]_2 \lambda_1 + (\mathbf{R}^{\mathsf{s:b}})^T \lambda_2 = \left[\mathbf{R}^{\mathsf{b:e}}\right]_3. \tag{3.105}$$

Setting $\lambda_2 = [\mathbf{R}^{\mathsf{s:b}}]_3$, Equation (3.105) becomes

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \left[\mathbf{R}^{\mathsf{b:e}}\right]_3 - \left[\mathbf{R}^{\mathsf{b:e}}\right]_2 \lambda_1 \tag{3.106}$$

Since $\mathbf{R}^{\mathsf{b:e}}$ is the rotation component for a DH matrix, it has the form shown in Equation (3.83), and thus Equation (3.106) can be expressed as

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} s(\theta_1)s(\alpha_1) \\ -c(\theta_1)s(\alpha_1) \\ c(\alpha_1) \end{bmatrix} - \begin{bmatrix} -s(\theta_1)c(\alpha_1) \\ c(\theta_1)c(\alpha_1) \\ s(\alpha_1) \end{bmatrix} \lambda_1. \tag{3.107}$$

Finally it can be shown that setting $\lambda_1 = -\tan(\alpha_1)$ results in the left hand side and right hand side equality for Equation (3.107). Using these results, we see that a subset of the columns of the system Jacobian for the one joint angle case can be expressed as the linear combination,

$$\left[\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{e:d}}}\right]_6 = \left[\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{e:d}}}\right]_5 \lambda_1 + \left[\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{s:b}}}\right]_{3:6} \lambda_2 \tag{3.108}$$

thus resulting in a system degeneracy. Note that both λ_1 and λ_2 are not dependent on the mechanism configuration and therefore, the degeneracy applies to all block-rows added to the system Jacobian from the collected measurement sets. The system Jacobian as it pertains to this degeneracy is visualized in Figure 3.12.

Second One Joint Mechanism Degeneracy:

First, let us investigate the following block columns of the derivative of $\mathbf{T}^{\mathsf{e:d}}$, from Equation (3.58),

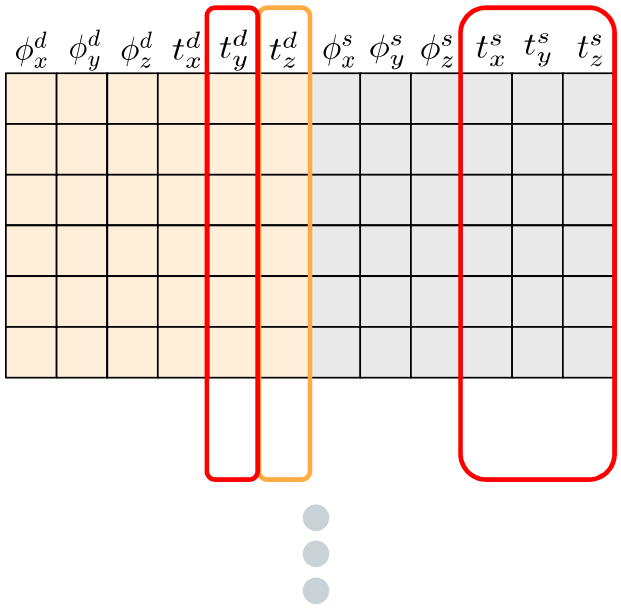


Figure 3.12: A visualization of the system Jacobian for the one joint axis special case of the DCC calibration problem. The Jacobian column corresponding to the z-axis direction of the dynamic-camera-to-end-effector transform can be expressed as a linear combination of other columns, thus causing a degeneracy in the system.

$$\left[\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathtt{e:d}}}\right]_{1:3} = \left[\begin{array}{c} \mathcal{J}_{\boxminus}^{\mathtt{s:d}} \mathbf{R}^{\mathtt{s:e}} \\ \hline 0 \end{array}\right], \tag{3.109}$$

and the following block columns of the derivative of $\mathbf{T}^{\mathtt{s:b}}$ from Equation (3.48),

$$\left[\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathtt{s:b}}}\right]_{1:3} = \left[\begin{array}{c} \mathcal{J}_{\boxminus}^{\mathtt{s:d}} \\ \hline -[\mathbf{R}^{\mathtt{s:b}} \mathbf{t}^{\mathtt{b:d}}]^{\wedge} \end{array}\right]. \tag{3.110}$$

Suppose we set $\lambda_1 = [\mathbf{R}^{\mathtt{b:e}}]_3^T$ and $\lambda_2 = [\mathbf{R}^{\mathtt{s:b}}]_3$. We shall create an expression, $\mathcal{Q}$, relating

the Jacobian columns from Equations (3.109) and (3.110), which evaluates to

$$\mathcal{Q} = \left[\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{e:d}}}\right]_{1:3} \lambda_1 - \left[\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{s:b}}}\right]_{1:3} \lambda_2 \tag{3.111}$$

$$= \left[\frac{\mathcal{J}_{\boxminus}^{\mathsf{s:d}} \mathbf{R}^{\mathsf{s:e}}}{0}\right] \left[\mathbf{R}^{\mathsf{b:e}}\right]_3^T - \left[\frac{\mathcal{J}_{\boxminus}^{\mathsf{s:d}}}{-[\mathbf{R}^{\mathsf{s:b}}\mathbf{t}^{\mathsf{b:d}}]^\wedge}\right] \left[\mathbf{R}^{\mathsf{s:b}}\right]_3 . \tag{3.112}$$

Expanding and simplifying numerator terms, we get

$$\mathcal{Q} = \left[\frac{\mathcal{J}_{\boxminus}^{\mathsf{s:d}} \mathbf{R}^{\mathsf{s:b}} \mathbf{R}^{\mathsf{b:e}} \left[\mathbf{R}^{\mathsf{b:e}}\right]_3^T}{0}\right] - \left[\frac{\mathcal{J}_{\boxminus}^{\mathsf{s:d}} \left[\mathbf{R}^{\mathsf{s:b}}\right]_3}{-[\mathbf{R}^{\mathsf{s:b}}\mathbf{t}^{\mathsf{b:d}}]^\wedge \left[\mathbf{R}^{\mathsf{s:b}}\right]_3}\right] \tag{3.113}$$

$$= \left[\frac{\mathcal{J}_{\boxminus}^{\mathsf{s:d}} \left[\mathbf{R}^{\mathsf{s:b}}\right]_3}{0}\right] - \left[\frac{\mathcal{J}_{\boxminus}^{\mathsf{s:d}} \left[\mathbf{R}^{\mathsf{s:b}}\right]_3}{-[\mathbf{R}^{\mathsf{s:b}}\mathbf{t}^{\mathsf{b:d}}]^\wedge \left[\mathbf{R}^{\mathsf{s:b}}\right]_3}\right] \tag{3.114}$$

$$= \left[\frac{0}{[\mathbf{R}^{\mathsf{s:b}}\mathbf{t}^{\mathsf{b:d}}]^\wedge \left[\mathbf{R}^{\mathsf{s:b}}\right]_3}\right] . \tag{3.115}$$

Using Identity (A.2), we can simplify the denominator,

$$\mathcal{Q} = \left[\frac{0}{\mathbf{R}^{\mathsf{s:b}}[\mathbf{t}^{\mathsf{b:d}}]^\wedge (\mathbf{R}^{\mathsf{s:b}})^T \left[\mathbf{R}^{\mathsf{s:b}}\right]_3}\right] \tag{3.116}$$

$$= \left[\frac{0}{\mathbf{R}^{\mathsf{s:b}}[\mathbf{t}^{\mathsf{b:d}}]^\wedge \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}\right] \tag{3.117}$$

$$= \left[\frac{0}{\mathbf{R}^{\mathsf{s:b}}[\mathbf{t}^{\mathsf{b:d}}]_3^\wedge}\right] . \tag{3.118}$$

Next, let us look at following block columns of the derivative of $\mathbf{T}^{\mathsf{e:d}}$, from Equation (3.58),

$$\left[\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{e:d}}}\right]_{4:6} = \left[\begin{array}{c} 0 \\ \hline -\mathbf{R}^{\mathsf{s:e}} \end{array}\right], \tag{3.119}$$

$$= \left[\begin{array}{c} 0 \\ \hline -\mathbf{R}^{\mathsf{s:b}}\mathbf{R}^{\mathsf{b:e}} \end{array}\right]. \tag{3.120}$$

We shall now construct an expression which relates the Jacobian columns from (3.120) to $\mathcal{Q}$,

$$\left[\begin{array}{c} 0 \\ \hline -\mathbf{R}^{\mathsf{s:b}}\mathbf{R}^{\mathsf{b:e}} \end{array}\right] \lambda_3 = \mathcal{Q} \tag{3.121}$$

$$= \left[\begin{array}{c} 0 \\ \hline \mathbf{R}^{\mathsf{s:b}}[\mathbf{t}^{\mathsf{b:d}}]_3^\wedge \end{array}\right]. \tag{3.122}$$

Equating the bottom partition of $\mathcal{Q}$ with the bottom partition of the left hand side of Equation (3.122), we get,

$$-\mathbf{R}^{\mathsf{s:b}}\mathbf{R}^{\mathsf{b:e}}\lambda_3 = \mathbf{R}^{\mathsf{s:b}}[\mathbf{t}^{\mathsf{b:d}}]_3^\wedge \tag{3.123}$$

$$\mathbf{R}^{\mathsf{b:e}}\lambda_3 = -[\mathbf{t}^{\mathsf{b:d}}]_3^\wedge \tag{3.124}$$

$$\mathbf{R}^{\mathsf{b:e}}\lambda_3 = -\mathcal{P}\mathbf{t}^{\mathsf{b:d}} \tag{3.125}$$

$$\lambda_3 = -(\mathbf{R}^{\mathsf{b:e}})^T\mathcal{P}\mathbf{t}^{\mathsf{b:d}}, \tag{3.126}$$

where $\mathcal{P}$ is a permutation matrix,

$$\mathcal{P} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.127}$$

which is used to express the column of the skew-symmetric matrix term from Equation (3.124) as a standard vector.

Since $\mathbf{t}^{\text{b:d}}$ is the translation vector corresponding to the transformation $\mathbf{T}^{\text{b:d}} = \mathbf{T}^{\text{b:e}}\mathbf{T}^{\text{e:d}}$, $\mathbf{t}^{\text{b:d}}$ can be expanded as

$$\mathbf{t}^{\text{b:d}} = \mathbf{R}^{\text{b:e}}\mathbf{t}^{\text{e:d}} + \mathbf{t}^{\text{b:e}}. \tag{3.128}$$

Substituting Equation (3.128) into Equation (3.126) results in,

$$\lambda_3 = -(\mathbf{R}^{\text{b:e}})^T\mathcal{P}\mathbf{R}^{\text{b:e}}\mathbf{t}^{\text{e:d}} - (\mathbf{R}^{\text{b:e}})^T\mathcal{P}\mathbf{t}^{\text{b:e}}. \tag{3.129}$$

Observe that for the one joint axis case, $\mathbf{R}^{\text{b:e}}$ and $\mathbf{t}^{\text{b:e}}$ corresponds to the rotation and translation component for a single DH matrix, respectively, thus they have the form shown in Equation (2.32). With this, it can be shown that

$$(\mathbf{R}^{\text{b:e}})^T\mathcal{P}\mathbf{R}^{\text{b:e}} = \begin{bmatrix} 0 & c(\alpha_1) & -s(\alpha_1) \\ -c(\alpha_1) & 0 & 0 \\ s(\alpha_1) & 0 & 0 \end{bmatrix} \tag{3.130}$$

$$= \mathcal{M}, \tag{3.131}$$

and similarly,

$$(\mathbf{R}^{\mathsf{b:e}})^T \mathcal{P} \mathbf{t}^{\mathsf{b:e}} = \begin{bmatrix} 0 \\ -a_1 c(\alpha_1) \\ a_1 s(\alpha_1). \end{bmatrix} \tag{3.132}$$

$$= \mathcal{V} \tag{3.133}$$

Thus, λ_3 can be expressed as

$$\lambda_3 = \mathcal{M} \mathbf{t}^{\mathsf{e:d}} + \mathcal{V} \tag{3.134}$$

Combining Equations (3.118) and (3.122), we see that we are able to express the linear combination of Jacobian columns,

$$\left[\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{e:d}}}\right]_{1:3} \lambda_1 - \left[\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{s:b}}}\right]_{1:3} \lambda_2 = \left[\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{e:d}}}\right]_{4:6} \lambda_3 \tag{3.135}$$

which holds true for our selected values of λ_1, λ_2 and λ_3. This linear combination of Jacobian columns indicates that a degeneracy is present in the system, but can be mitigated by the removal of any of the parameters associated with the Jacobian columns present in (3.135). Once again, note that the expressions for λ_1, λ_2 and λ_3 are constructed using parameters that do not change with the DCC configuration, thus, this degeneracy is present for all block-rows added to the system Jacobian as part of a measurement set. The system Jacobian as it pertains to this degeneracy is visualized in Figure 3.13.

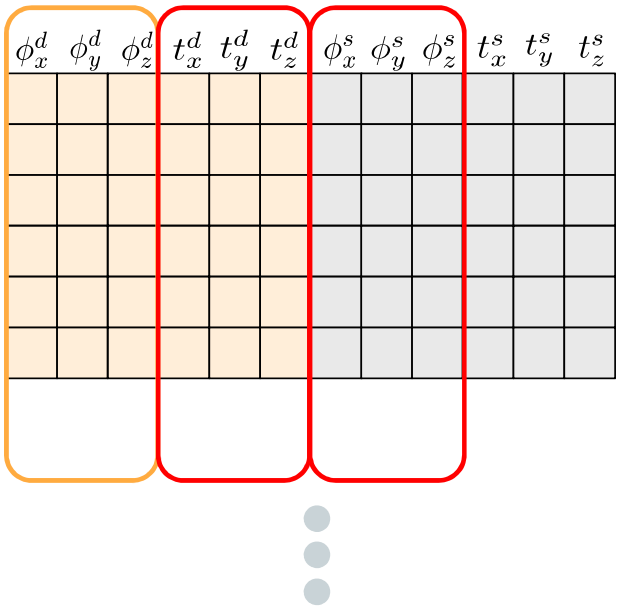


Figure 3.13: A visualization of the system Jacobian for the one joint axis special case of the DCC calibration problem. The Jacobian columns corresponding to dynamic-camera-to-end-effector rotation can be expressed as a linear combination of other columns, thus causing a degeneracy in the system.

3.5 Non-Degeneracy of the Minimal Parameterization

In the previous section, we identified calibration parameters that must be removed in order to avoid degeneracies in the calibration system. In this section, we shall analyse the system Jacobian structure after these identified degeneracies have been removed, and will identify the sufficient conditions required for the system to be *non-degenerate*. The non-degeneracy conditions will be used to illustrate the efficacy of the minimal parameterization identified in Section 3.4.6 for the 1-DOF case, and demonstrate configuration-specific degeneracies that occur for larger degree-of-freedom mechanisms. For the analysis, we will operate on Jacobian matrices constructed using the minimal number of measurements required to solve the system. If the non-degeneracy analysis holds for this minimal system, it must also hold for systems which are constructed using additional measurements, and is therefore a worst case assessment of the conditions for non-degeneracy.

3.5.1 Constructing the Minimal Jacobian

Recall that the box-minus operator from Equation (2.21) computes the tangent space difference between two transformation matrices which are members of the $\mathbb{SE}(3)$ Lie group. Inspecting the poseloop error from Equation (3.10), we see that each collected measurement set provides three measurements corresponding to the rotation component of the poseloop error, and three measurements corresponding to the translation component of the poseloop error, generating six total measurements. Let us denote the top and bottom partitions of the poseloop error from Equation (3.10) as

$$\varepsilon(\nu, \bar{\theta}_i) = \tilde{\mathbf{T}}_i^{\mathtt{s:d}} \boxminus \mathbf{T}_{\nu,\bar{\theta}_i}^{\mathtt{s:d}} \tag{3.136}$$

$$= \begin{bmatrix} \varepsilon_i^r \\ \hline \varepsilon_i^t \end{bmatrix}, \tag{3.137}$$

where ε_i^r and ε_i^t denote the rotation and translation components,respectively, of the poseloop error corresponding to the i^{th} measurement set.

Similar to the process described in [5] for non-degeneracy analysis, the system Jacobian will be constructed using the minimum number of measurements required to solve the system, and we shall refer to this Jacobian as the *minimal Jacobian*. To that end, in order to solve a system with N calibration parameters, a minimum of N independent measurements is required, and therefore the minimal Jacobian will be a square matrix. As discussed in Section 3.4.6, the 1-DOF DCC calibration case requires the estimation of five rotation and five translation parameters, for a total of 10 estimated parameters. Since each measurement set provides six independent measurements, at minimum, we would require the collection of two measurement sets. Note that these two measurement sets would collect six translation and rotation measurements, therefore one of each type of measurement can be discarded in order to generate the minimal Jacobian.

The 2-DOF case requires the estimation of seven rotation and seven translation parameters, and thus 3 measurement sets must be collected in order to construct the minimal Jacobian. Beyond the 2-DOF case, for every additional degree of freedom, one rotation parameter, α_j, and two additional translation parameters, d_j and a_j, are added as calibration parameters. Suppose the DCC mechanism contains D degrees of freedom. Then, for $D > 1$, the number of rotation parameters to be estimated is,

$$N_r = 5 + D, \tag{3.138}$$

and the number of translation parameters is

$$N_t = 5 + 2(D - 1). \tag{3.139}$$

It is evident that the number of estimated translation parameters is greater than or equal to the number of rotation parameters, thus, the number of translation parameters in the system dictates the number of measurement sets required to construct the minimal Jacobian, and can be calculated as

$$K_m(D) = \left\lceil \frac{N_t}{3} \right\rceil, \tag{3.140}$$

where $\lceil \cdot \rceil$ denotes the ceiling operator.

Using the minimum number of measurement sets calculated with Equation (3.140), we shall now construct the minimal Jacobian for a general DCC mechanism, and provide the sufficient conditions for the system to be non-degenerate. The strategy is to construct the square minimal Jacobian, and show that its determinant is a non-zero value through analysis of the matrix rank.

Assuming that the parallel axis case described in Section 3.4.5 is not present, and taking into account the degeneracies described in Section 3.4, the system Jacobian i^{th} row block

has the form,

$$\mathbf{J}_i = \left[\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{e:d}}} \cdots \frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial d_{M-j}} \quad \frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial a_{M-j}} \quad \frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \alpha_{M-j}} \cdots \frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial a_M} \quad \frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \alpha_M} \quad \frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{s:b}}} \right], \tag{3.141}$$

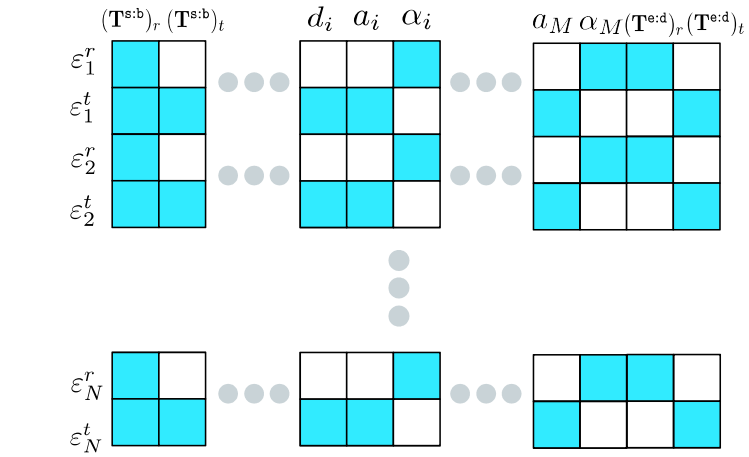
where $\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{e:d}}}$ and $\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{s:b}}}$ are given by Equations (3.58) and (3.48), respectively, and the remaining derivative quantities can be readily calculated using the techniques outlined in Section 3.4. Combining the row blocks from Equation (3.141), the minimal Jacobian for a D degree of freedom mechanism will have the form

$$\mathbf{J}_m^D = \begin{bmatrix} \mathbf{J}_1 \\ \vdots \\ \mathbf{J}_{K_m(D)} \end{bmatrix}. \tag{3.142}$$

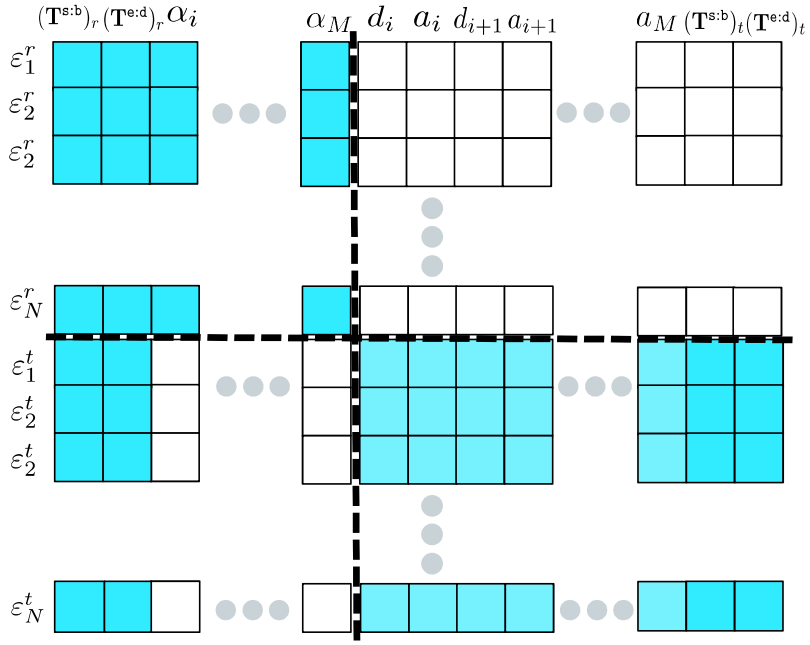
Using the individual Jacobian quantities for the derivatives outlined in Equation (3.141), it is easily demonstrated that the rows and columns of the general minimal Jacobian from Equation (3.142), can be reordered into a block triangular structure. To do so, the columns corresponding to the rotation parameters grouped into the left partition, the columns corresponding to the translation parameters are grouped into the right partition, the rows corresponding to rotation measurements are grouped into the top partition, and finally, the rows corresponding to translation measurements are grouped into the bottom partition. The row and column re-ordering is visualized in Figure 3.14. Note that performing these elementary row and column operations only changes the sign of the determinant, and not the absolute value [5].

The reordered minimal Jacobian now has the form

$$\mathbf{J}_m^D = \left[\begin{array}{c|c} \mathcal{B}^r & 0 \\ \hline \mathcal{B}^{rt} & \mathcal{B}^t \end{array} \right], \tag{3.143}$$



(a) Minimal Jacobian Structure



(b) Re-ordered minimal Jacobian structure

Figure 3.14: Comparison of the minimal Jacobian structure before and after performing row and column re-ordering. (a) depicts the structure of the minimal Jacobian, and (b) shows the minimal Jacobian structure after performing re-ordering. Shaded and white cells depict non-zero and zero entries, respectively. Note the large block of zeros in the top right corner of the re-ordered Jacobian in (b).

and due to the block-diagonal structure of (3.143), the determinant of the minimal Jacobian can be written as

$$|\mathbf{J}_m^D| = |\mathcal{B}^r||\mathcal{B}^t|. \tag{3.144}$$

Thus, the necessary and sufficient conditions for the minimal Jacobian to be degenerate are that, $\mathcal{B}^r$ or $\mathcal{B}^t$, must have a determinant of zero.

3.5.2 1-DOF Non-degeneracy Analysis

We shall now use the sufficient conditions for non-degeneracy from Equation (3.144) to analyse the minimal Jacobian for the 1-DOF case. Recall that for the 1-DOF mechanism that five rotation and five translation parameters are estimated, and that two measurement sets are required to construct the minimal Jacobian.

Translation Block: The augmented translation block from Equation (3.144) has the form

$$\bar{\mathcal{B}}^t = \left[\begin{array}{c|c} I & \left[\mathbf{R}_{\theta^1}^{\mathsf{s:e}}\right]_{1:2} \\ \hline I & \left[\mathbf{R}_{\theta^2}^{\mathsf{s:e}}\right]_{1:2} \end{array}\right]. \tag{3.145}$$

We shall now perform rank-invariant operations in order to simplify $\bar{\mathcal{B}}_t$ and compute its rank. Note that $\bar{\mathcal{B}}^t$ is a 6×5 matrix, but one row will be removed after simplifying, in order to construct the square matrix, $\mathcal{B}^t$. Pre-multiplying $\bar{\mathcal{B}}_t$ by matrix $\mathcal{E}_1$,

$$\mathcal{E}_1 = \left[\begin{array}{c|c} I & 0 \\ \hline I & -I \end{array}\right], \tag{3.146}$$

which subtracts the bottom partition from the top partition, results in

$$\mathcal{E}_1\bar{\mathcal{B}}^t = \left[\begin{array}{c|c} I & \left[\mathbf{R}_{\theta^1}^{\mathsf{s:e}}\right]_{1:2} \\ \hline 0 & \left[\mathbf{R}_{\theta^1}^{\mathsf{s:e}}\right]_{1:2} - \left[\mathbf{R}_{\theta^2}^{\mathsf{s:e}}\right]_{1:2} \end{array}\right]. \tag{3.147}$$

Next, pre-multiplying Equation (3.147) by matrix $\mathcal{E}_2$,

$$\mathcal{E}_2 = \left[\begin{array}{c|c} I & 0 \\ \hline 0 & \mathbf{R}^{\mathsf{b:s}} \end{array}\right], \tag{3.148}$$

yields

$$\mathcal{E}_2\mathcal{E}_1\bar{\mathcal{B}}^t = \left[\begin{array}{c|c} I & \left[\mathbf{R}_{\theta^1}^{\mathsf{s:e}}\right]_{1:2} \\ \hline 0 & \left[\mathbf{R}_{\theta^1}^{\mathsf{b:e}}\right]_{1:2} - \left[\mathbf{R}_{\theta^2}^{\mathsf{b:e}}\right]_{1:2} \end{array}\right]. \tag{3.149}$$

We shall now show that the matrices $\mathcal{E}_1$ and $\mathcal{E}_2$ perform rank-invariant operations to a matrix when pre-multiplied. By inspection, it is evident that the matrix $\mathcal{E}_1$ is a matrix with a full rank of six. Therefore,

$$\text{rank}(\mathcal{E}_1\mathcal{E}_2\bar{\mathcal{B}}^t) = \text{rank}(\mathcal{E}_2\bar{\mathcal{B}}^t). \tag{3.150}$$

In order to determine the rank of matrix $\mathcal{E}_2$, we shall compute its determinant. Since $\mathcal{E}_2$ has a block-diagonal structure, the determinant is computed as

$$|\mathcal{E}_2| = |I||\mathbf{R}^{\mathsf{b:s}}|. \tag{3.151}$$

As $\mathbf{R}^{\mathsf{b:s}}$ is a rotation matrix, its determinant must equal 1. Thus,

$$|\mathcal{E}_2| = 1, \tag{3.152}$$

$\mathcal{E}^2$ has a full rank of 6, therefore,

$$\text{rank}(\mathcal{E}_2\bar{\mathcal{B}}^t) = \text{rank}(\bar{\mathcal{B}}^t), \tag{3.153}$$

and the rank of the original augmented matrix $\bar{\mathcal{B}}^t$ is equal to the rank of the expression

from Equation (3.149).

Inspecting equation (3.149), note that for the 1-DOF case, the rotation, $\mathbf{R}^{\mathsf{b:e}}_{\theta^j}$, is a function of the single θ and α parameters for the one link in the mechanism. However, as a result of degeneracy analysis conducted in Section 3.4.4, the α parameter must not be estimated in order to avoid degeneracy of the system. As is the case with any degenerate parameter identified in Section 3.4, its value can be set to any finite value for use in the construction of the static to dynamic camera transformation chain. In order to simplify the non-degeneracy analysis, and without loss of generality, we can set $\alpha = 0$ for the 1-DOF system, which results in

$$\mathbf{R}^{\mathsf{b:e}}_{\theta^j} = \begin{bmatrix} c(\theta^j) & -s(\theta^j) & 0 \\ s(\theta^j) & c(\theta^j) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.154}$$

Substituting the result from Equation (3.154) into Equation (3.149), results in

$$\mathcal{E}_2\mathcal{E}_1\bar{\mathcal{B}}^t = \left[\begin{array}{c|c} I & \left[\mathbf{R}^{\mathsf{s:e}}_{\theta^1}\right]_{1:2} \\ \hline 0 & \begin{bmatrix} c(\theta^1) & -s(\theta^1) \\ s(\theta^1) & c(\theta^1) \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} c(\theta^2) & -s(\theta^2) \\ s(\theta^2) & c(\theta^2) \\ 0 & 0 \end{bmatrix} \end{array}\right]. \tag{3.155}$$

In order to determine the rank of the expression from Equation (3.155) we shall use the *determinantal* rank theorem, which states that, for a matrix, $\mathcal{A}$, its rank is equal to the order of its largest square sub-matrix with non-zero determinant [90]. To that end, we shall now compute the rank of the matrix from Equation (3.155), using the determinant

of the 5×5 sub-matrix

$$\mathcal{C}^t = \left[\begin{array}{c|c} I & \left[\mathbf{R}^{\mathsf{s:e}}_{\theta^1}\right]_{1:2} \\ \hline 0 & \begin{bmatrix} c(\theta^1) & -s(\theta^1) \\ s(\theta^1) & c(\theta^1) \end{bmatrix} - \begin{bmatrix} c(\theta^2) & -s(\theta^2) \\ s(\theta^2) & c(\theta^2) \end{bmatrix} \end{array}\right] \tag{3.156}$$

$$= \left[\begin{array}{c|c} \mathcal{C}^t_1 & \mathcal{C}^t_2 \\ \hline 0 & \mathcal{C}^t_3 \end{array}\right]. \tag{3.157}$$

Due to the block structure of Equation (3.157), the determinant of $\mathcal{C}^t$ is computed as

$$|\mathcal{C}^t| = |\mathcal{C}^t_1||\mathcal{C}^t_3|. \tag{3.158}$$

By inspection, it is clear that $|\mathcal{C}^t_1| = 1$, and it can be shown that

$$|\mathcal{C}^t_3| = 2 - 2\cos(\theta^1 - \theta^2). \tag{3.159}$$

Therefore, the determinant $\mathcal{C}^t$ will evaluate to zero if $\cos(\theta^1 - \theta^2) = 1$. Otherwise, the determinant of $\mathcal{C}^t$ is non-zero, implying that $\mathcal{C}^t$ has full rank. Since $\mathcal{C}^t$ is the largest square sub-matrix of Equation (3.155) with full rank,

$$\operatorname{rank}(\bar{\mathcal{B}}^t) = \operatorname{rank}(\mathcal{C}^t) \tag{3.160}$$

$$= 5. \tag{3.161}$$

The condition,

$$\cos(\theta^1 - \theta^2) = 1, \tag{3.162}$$

will be true if

$$\theta^1 - \theta^2 = 2\pi n, \quad n \in \mathbb{Z} \tag{3.163}$$

which occurs if two measurement sets are collected from the same DCC configuration.

Finally, recall that $\bar{\mathcal{B}}^t$ is a 6×5 matrix which was constructed in order to facilitate block-wise operations during the simplification process. The original square matrix, from Equation (3.144), $\mathcal{B}^t$, is simply $\bar{\mathcal{B}}^t$ with one row removed, thus $\text{rank}(\mathcal{B}^t) = 5$ so long as the condition from Equation 3.162 is not true.

Rotation Block: The augmented rotation block from Equation (3.144) has the form

$$\bar{\mathcal{B}}^r = \left[\begin{array}{c|c} \mathcal{J}_{\boxminus}^{\mathsf{s:d^1}} & \mathcal{J}_{\boxminus}^{\mathsf{s:d^1}} \left[\mathbf{R}_{\theta^1}^{\mathsf{s:e}}\right]_{1:2} \\ \hline \mathcal{J}_{\boxminus}^{\mathsf{s:d^2}} & \mathcal{J}_{\boxminus}^{\mathsf{s:d^2}} \left[\mathbf{R}_{\theta^2}^{\mathsf{s:e}}\right]_{1:2} \end{array}\right] \tag{3.164}$$

$$= \left[\begin{array}{c|c} \mathcal{J}_{\boxminus}^{\mathsf{s:d^1}} & 0 \\ \hline 0 & \mathcal{J}_{\boxminus}^{\mathsf{s:d^2}} \end{array}\right] \left[\begin{array}{c|c} I & \left[\mathbf{R}_{\theta^1}^{\mathsf{s:e}}\right]_{1:2} \\ \hline I & \left[\mathbf{R}_{\theta^2}^{\mathsf{s:e}}\right]_{1:2} \end{array}\right] \tag{3.165}$$

$$= \mathcal{E}_3 \bar{\mathcal{B}}^t. \tag{3.166}$$

The rank of Equation (3.165) can be computed as

$$\text{rank}(\mathcal{E}_3 \bar{\mathcal{B}}^t) = \text{rank}(\bar{\mathcal{B}}^t) \tag{3.167}$$

so long as $\mathcal{E}_3$ is a full rank matrix. To determine the rank of $\mathcal{E}_3$, we can compute its determinant,

$$|\mathcal{E}_3| = |\mathcal{J}_{\boxminus}^{\mathsf{s:d^1}}||\mathcal{J}_{\boxminus}^{\mathsf{s:d^2}}|, \tag{3.168}$$

where $\mathcal{J}_{\boxminus}^{\mathsf{s:d^1}}$ and $\mathcal{J}_{\boxminus}^{\mathsf{s:d^2}}$ are the Jacobian matrices for the rotation component of the manifold-minus operation, computed from the two measurement sets used to construct the minimal Jacobian. A full derivation of this Jacobian quantity is provided in Appendix B. The key observation is that the Jacobian $\mathcal{J}_{\boxminus}^{\mathsf{s:d^i}}$ is computed using the series of compositions,

$$\mathcal{J}_{\boxminus}^{\mathsf{s:d^i}} = [\Gamma(\log(\tilde{\mathbf{R}}^{\mathsf{s:d^i}} \boxminus \mathbf{R}^{\mathsf{s:d^i}}))]^{-1} \tilde{\mathbf{R}}^{\mathsf{s:d^i}} (\mathbf{R}^{\mathsf{s:d^i}})^T \tag{3.169}$$

where Γ is the Jacobian of the exponential map function for $\mathbb{SO}(3)$, as described in Appendix B, and $\tilde{\mathbf{R}}^{\mathtt{s:d^i}}$ is the rotation component of the measured transformation, $\tilde{\mathbf{T}}^{\mathtt{s:d^i}}$, from Equation (3.8). It is evident that Equation (3.169) is the composition of three full rank matrices, as the rotations $\mathbf{R}^{\mathtt{s:d^i}}$ and $\tilde{\mathbf{R}}^{\mathtt{s:d^i}}$ have a determinant of one, and Γ is an invertible matrix [77]. Thus, the determinant

$$|\mathcal{E}_3| \neq 0, \tag{3.170}$$

and $\mathcal{E}_3$ has full rank.

Note that the right hand side matrix of Equation (3.165) is exactly the minimal Jacobian translation block from Equation (3.145). Therefore, we can apply the same analysis as performed for the translation block and see that $\mathcal{B}^r$ has full rank so long as the condition from Equation (3.162) is not true.

Finally, given the analysis performed on the translation block in Section 3.5.2, and the rotation block in Section 3.5.2, we conclude that, for the 1-DOF case, the system does not suffer from degeneracy so long as the two viewpoints used to construct the minimal Jacobian are distinct.

3.5.3 Beyond the 1-DOF Case

Beyond the 1-DOF case, performing analytical non-degeneracy analysis becomes very involved, as a D degree-of-freedom manipulator requires the zeroing of $K_m(D)$ row blocks in order to manipulate the minimal Jacobian into a block triangular form. The analytical zeroing process quickly degrades to the manipulation of cumbersome expressions which do not offer significant insight into the necessary and sufficient conditions required to avoid degeneracies. To that end, in this section, we shall discuss some specific configurations which have been identified to result in degeneracy, and their relationship to other degeneracies which exist within the literature.

Many of the documented degeneracies for DH parameterization occur when solving the inverse kinematics problem, when the manipulator is in a configuration where it is not possible to determine the joint angle values, or occur as a result of motion degeneracies, which are present in configurations where joint angle velocities do not map uniquely to end effector velocities. On the other hand, degeneracies for the DCC calibration problem identify situations where it is not possible to uniquely estimate a subset of the calibration parameters. Let us now investigate two examples of configuration specific degeneracies that arise in the general DCC calibration problem for a D degree-of-freedom system.

Configuration specific degeneracy 1: Zero angle between subsequent joints Recall that the general form of the Jacobian row block has the form from Equation (3.141). Let us inspect the Jacobian columns corresponding to the a_j and a_{j+1} parameters for two subsequent links,

$$\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial a_j} = \begin{bmatrix} 0 \\ \hline -\left[\mathbf{R}^{\mathsf{s}:\mathsf{e}_{\mathsf{j}-1}}\right]_1 \end{bmatrix} \tag{3.171}$$

$$\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial a_{j+1}} = \begin{bmatrix} 0 \\ \hline -\left[\mathbf{R}^{\mathsf{s}:\mathsf{e}_{\mathsf{j}}}\right]_1 \end{bmatrix} \tag{3.172}$$

The bottom partition of Equation (3.171) can be expanded as

$$\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial a_j} = \begin{bmatrix} 0 \\ \hline -\mathbf{R}^{\mathsf{s}:\mathsf{e}_{\mathsf{j}}} \left[\mathbf{R}^{\mathsf{e}_{\mathsf{j}}:\mathsf{e}_{\mathsf{j}-1}}\right]_1 \end{bmatrix}. \tag{3.173}$$

The term $\left[\mathbf{R}^{\mathsf{e}_{\mathsf{j}}:\mathsf{e}_{\mathsf{j}-1}}\right]_1$ corresponds to the first column of the rotation component from the DH parameter transformation matrix from Equation (2.32), between link frames $\mathcal{F}_{\mathsf{e}_{\mathsf{j}}}$ and

$\mathcal{F}_{\mathsf{e}_{j-1}}$, and has the form

$$[\mathbf{R}^{\mathsf{e}_j:\mathsf{e}_{j-1}}]_1 = \begin{bmatrix} \cos(\theta_j) \\ \sin(\theta_j) \\ 0 \end{bmatrix} \tag{3.174}$$

It is evident that when $\theta_j = 0$ or $\theta_j = \pi$, Equation (3.174) evaluates to

$$[\mathbf{R}^{\mathsf{e}_j:\mathsf{e}_{j-1}}]_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \tag{3.175}$$

which results in

$$\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial a_j} = \left[\begin{array}{c} 0 \\ \hline -[\mathbf{R}^{\mathsf{s}:\mathsf{e}_j}]_1 \end{array}\right]. \tag{3.176}$$

Thus, when the joint angle is selected to be

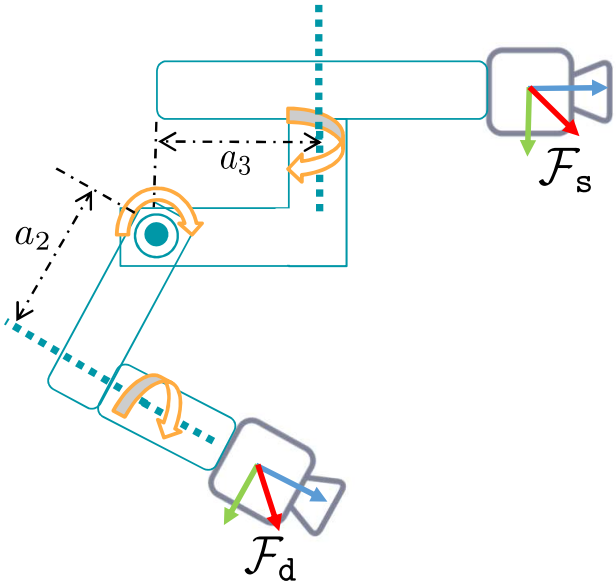
$$\theta_j = \pi n, \quad n \in \mathbb{Z}, \tag{3.177}$$

over the number of measurement sets required to construct the minimal Jacobian, the associated columns,

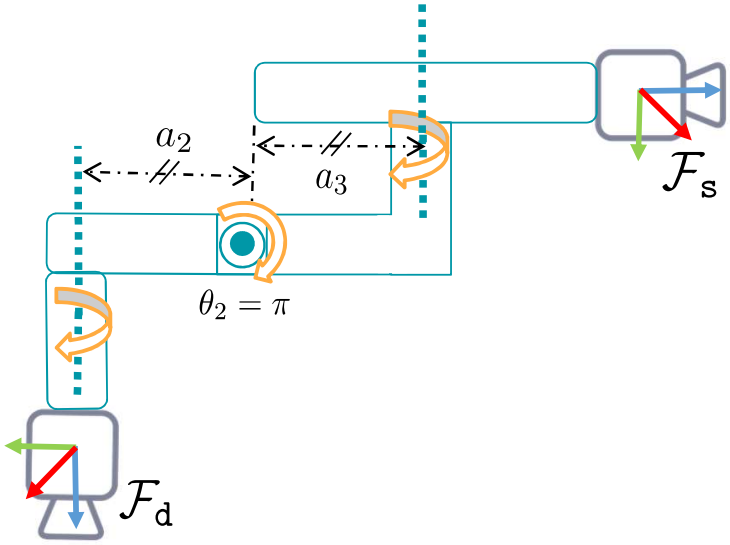
$$\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial a_j} = \frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial a_{j+1}} \tag{3.178}$$

result in a degeneracy. An example of this degeneracy for the 3-DOF gimbal DCC exists when the second joint angle is actuated such that the first rotation axis and the third rotation axis are aligned, as depicted in Figure 3.15. This condition is also known as the *gimbal lock* configuration, which is a well studied degeneracy for Euler angle rotation parametrizations [91].

Configuration specific degeneracy 2: DH chain results in identity rotation We



(a) Non-degenerate Configuration



(b) Degenerate Configuration

Figure 3.15: Non-degenerate and degenerate configurations for the 3-DOF gimbal DCC. (a) depicts the non-degenerate configuration, while (b) illustrates the degeneracy present when $\theta_2 = \pi$, which causes perturbations in a_2 and a_3 to have the same effect on the motion of the dynamic camera frame, $\mathcal{F}_\mathtt{d}$. Note that this degeneracy is also present for $\theta_2 = 0$.

begin by inspecting the translation block from the minimal Jacobian of Equation (3.143), and analysing the columns corresponding to the derivatives with respect to $\mathbf{T}^{\mathsf{e:d}}$ and $\mathbf{T}^{\mathsf{s:b}}$. The i^{th} row-block has the form

$$\mathcal{B}_i^t = \left[\left(\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{e:d}}} \right)_t \cdots \left(\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{s:b}}} \right)_t \right] \tag{3.179}$$

Where $\left(\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{e:d}}} \right)_t$ and $\left(\frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \mathbf{T}^{\mathsf{s:b}}} \right)_t$ correspond to the translation component, or the bottom partition of the Jacobian from Equations (3.58) and (3.48), respectively. Substituting the appropriate partitions from Equations (3.58) and (3.48), $\mathcal{B}^t$ from Equation (3.179) becomes

$$\mathcal{B}_i^t = \left[\mathbf{R}^{\mathsf{s:e}} \cdots I \right]. \tag{3.180}$$

It is evident that if

$$\mathbf{R}^{\mathsf{s:e}} = I, \tag{3.181}$$

for all measurement sets used to construct the minimal Jacobian block, $\mathcal{B}^t$, then the columns corresponding to the derivatives with respect to $\mathbf{T}^{\mathsf{e:d}}$ and $\mathbf{T}^{\mathsf{s:b}}$ become equal, which clearly results in a rank loss of $\mathcal{B}^t$, causing a degeneracy in the system. This configuration specific degeneracy is present for any degree-of-freedom DCC system, so long as the rotational component of the end-effector frame, $\mathcal{F}_{\mathsf{e}}$, is aligned with the static camera frame, $\mathcal{F}_{\mathsf{s}}$.

It should be noted that the degeneracies discussed in this section are *configuration specific*, in that they occur only for specific selections of joint angles used to collect the measurement set. In Chapter 4, we will present an automated approach which avoids configuration specific degeneracies by selecting mechanism joint angles that result in the collection of image measurements with the highest information content.

3.6 Experimental Results

This section presents experimental results which validate the DCC calibration approach presented in Sections 3.1 and 3.2. Note that we only test the static-to-dynamic calibrations, as dynamic-to-dynamic calibration is a straightforward extension of the static-to-dynamic calibration case. First, we shall investigate the calibration performed in a simulation environment, and second, we will discuss the results of performing the DCC calibration using physical hardware.

Similar to existing system ID and calibration methods, sufficiently rich input data is required to ensure the estimated parameters can be accurately determined. Existing marker based MCC calibration relies on relative motion between the marker and camera rig to collect 3D point and image measurement information from multiple viewpoints, whereas a dynamic MCC is able to observe the marker from different viewpoints by actuating the camera. To that end, the data for both the simulation and hardware experiments is collected through a linear spacing strategy, where the measurement configurations are selected by discretizing the joint angle space and systematically collecting measurements from all possible discrete combinations. This is indeed a naive sampling strategy, and the preferred information-theoretic sampling strategy is discussed in Chapter 4.

Finally, we assume that the intrinsic calibration of the lenses is performed prior to the DCC calibration, using any of the widely available lens calibration methods [82, 19].

3.6.1 Simulation Experiments

To validate our calibration approach, we generate a 2-,3-, and 5-DOF mechanism in simulation, and perform the DCC calibration using measurements from a simulated fiducial target. In all three cases, we generate a static camera which is fixed in the world, and a moving camera which is attached to the end effector of each actuated mechanism.

The 5-DOF mechanism simulates a five degree-of-freedom robotic manipulator, similar to the one shown in Figure 3.17(a), with a static camera mounted to the robot base, and the dynamic camera attached to its end effector. The 3-DOF mechanism simulates a 3-axis gimbal, similar to the one shown in Figure 3.17(b), and allows the camera to perform yaw, pitch, and roll motions. Finally, The 2-DOF mechanism is similar to the 3-DOF system, except it only allows for yaw and roll motions of the dynamic camera. Note that the mechanisms are generated with translational offsets between links, which allow for the dynamic camera to undergo translation and rotation when the mechanism inputs are excited. The simulated cameras use a pinhole intrinsic lens model, which includes radial and tangential distortion. Realistic intrinsic model parameters, determined by calibrating the cameras depicted in the hardware set-up from Figure 3.17, are used. The simulated fiducial target is a 6×7 chessboard.

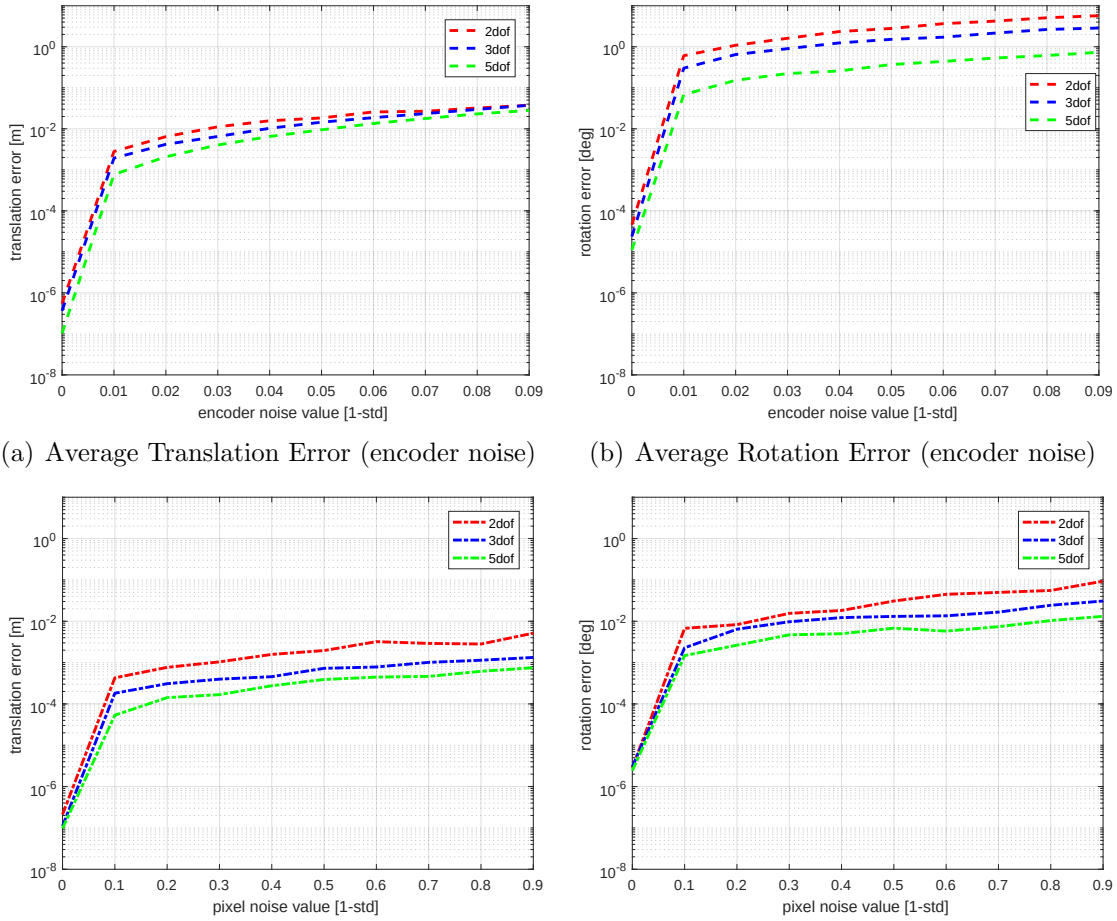
The goal of this simulation study is to understand the effect that noise present in the measured encoder and pixel values has on the overall calibration quality. For each DCC configuration (2-,3-, and 5-DOF), we perform a simulated calibration trials while varying the amount of noise added to the encoder and pixel measurements. Note that we simulate the effect of the encoder and pixel noises independently, therefore, when pixel noise is added to the system, the encoder noise is set to zero, and vice-versa. Gaussian distributed 1-std noise values between zero and 0.09 rad, and zero to 0.9 pixels, are added to the measured encoder and pixel values, respectively. As a point of comparison, a typical machine vision of reasonable quality would exhibit between 0.25 - 0.5 pixel noise, an optical encoder for a high precision manipulator may exhibit noise values less than 0.001 rad, and a magnetic or hall effect encoder can exhibit noise values upwards of 0.05 rad.

For each trial, the initial condition for the calibration parameters is generated by randomly perturbing the ground truth parameters by values sampled from a uniform distribution of zero to 2 cm translation, and zero to 5 degrees rotation. Next, a noise value is

sampled, and is added to the system prior to running the optimization. We then record the optimized parameters and compare that to the ground truth calibration, in order to determine the calibration error. Note that for each noise scenario, 50 noise samples are drawn, resulting in 50 simulated calibration trials, whose errors are then averaged in order to determine the final calibration error for that noise scenario. Finally, for each DCC configuration, the final calibration errors are averaged across translation and rotation parameters, which are presented in Figure 3.16.

From Figure 3.16, first notice that at zero noise, the translation and rotation errors are extremely low (on the order of 10^{-7} meters and 10^{-5} degrees), which verifies that the calibration formulation as described in Sections 3.1 and 3.2, along with the parameter removal strategy outlined in Section 3.4, results in successful calibrations which converge to the ground truth parameters.

Second, Figure 3.16 also illustrates that the encoder noise has a larger effect on the calibration error, as compared to the pixel noise error, for both the translation and rotational components. Such a result suggests that the calibration process is highly sensitive to errors in the encoder values, which geometrically, can be interpreted as angular errors present in the encoders amplifying the position error of each successive joint co-ordinate frame, due to the length and geometry of the link. In contrast, large pixel noise does not have as great of an affect on the calibration error, as the image pixel noise only affects the transformation from the fiducial target to the observing camera. This result suggests that, in order to achieve a high accuracy calibration, when implementing the DCC, it is more important to fit the system with high accuracy encoders, than it is to use an imaging sensor which exhibits low noise characteristics.



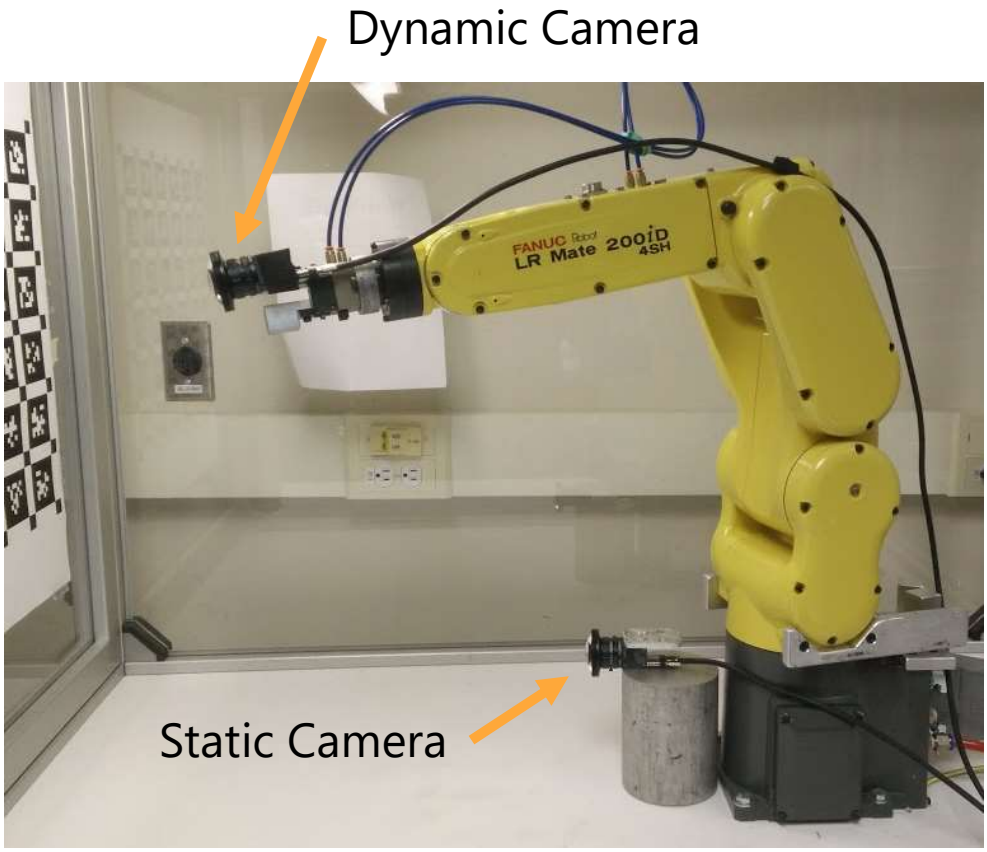
(a) Average Translation Error (encoder noise)

(b) Average Rotation Error (encoder noise)

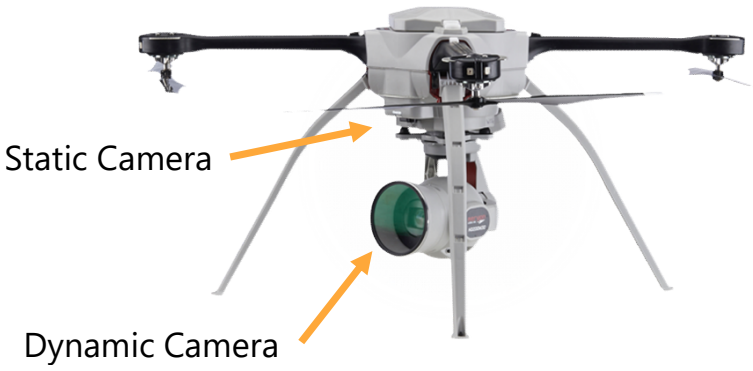
(c) Average Translation Error (pixel noise)

(d) Average Rotation Error (pixel noise)

Figure 3.16: Average rotation and translation errors for the simulated 2-,3-, and 5-DOF mechanisms, as encoder and pixel noise amounts are increased. Plots (a) and (b) show the translation and rotation error when encoder noise is added to the system, while plots (c) and (d) similarly show the errors when pixel noise is added to the system.



(a) Fanuc 5-dof DCC



(b) Aeryon 3-dof DCC

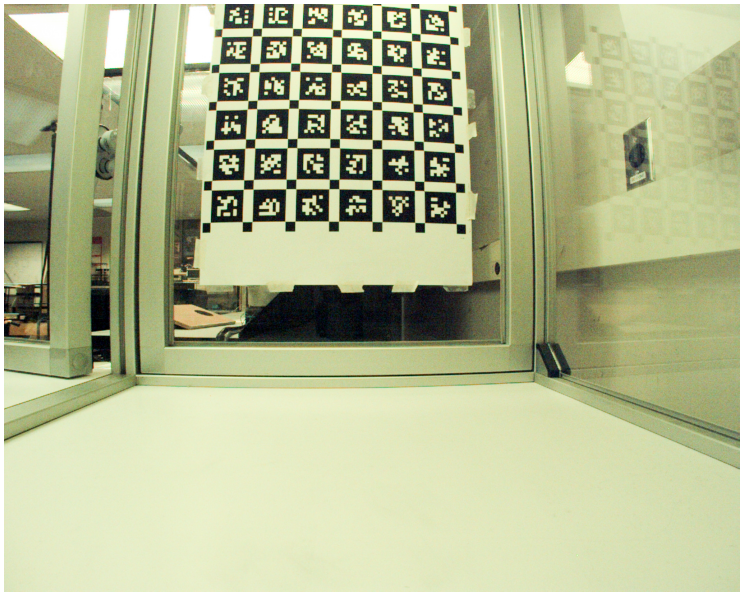
Figure 3.17: Physical hardware set-ups for DCC calibration experiments

3.6.2 Physical Hardware Experiments

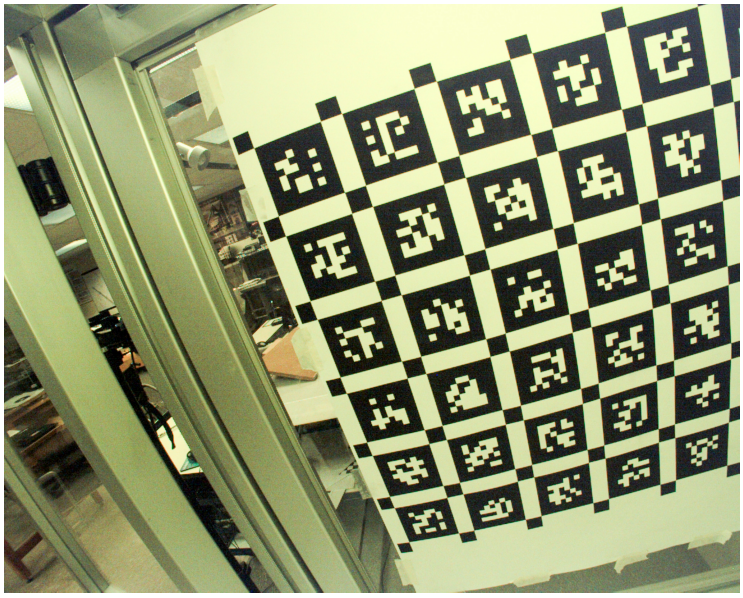
The proposed calibration approaches are further verified using the physical hardware DCCs shown in Figure 3.17. The Fanuc manipulator based DCC, depicted in Figure 3.17(a), is comprised of a 5-DOF Fanuc LR-Mate 200iD manipulator, and two Ximea xIQ cameras which operate at 60fps and 1280×1024 resolution, fitted with 120 degree wide FOV lenses. To build the DCC, one camera is statically mounted to the base, and the other is attached to the manipulator's end effector. The images captured from the static and dynamic cameras of the Fanuc based DCC are presented in Figure 3.18.

The gimbal based DCC, depicted in Figure 3.17(b), consists of a 3-DOF gimballed camera mounted to an Aeryon Skyranger drone, and a downward facing static camera mounted to the frame of the vehicle. The static camera operates at 100 fps and 640×480 resolution, and is fitted with a 150 degree wide FOV lens. The gimbal camera, the Aeryon HD-Zoom 30, captures images at approximately 2 fps and 1920×1080 pixels. The gimbal camera is designed for long-distance surveillance, and thus has zoom functionality and a very narrow FOV. For these experiments, the zoom level of the gimbal camera was set to its widest setting, which resulted in an FOV of approximately 90 degrees. The difference in the captured images for the Aeryon DCC is shown in Figure 3.19. Finally, we use the April grid detector bundled with Kalibr[19] to detect the location of a 6×6 April grid target with respect to the camera, although any fiducial target with known scale is suitable for this application.

For each tested mechanism a *calibration* set is collected using the linear spacing strategy, which is used to perform the optimization and generate the calibration parameters of the DCC. An independent *verification* set is also collected using a random sampling strategy, and is strictly used to verify the results generated using the calibration set. For the Fanuc-DCC, 32 calibration / verification images are collected, and for the Aeryon-DCC, 27 calibration / verification images are collected. Finally, both the reprojection formulation



(a) Static Camera Viewpoint



(b) Dynamic Camera Viewpoint

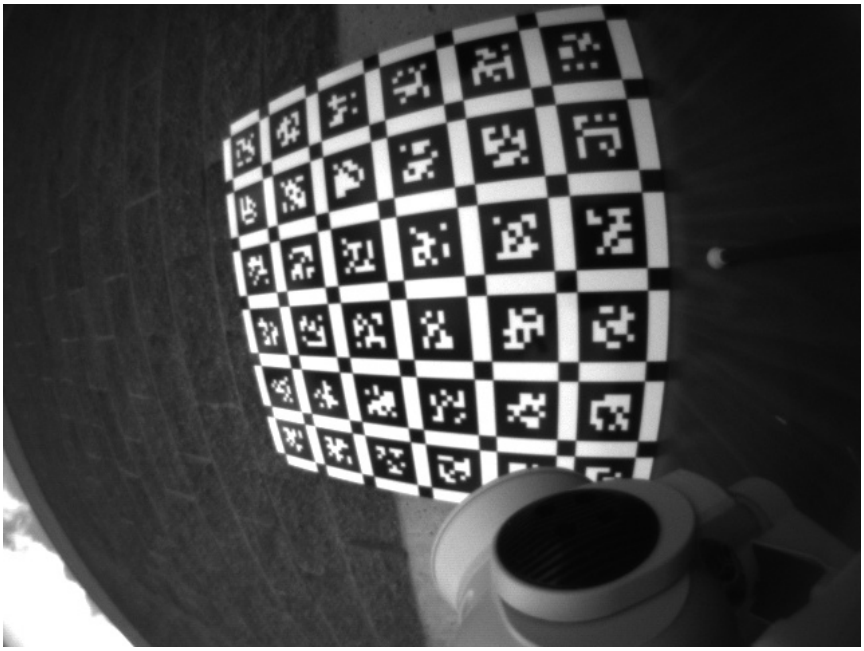
Figure 3.18: Static and dynamic camera images collected using the Fanuc DCC. Note that both images were captured from the same physical location of the robot, and the difference in viewpoint is due to the configuration of the Fanuc manipulator.

from Section 3.1, and the poseloop formulation from Section 3.2, are tested.

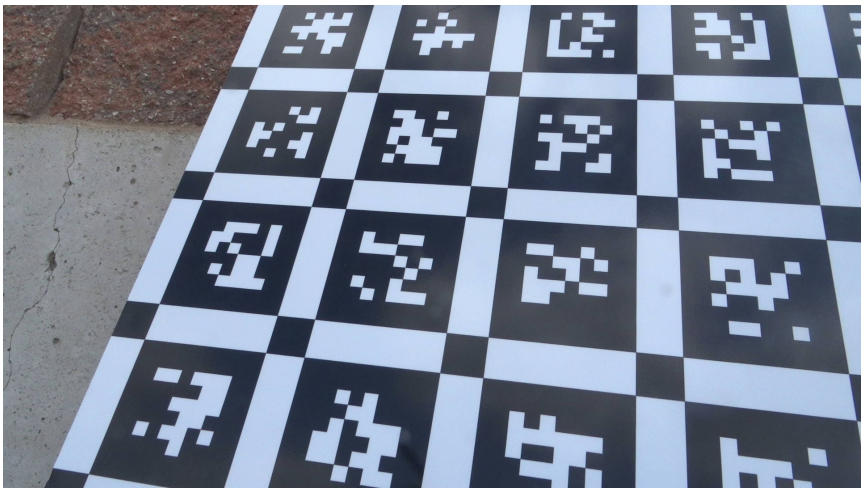
Tables 3.1 and 3.2 present the summary calibration statistics for the Fanuc-DCC and Aeryon-DCC, respectively. For the Fanuc-DCC, both the reprojection and poseloop RMSE error of the calibration set is fairly low, and the verification set RMSE is comparable to that of the calibration set (within 0.03 - 0.05 pixels), which verifies that both of our proposed approaches are able to generate a high quality calibration.

The Aeryon-DCC also demonstrates similar performance between the reprojection and poseloop approaches based on the calibration set RMSE. However, compared to the Fanuc-DCC, we see that the Aeryon-DCC exhibits significantly higher RMSE values. The increased error is likely due to less precise encoders used for joint angle measurements in the Aeryon-DCC, when compared to Fanuc-DCC. Since the Fanuc manipulator is designed for high-precision ($\pm$0.02mm) pick and place applications, the required accuracy of the encoders is quite high, whereas the Aeryon gimbal only requires the encoder measurements to assist with image stabilization, and thus does not require high accuracy. As demonstrated in the simulation results from Section 3.6.1, precise encoder measurements are fundamental to achieving high quality calibrations, which corroborates our experimental results of the Fanuc-DCC out-performing the Aeryon-DCC.

Additional sources of error in the Aeryon-DCC are non-rigidity in the system, and the lower quality optics found in the static camera. Our formulations assume that the chain transformations take place between rigid links, and thus any flex, backlash, or non-rigid components in the system are not modeled and contribute to error. Since the Fanuc manipulator is used for precise and repeatable operations, its material and structural design maximize rigidity. The gimbal on the Aeryon-DCC, on the other hand, is designed to be low-weight in order to maximize drone flight times, and likely sacrifices some system rigidity in order to maintain the low-weight design. Finally, observing Figure 3.19(a), it is evident that the integrated static camera exhibits decreased sharpness and image quality, which



(a) Static Camera Viewpoint



(b) Dynamic Camera Viewpoint

Figure 3.19: Static and dynamic camera images collected using the Aeryon DCC. Note that both images were captured from the same physical location of the vehicle, and the difference in the images is due to the dynamic camera having a much narrower FOV.

decreases the positioning accuracy of the camera relative to the target.

Table 3.1: Summary statistics for Fanuc manipulator DCC calibration

	Calibration Set Error (pixel RMSE)	**Validation Set Error (pixel RMSE)**
Reprojection	0.55	0.58
Poseloop	0.54	0.59

Table 3.2: Summary statistics for Aeryon Gimbal DCC calibration

	Calibration Set Error (pixel RMSE)	**Validation Set Error (pixel RMSE)**
Reprojection	1.98	2.16
Poseloop	2.01	2.13

Chapter 4

Information Theoretic Next-Best-View [1]

While Chapter 3 shows that it is possible to achieve a calibration of the dynamic camera cluster, the quality of the calibration is heavily dependent on the ability to collect an extensive set of measurements while providing sufficient excitation to the joints inputs, as the parameter estimates are highly sensitive to the selected measurement sets. The manual measurement collection process is increasingly onerous as the number of degrees of freedom of the manipulator increase, as it becomes difficult to ensure sufficient measurement excitation through the mechanism's configuration space. Even if an exhaustive measurement set is collected, it does not guarantee accurate calibration after the optimization, as the relationship between manually collected measurements and uncertainty of the estimation parameters is unclear. In this chapter, we present two methods that seeks to find a next-best-view which locally minimizes calibration parameters' covariance with each

[1] Partial contents of this chapter have been incorporated within a co-authored paper that has been accepted for publication. J. Rebello, A. Das and S.L. Waslander, "Autonomous Active Calibration of a Dynamic Camera Cluster using Next-Best-View." 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). All authors contributed equally to the work.

successively collected measurement, until a user specified accuracy is achieved. The first method will look at minimizing the *entropy* of the calibration parameters, while the second method will look to select the viewpoints which maximize the *mutual information* between the joint angle input and calibration parameters. Note that the automatic calibration processes can be performed using the re-projection error or pose-loop error methods outlined in Chapter 3.

4.1 Parameter Initialization

The autonomous calibration process is initialized with a collection of M measurement sets. We shall denote a generic measurement set collection as $D_{1:M}$, which are initially obtained by sampling the configuration space. Common strategies such as *random*, *systematic*, or *cluster* [92] sampling can be used to generate the initial measurement set collection. Each measurement set D_i is obtained from a sampled mechanism input $\bar{\theta}_i$. The prior mean and covariance of the calibration parameters are obtained by optimizing Equation (3.6) or (3.11) using all of the sampled measurements, $L_{1:M}$. The resulting parameter estimate, $\widetilde{\nu}_{1:M}$, produced by the optimization is assumed to be Gaussian distributed with covariance $\Sigma_{\widetilde{\nu},\bar{\theta}_{1:M}}$.

4.2 Covariance Entropy Minimization

In order to compute the covariance of the parameters, $\Sigma_{\tilde{\nu},\bar{\theta}_{1:M}}$, the Jacobian of the residual equation, $\mathbf{J}_{\tilde{\nu},\bar{\theta}_{1:M}}$, is required, where

$$\mathbf{J}_{\tilde{\nu},\bar{\theta}_{1:M}} = \begin{bmatrix} \mathbf{J}_{\tilde{\nu},\theta_1} \\ \vdots \\ \mathbf{J}_{\tilde{\nu},\theta_M} \end{bmatrix}. \tag{4.1}$$

Each row-block of (4.1), $\mathbf{J}_{\tilde{\nu},\bar{\theta}_i}$, corresponds to the Jacobian contribution of the configuration associated with the i^{th} joint input, $\bar{\theta}_i$, and can be calculated as

$$\mathbf{J}_{\tilde{\nu},\bar{\theta}_i} = \frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \nu} \tag{4.2}$$

using the techniques outlined in Chapter 3. Finally, using a first order approximation of the Fisher information matrix [93], the parameter covariance is given as,

$$\Sigma_{\tilde{\nu},\bar{\theta}_{1:M}} = (\mathbf{J}^T_{\tilde{\nu},\bar{\theta}_{1:M}} \Omega \mathbf{J}_{\tilde{\nu},\bar{\theta}_{1:M}})^{-1}. \tag{4.3}$$

where Ω is the measurement noise information matrix, as described in Section 2.5.

4.2.1 Next-Best-View Configuration Selection

To reduce the uncertainty in the calibration parameters with each subsequent measurement set, we seek a locally optimal mechanism configuration, $\bar{\theta}^*$, which will minimize the entropy of the estimation parameters. Suppose we have an arbitrary mechanism configuration for the next-best-view, $\hat{\bar{\theta}}$. Then, the resulting measurement Jacobian matrix, which includes the measurement from $\hat{\bar{\theta}}$, has the form

$$\mathbf{J}_{\widetilde{\nu},\eta}(\hat{\bar{\theta}}) = \begin{bmatrix} \mathbf{J}_{\widetilde{\nu},\bar{\theta}_{1:M}} \\ \mathbf{J}_{\widetilde{\nu},\hat{\bar{\theta}}} \end{bmatrix}, \tag{4.4}$$

where $\eta = \{\bar{\theta}_{1:M}, \hat{\bar{\theta}}\}$ denotes the set of actuator inputs from $\bar{\theta}_{1:M}$ and the optimal next-best-view configuration, $\hat{\bar{\theta}}$. Using (4.4), the parameter covariance for the estimation parameters can be predicted by

$$\Sigma_{\widetilde{\nu},\eta}(\hat{\bar{\theta}}) = ((\mathbf{J}_{\widetilde{\nu},\eta}(\hat{\bar{\theta}})^T \Omega \mathbf{J}_{\widetilde{\nu},\eta}(\hat{\bar{\theta}}))^{-1}. \tag{4.5}$$

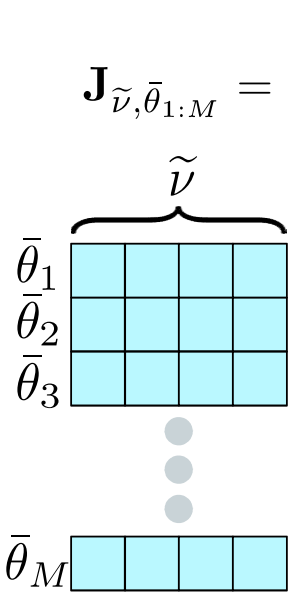
Note that (4.5) is an approximation to the true parameter uncertainty when measurements from $\hat{\bar{\theta}}$ are included, as $\widetilde{\nu}$ is computed using the configurations from $\bar{\theta}_{1:M}$, and does not include $\hat{\bar{\theta}}$. The accuracy of this approximation will degrade according to the error between $\widetilde{\nu}$ and the true estimation parameters. However, in our experiments, we have seen promising results with $\widetilde{\nu}$ being initialized according to the process described in Section 4.1. Further, the approximation improves as the calibration process proceeds and converges to an accurate set of parameters. The construction of the predicted covariance matrix is visualized in Figure 4.1.

Our next-best-view configuration is determined by formulating a cost function using the covariance matrix given in (4.5). Suppose we have an actuated mechanism with L joints. Then, we shall define a cost $\Lambda_e : \mathbb{R}^L \mapsto \mathbb{R}$ which is given as,

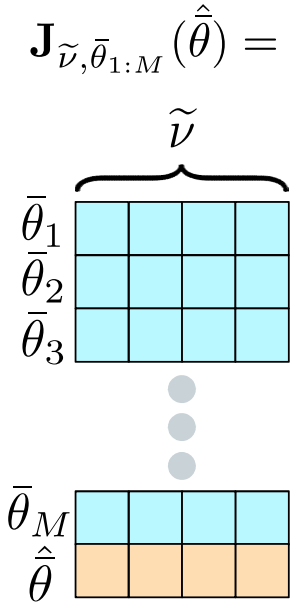
$$\Lambda_e(\hat{\bar{\theta}}) = h_e(\Sigma_{\widetilde{\nu},\eta}(\hat{\bar{\theta}})), \tag{4.6}$$

where $h_e(\cdot)$ denotes the entropy function, as described in Chapter 2. The cost function in (4.6) maps a next-best-view mechanism input, $\hat{\bar{\theta}}$, to the expected entropy of the parameter covariance matrix from (4.5).

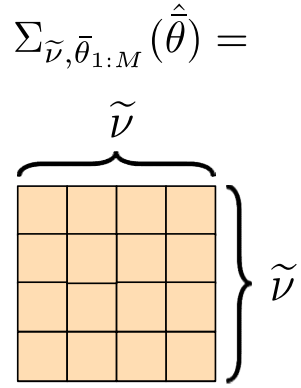
Although the cost defined in (4.6) uses the entropy of the covariance matrix in order to quantify the parameter uncertainty, it is possible to also use other metrics, such as the



(a) Initial Jacobian



(b) Row addition from $\hat{\bar{\theta}}$



(c) Predicted covariance matrix

Figure 4.1: Visualization of the predicted covariance matrix construction. (a) depicts the initial Jacobian matrix containing contributions from measurement set 1 to M. (b) illustrates how the contribution from the candidate next-best-view measurement, $\hat{\bar{\theta}}$, is added to the bottom of the Jacobian. (c) shows the covariance matrix computed using the Jacobian from (b), which predicts the parameter uncertainty if $\hat{\bar{\theta}}$ were to be used in the estimation process.

Frobenius norm, trace of the covariance matrix, and mutual information. Entropy of the covariance matrix was selected in this case, as that metric has been shown to work well in related applications, such as key-frame selection for visual SLAM, which is discussed in more detail in Chapter 5. A mutual information based approach is presented in Section 4.4, and we demonstrate its relationship to the entropy based approach in Section 4.5.

In order to find the optimal next-best-view, $\bar{\theta}^*$, the cost function from (4.6) is optimized over the feasible configurations of the actuated mechanism,

$$\begin{aligned} \min \quad & \Lambda_e(\hat{\bar{\theta}}) \\ \text{subject to} \quad & \lambda^l < \hat{\bar{\theta}} < \lambda^u, \end{aligned} \tag{4.7}$$

where $\bar{\theta}^l$ and $\bar{\theta}^u$ are the upper and lower bounds, respectively, of the mechanism input angles. Note that in practice the bounds of each joint angle are generally available from either the mechanism manufacturer or can be determined using a homing process.

4.3 Entropy Optimization with Successive Next-Best-View Measurements

Once the next-best-view configuration, $\bar{\theta}^*$, is determined, the actuated mechanism is moved and a measurement set, $D_{\bar{\theta}^*}$, is collected from the corresponding optimal configuration. The measurement set is then appended such that

$$D_{1:M+1} \leftarrow D_{1:M} \cup D_{\bar{\theta}^*} \tag{4.8}$$

$$M = M + 1. \tag{4.9}$$

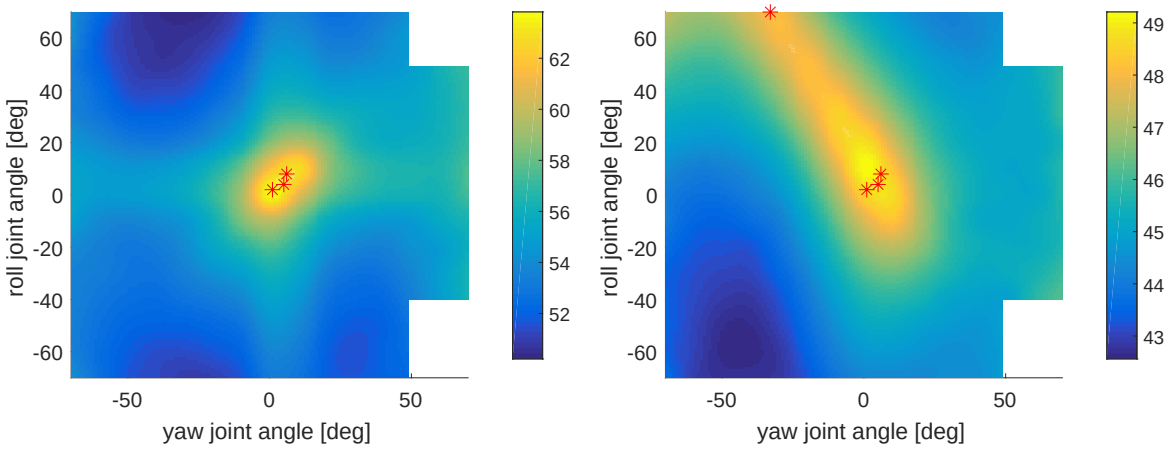
The estimation parameters are optimized using the updated measurement sets, in order

to recompute $\widetilde{\nu}_{1:M}$ using the additional measurements from $D_{\bar{\theta}^*}$. Then, the next-best-view selection procedure described in Section 4.2.1 is performed again. The process of selecting the next-best-view, then re-calculating the estimation parameters, $\widetilde{\nu}_{1:M}$, is repeated until the entropy score from (4.6) reaches a user selected threshold, or a maximum number of views is selected. Figure 4.2 visualizes the next-best-view cost from (4.6) for a two degree of freedom (DOF) mechanism, and also illustrates the NBV optimization and selection process over 4 measurement collections.

Note that our next-best-view approach performs a continuous optimization over the mechanism's configuration space, and also takes into account the viewpoint of the fiducial target implicitly in the formulation through the kinematic and projection Jacobians. Thus, our approach does not require discretization of the configuration space, or predefined motion paths over a finite set of target positions. Instead, we are able to perform next-best-view selection for arbitrary target positions, so long as corresponding feature measurements can be acquired from both the static and dynamic camera.

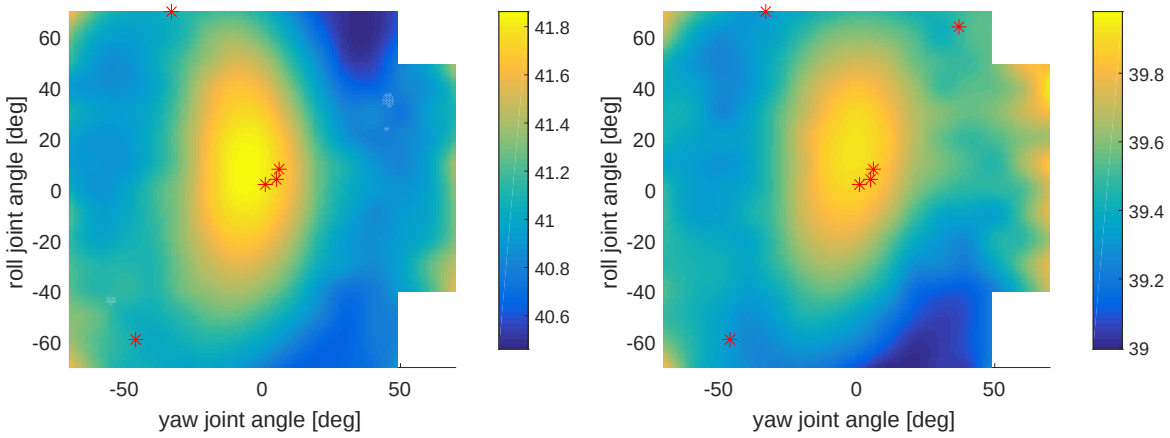
4.4 Mutual Information Maximization

Since the entropy based approach discussed in Section 4.2 operates on the *entire* parameter covariance matrix in order to determine the next-best-view score, it can be thought of as a method which aggregates and quantifies the uncertainty across all of the estimation parameters. In some cases, it is more useful to understand how a subset of parameters is affected by knowledge of a different set of variables. For example, if performing the DCC calibration in a situation where additional feature landmarks must be estimated along side the calibration parameters, it would be beneficial to understand how the mechanism configuration affects both the calibration parameters and feature landmarks independently. To that end, the *mutual information* which exists between variables can be used to quantify



(a) Initial Surface

(b) 1 NBV Addition



(c) 2 NBV Additions

(d) 3 NBV Additions

Figure 4.2: Progression of the entropy based NBV cost surface for a two degree-of-freedom mechanism, as measurements are added from the NBV configurations. In this case, the autonomous calibration is initialized using 3 configurations, then 3 additional measurements are added using the proposed NBV approach. The red asterisks denote configurations from which the collected measurements were used for parameter estimation. (a) shows the cost surface after initialization, while (b)-(d) depict the changing cost surface as NBV measurements are added. Note that the plots are coloured according to the covariance entropy in nats.

how knowledge of one set of parameters affects the uncertainty of another.

In this section, we shall present a more general formulation for next-best-view selection, which maximizes the mutual information between the candidate measurement and the calibration parameters, and can also be easily be extended to perform NBV selection for calibration problems which contain additional estimation variables. We shall also show how this formulation of next-best-view is related to the entropy minimization method described in Section 4.2.

4.4.1 Cost formulation

Let us define the residual error term for the candidate measurement, $\hat{\bar{\theta}}$, as

$$\varepsilon_{\hat{\bar{\theta}}} = \bar{\theta} - \hat{\bar{\theta}}, \tag{4.10}$$

which has the Jacobian

$$\frac{\partial \varepsilon_{\hat{\bar{\theta}}}}{\partial \hat{\bar{\theta}}} = -\mathbf{I}. \tag{4.11}$$

To construct the required Jacobian matrix for mutual information based next-best-view, we first augment the Jacobian from Equation (4.4), $\mathbf{J}_{\tilde{\nu},\eta}(\hat{\bar{\theta}})$, with an additional column, which is the derivative of the poseloop error from Equation (3.10) with respect to the candidate measurement, $\hat{\bar{\theta}}$,

$$\mathbf{J}_{\hat{\bar{\theta}}} = \frac{\partial \varepsilon(\nu, \bar{\theta}_i)}{\partial \hat{\bar{\theta}}}, \tag{4.12}$$

which can be readily computed using the techniques outlined in Chapter 3. Second, we augment the Jacobian from Equation (4.4), with the candidate measurement Jacobian from Equation 4.11. This augmentation is due to the added residual term from Equation (4.10), which reflects the direct measurements of the joint angles that are available as part of the system. The augmented Jacobian, which will be used for mutual information computation,

is given as

$$\mathbf{J}_{\widetilde{\nu},\hat{\tilde{\theta}}} = \left[\begin{array}{c|c} \mathbf{J}_{\widetilde{\nu},\eta}(\hat{\tilde{\theta}}) & \mathbf{J}_{\hat{\tilde{\theta}}} \\ \hline 0 & -I \end{array}\right]. \tag{4.13}$$

Similar to the covariance matrix calculated in Equation (4.5), we shall now compute the augmented covariance matrix using the Jacobian from Equation (4.13),

$$\Sigma_{\widetilde{\nu},\hat{\tilde{\theta}}} = ((\mathbf{J}_{\widetilde{\nu},\hat{\tilde{\theta}}})^T \Omega \mathbf{J}_{\widetilde{\nu},\hat{\tilde{\theta}}})^{-1} \tag{4.14}$$

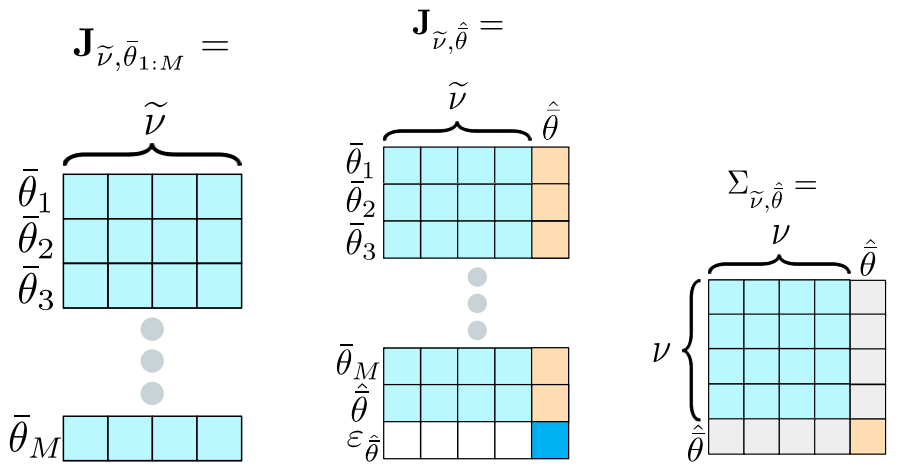
$$= \left[\begin{array}{c|c} \Sigma_{\widetilde{\nu}\widetilde{\nu}} & \Sigma_{\widetilde{\nu}\hat{\tilde{\theta}}} \\ \hline \Sigma_{\hat{\tilde{\theta}}\widetilde{\nu}} & \Sigma_{\hat{\tilde{\theta}}\hat{\tilde{\theta}}} \end{array}\right] \tag{4.15}$$

which is a covariance matrix which has entries corresponding to both the calibration parameters from the current linearization point, $\widetilde{\nu}$, and the candidate joint angle measurement, $\hat{\tilde{\theta}}$. The construction of this augmented covariance matrix is visualized in Figure 4.3. Note that the inclusion of the derivatives from Equation (4.11) in the augmented Jacobian ensure that the approximated Hessian matrix from Equation (4.14) is always invertible, as the residual term from Equation (4.10) indicates that we always have a direct measurement of the additional variables added to the system through the column augmentation of Equation (4.12).

The mutual information based next-best-view is determined by formulating a cost function using the covariance matrix given in (4.14). Using the techniques described in Section 2 for computing the mutual information between two partitions of a Gaussian distribution, we can develop the mutual information expression between the calibration parameters from the current linearization point, $\widetilde{\nu}$, and the candidate joint angle measurement, $\hat{\tilde{\theta}}$, as

$$I(\widetilde{\nu}; \hat{\tilde{\theta}}) = \frac{1}{2} \log \frac{|\Sigma_{\widetilde{\nu}\widetilde{\nu}}||\Sigma_{\hat{\tilde{\theta}}\hat{\tilde{\theta}}}|}{|\Sigma_{\widetilde{\nu},\hat{\tilde{\theta}}}|}. \tag{4.16}$$

Similar to the entropy minimization approach outlined in Sections 4.2.1 and 4.3, we



(a) Initial Jacobian

(b) Row and column augmentation from $\hat{\bar{\theta}}$

(c) Covariance matrix

Figure 4.3: Visualization of the covariance matrix construction for mutual information maximization. (a) depicts the initial Jacobian matrix containing contributions from measurement set 1 to M. (b) illustrates how the contribution from the candidate next-best-view measurement, $\hat{\bar{\theta}}$, is added to the bottom row of the Jacobian, as well as to the right-hand-most column. (c) shows the covariance matrix computed using the Jacobian from (b), which contains elements from the current linearization point, $\widetilde{\nu}$, and the candidate joint angle measurement, $\hat{\bar{\theta}}$.

shall define a mutual information cost $\Lambda_m : \mathbb{R}^L \mapsto \mathbb{R}$, which is given as,

$$\Lambda_m(\hat{\bar{\theta}}) = I(\widetilde{\nu}; \hat{\bar{\theta}}). \tag{4.17}$$

In order to find the optimal next-best-view, $\bar{\theta}^*$, the cost function from (4.17) is optimized over the feasible configurations of the actuated mechanism,

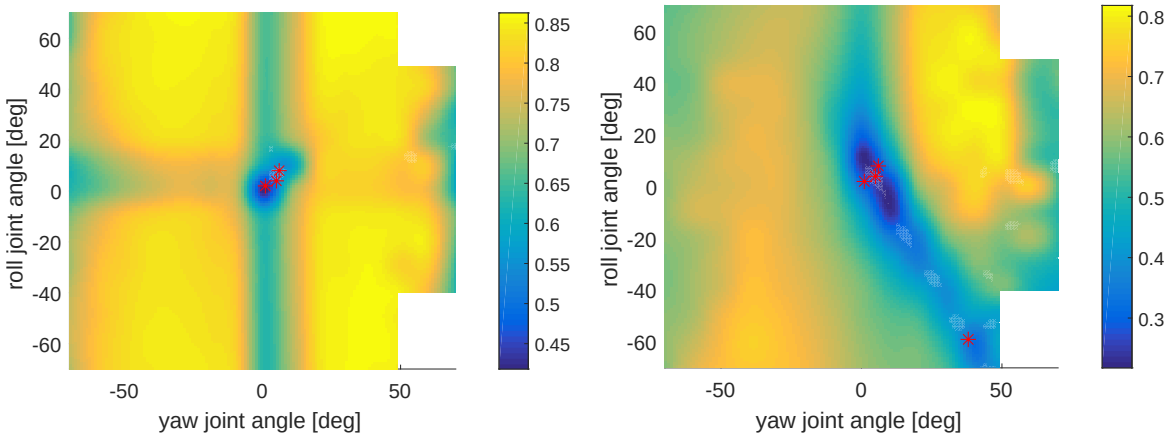
$$\begin{aligned} \max \quad & \Lambda_m(\hat{\bar{\theta}}) \\ \text{subject to} \quad & \lambda^l < \hat{\bar{\theta}} < \lambda^u, \end{aligned} \tag{4.18}$$

where $\bar{\theta}^l$ and $\bar{\theta}^u$ are the upper and lower bounds, respectively, of the mechanism input angles. Figure 4.4 visualizes the next-best-view cost from (4.17) for a two degree of freedom (DOF) mechanism, and also illustrates the mutual information based NBV optimization and selection process over 4 measurement collections.

Finally, it should be noted that, in order to extend this approach for problems with additional estimation parameters, we can simply replace the Jacobian blocks corresponding to the calibration parameters and mechanism joint angles with the Jacobian blocks of any other parameters of interest.

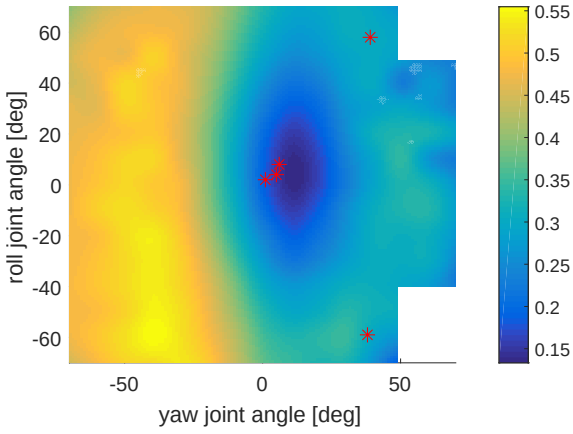
4.5 Relationship to Entropy Minimization Approach

We shall now investigate the relationship between the mutual information maximization, and entropy based minimization approaches for NBV selection. First, recall the covariance matrix from Equation (4.14), $\Sigma_{\widetilde{\nu},\hat{\bar{\theta}}}$, which represents the uncertainty in both the estimation

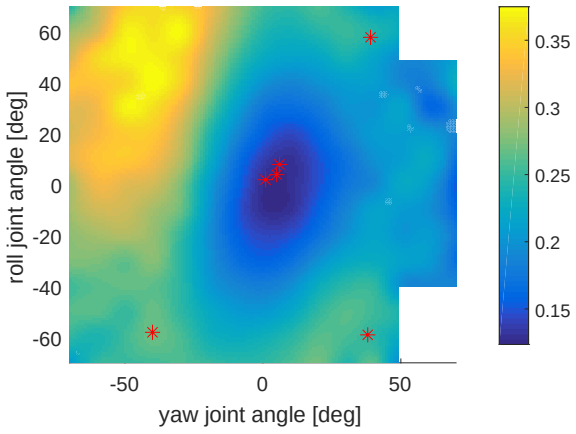


(a) Initial Surface

(b) 1 NBV Addition



(c) 2 NBV Additions



(d) 3 NBV Additions

Figure 4.4: Progression of the mutual information based NBV cost surface for a two degree-of-freedom mechanism, as measurements are added from the NBV configurations. In this case, the autonomous calibration is initialized using 3 configurations, then 3 additional measurements are added using the proposed mutual information maximization approach. The red asterisks denote configurations from which the collected measurements were used for parameter estimation. (a) shows the cost surface after initialization, while (b)-(d) depict the changing cost surface as NBV measurements are added. Note that the plots are coloured according to the mutual information in nats.

parameters and the candidate joint angle measurement, has the form

$$\Sigma_{\widetilde{\nu},\hat{\tilde{\theta}}} = \left(\left[\begin{array}{c|c} \mathbf{J}_{\widetilde{\nu},\eta}(\hat{\tilde{\theta}}) & \mathbf{J}_{\hat{\tilde{\theta}}} \\ \hline 0 & -I \end{array} \right]^T \left[\begin{array}{c|c} \mathbf{J}_{\widetilde{\nu},\eta}(\hat{\tilde{\theta}}) & \mathbf{J}_{\hat{\tilde{\theta}}} \\ \hline 0 & -I \end{array} \right] \right)^{-1} \tag{4.19}$$

$$= \left(\left[\begin{array}{c|c} (\mathbf{J}_{\widetilde{\nu},\eta}(\hat{\tilde{\theta}}))^T \mathbf{J}_{\widetilde{\nu},\eta}(\hat{\tilde{\theta}}) & (\mathbf{J}_{\widetilde{\nu},\eta}(\hat{\tilde{\theta}}))^T \mathbf{J}_{\hat{\tilde{\theta}}} \\ \hline (\mathbf{J}_{\hat{\tilde{\theta}}})^T \mathbf{J}_{\widetilde{\nu},\eta}(\hat{\tilde{\theta}}) & (\mathbf{J}_{\hat{\tilde{\theta}}})^T \mathbf{J}_{\hat{\tilde{\theta}}} + \mathbf{I} \end{array} \right] \right)^{-1} \tag{4.20}$$

$$= \left[\begin{array}{c|c} \Delta_A & \Delta_B \\ \hline \Delta_C & \Delta_D \end{array} \right]. \tag{4.21}$$

Using the *Schur Complement* [90], we can relate the block partitions from Equation (4.21) to the marginal covariances from Equation (4.14) as,

$$\Sigma_{\widetilde{\nu}\widetilde{\nu}} = (\Delta_A - \Delta_B \Delta_D^{-1} \Delta_C)^{-1} \tag{4.22}$$

$$\Sigma_{\hat{\tilde{\theta}}\hat{\tilde{\theta}}} = (\Delta_D - \Delta_C \Delta_A^{-1} \Delta_B)^{-1}. \tag{4.23}$$

Next, let us manipulate the mutual information expression from Equation (4.16). Using the determinant inverse identity [90], we can express the determinant of the inverse covariance matrix as,

$$|\Sigma^{-1}_{\widetilde{\nu},\hat{\tilde{\theta}}}| = \frac{1}{|\Sigma_{\widetilde{\nu},\hat{\tilde{\theta}}}|}, \tag{4.24}$$

which can be substituted into Equation (4.16), yielding

$$I(\widetilde{\nu}; \hat{\tilde{\theta}}) = \frac{1}{2} \log(|\Sigma_{\widetilde{\nu}\widetilde{\nu}}||\Sigma_{\hat{\tilde{\theta}}\hat{\tilde{\theta}}}||\Sigma^{-1}_{\widetilde{\nu},\hat{\tilde{\theta}}}|). \tag{4.25}$$

Using the results derived from the Schur Complement [90], it is also possible to write

the determinant, $|\Sigma^{-1}_{\widetilde{\nu},\hat{\widetilde{\theta}}}|$ as,

$$|\Sigma^{-1}_{\widetilde{\nu},\hat{\widetilde{\theta}}}| = |(\Delta_A)(\Delta_D - \Delta_C\Delta_A^{-1}\Delta_B)| \tag{4.26}$$

$$= |\Delta_A||(\Delta_D - \Delta_C\Delta_A^{-1}\Delta_B)| \tag{4.27}$$

Substituting the terms for the marginal covariances from Equations (4.22) and (4.23), and the determinant identity for the inverse covariance matrix from Equation (4.26), into the mutual information expression from Equation (4.25) yields,

$$I(\widetilde{\nu};\hat{\widetilde{\theta}}) = \frac{1}{2}\log\frac{|\Delta_A|}{|\Delta_A - \Delta_B\Delta_D^{-1}\Delta_C|}. \tag{4.28}$$

Finally, observe that the numerator term in Equation (4.28) can be expressed as

$$|\Delta_A| = |(\mathbf{J}_{\widetilde{\nu},\eta}(\hat{\widetilde{\theta}}))^T\mathbf{J}_{\widetilde{\nu},\eta}(\hat{\widetilde{\theta}})|, \tag{4.29}$$

which is the inverse of the entropy cost given in Equation (4.6). The minimization of the NBV entropy cost therefore maximizes $|\Delta_A|$, and in turn, maximizes the the mutual information cost presented in Equation (4.28). Thus, we see that the entropy minimization strategy presented in Section 4.2 also selects next-best-view configurations which maximize the mutual information between the estimation parameters and the candidate joint angle, and we would expect both approaches to perform similarly. This insight is corroborated in Figures 4.2 and 4.4, which demonstrate that the entropy and mutual information based approaches pick similar NBV configurations when given the same calibration problem.

4.5.1 Relationship to Degeneracy

Let us denote the product inside of the log operation of Equation (4.25) as

$$\mathcal{A} = |\Sigma_{\widetilde{\nu}\widetilde{\nu}}||\Sigma_{\hat{\theta}\hat{\theta}}||\Sigma^{-1}_{\widetilde{\nu},\hat{\theta}}| \tag{4.30}$$

Recall that the matrix $\Sigma_{\widetilde{\nu}\widetilde{\nu}}$ represents the covariance matrix of the calibration parameters, which is the inverse of the system Hessian presented in Equation (3.16). As discussed in Chapter 3, this Hessian matrix must be invertible in order for the system to be non-degenerate, and therefore, its determinant must be non-zero. Conversely, if the system is degenerate, then the determinant of the matrix $\Sigma_{\widetilde{\nu},\widetilde{\nu}}$ becomes zero, which results in $\mathcal{A}$ evaluating to zero, and an undefined mutual information score.

The presented mutual information based NBV approach seeks configurations which *maximize* the mutual information score, and thus looks to maximize the quantity denoted by $\mathcal{A}$. Therefore, this approach selects measurement configurations which steer the system away from degeneracy, and degeneracies that are caused by specific system configurations are systematically avoided.

4.6 Experimental Results

This section presents experimental results which validate the NBV calibration approaches presented in Sections 4.2 and 4.4. First, we shall investigate the calibration performed in a simulation environment, and second, we will discuss the results of performing the DCC calibration using physical hardware. The experimental platforms used for the physical hardware experiments are the same as those used in Chapter 3, and is depicted in Figure 3.17.

4.6.1 Simulation Experiments

To validate our next-best-view approach, we generate the same 2-,3-, and 5-DOF mechanism in simulation, as described in Section 3.6.1, and perform the NBV calibration using measurements from simulated fiducial targets. In all three cases, we generate a static camera which is fixed in the world, and a moving camera which is attached to the end effector of each actuated mechanism.

The 5-DOF mechanism simulates a five degree-of-freedom robotic manipulator, similar to the one shown in Figure 3.17(a), with a static camera mounted to the robot base, and the dynamic camera attached to its end effector. The 3-DOF mechanism simulates a 3-axis gimbal, similar to the one shown in Figure 3.17(b), and allows the camera to perform yaw, pitch, and roll motions. Finally, The 2-DOF mechanism is similar to the 3-DOF system, except it only allows for yaw and roll motions of the dynamic camera. Gaussian distributed noise with 0.5 deg std are added to all joint angle encoder measurements. Note that the mechanisms are generated with translational offsets between links, which allow for the dynamic camera to undergo translation and rotation when the mechanism inputs are excited. The simulated cameras use a pinhole intrinsic lens model, which includes radial and tangential distortion. Realistic intrinsic model parameters, determined by calibrating the cameras depicted in the hardware set-up from Figure 3.17, are used, and Gaussian distributed pixel noise with 0.25 std deviation are added to the simulated image measurements.

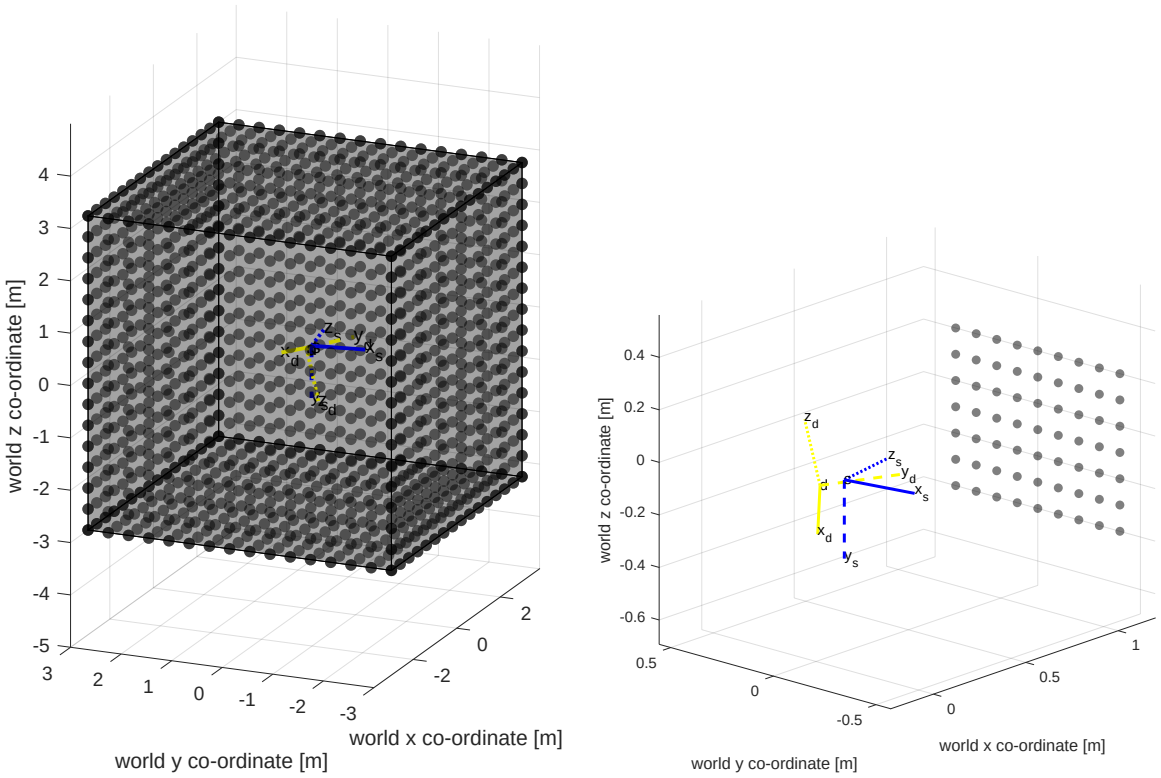
In order to demonstrate how the automatic next-best-view approaches can be used in a field calibration setting, we compare our method to two other view-point selection strategies which could easily be performed by a non-expert human operator in the field. The first competing strategy simply selects random viewpoints within the bounds of the configuration space of the mechanism. The second competing strategy discretizes the configuration space of the mechanism using a linear spacing. For example, suppose we wish

to collect measurements from a 2-DOF mechanism with angle bounds of -20 to 20 degrees for each axis. Using a spacing of 20 degrees, the set of M desired viewpoint angles generated using this linear spacing approach is $\lambda_{1:M} = \{(-20,-20),\ (-20,0),\ (-20,20),\ (0,-20),\ (0,0),\ (0,20),\ (20,-20),\ (20,0),\ (20,20)\}$. Note that linear spacing based selections are sampled from the joint angle space as opposed to the two dimensional image space, as the former offers a richer set of configurations to consider for maximum covariance reduction. For each tested mechanism, the number of collected viewpoints was selected to be the same, in order to compare the entropy reduction for each approach after collecting the same number of measurements.

We also compare the effect of two different calibration fiducial targets, the *calibration plane* and the *calibration box*. The calibration plane is a fairly typical set-up used within the robotics and computer vision communities, where a fiducial target of known scale is printed on a flat plane, and the viewpoints of the observing cameras are manipulated in order to collect a rich set of measurements of the plane from multiple angles and distances. The calibration box, on the other hand, is an extension of the calibration plane approach, where six planar targets are assembled to form a cube, and the device to be calibrated is positioned within the cube. Such an approach provides the advantage of similar measurement collection quality, regardless of the camera viewpoint. The two set-ups are depicted in Figure 4.5.

Figures 4.6 and 4.7 present the calibration box results, and plot the entropy and trace of the covariance matrix, respectively, as the number of views used for the calibration is increased. Note that the covariance matrix trace is an alternative metric to quantify the overall uncertainty, and is included in these results to further demonstrate the effectiveness of our NBV approaches. Tables 4.1 and 4.2 present the entropy and trace scores, for the final view-point addition, of the tested mechanism configurations.

For the calibration box case, all tested methods provide similar performance with re-



(a) Calibration box

(b) Calibration plane

Figure 4.5: (a) shows the calibration box set-up, while (b) depicts the calibration plane set-up. In both cases, the static and dynamic camera frames are drawn in order to illustrate the DCC position relative to the target points. Note that the calibration box from (a) is constructed from six square targets assembled in a cube configuration.

Table 4.1: Summary of the calibration box entropy scores at the final view point addition, for the tested DCC configurations. The performance of the tested methods are fairly comparable, as they converge to similar scores.

	2-DOF	**3-DOF**	**5-DOF**
Linear Spacing	3.57	-4.62	-8.50
Random	3.66	-4.31	-7.98
Entropy	3.12	-4.78	-8.87
Mutual Information	3.61	-4.78	-8.62

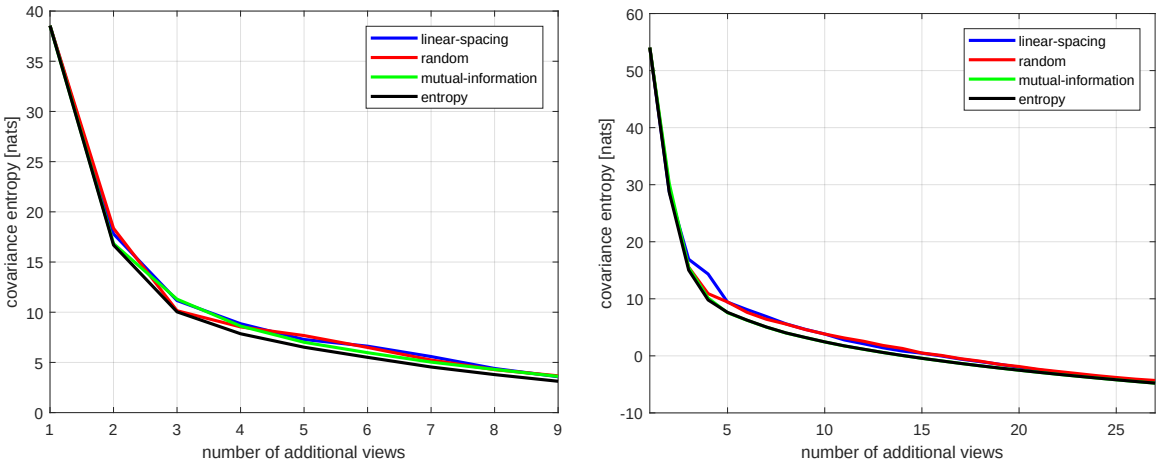
Table 4.2: Summary of the calibration box trace scores at the final view point addition, for the tested DCC configurations. The performance of the tested methods are fairly comparable, as they converge to similar scores.

	2-DOF	**3-DOF**	**5-DOF**
Linear Spacing	1.46	0.58	0.66
Random	1.50	0.63	0.71
Entropy	1.28	0.56	0.62
Mutual Information	1.49	0.57	0.65

spect to parameter uncertainty reduction. This result corroborates that the calibration box provides image measurements with similar information content, regardless of viewing angle. The rapid trace score decreases exhibited by the linear spacing strategy in Figure 4.7 are due to that strategy holding certain joint angles fixed over multiple viewpoint additions, and once that joint is actuated, a sharp decrease in the trace score occurs due to the improvement in measurement excitation.

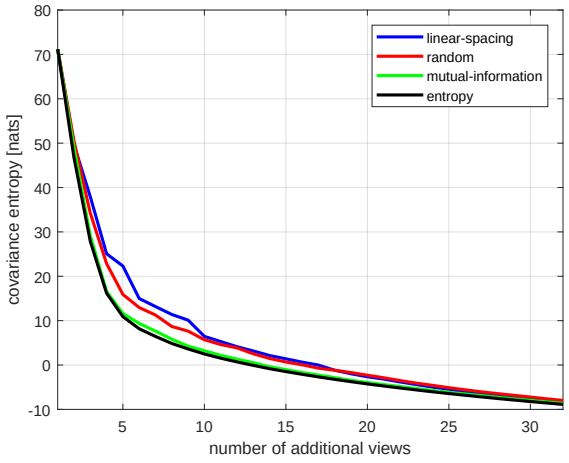
Figures 4.8 and 4.9 present the calibration plane results, and plot the entropy and trace of the covariance matrix, respectively, as the number of views used for the calibration is increased. Tables 4.3 and 4.4 present the entropy and trace scores, for the final view-point addition, of the tested mechanism configurations.

It is evident that for all tested mechanisms, the next-best-view approaches provide the lowest covariance matrix entropy and trace scores over all collected measurements. Our approaches are able to provide lower parameter uncertainty, when compared to the ran-



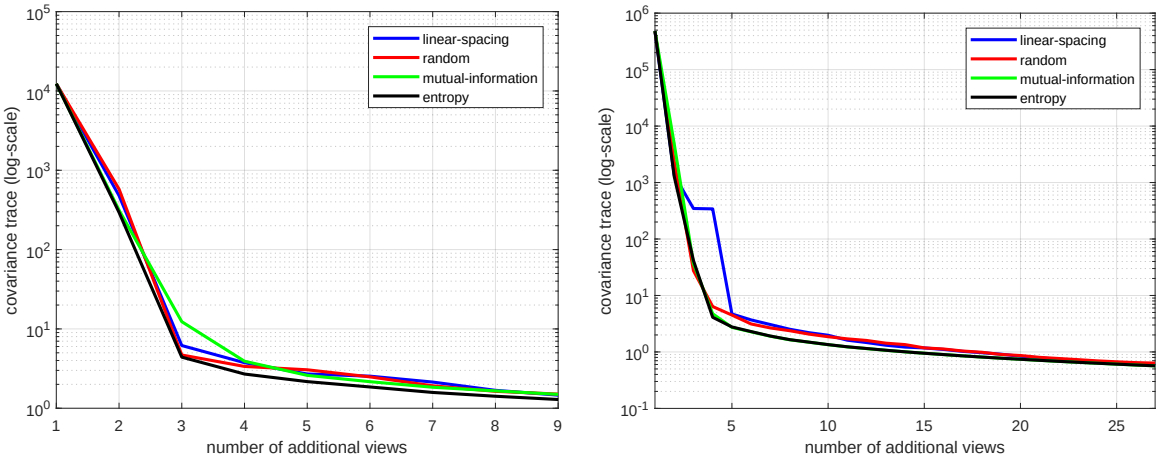
(a) 2-DOF

(b) 3-DOF



(c) 5-DOF

Figure 4.6: Calibration box: comparison of the parameter covariance entropy versus the number of views, for the random sampling, linear spacing, and next-best-view approaches. (a)-(c) present the results for the 2-,3-,and 5-DOF cases, respectively.



(a) 2-DOF

(b) 3-DOF

(c) 5-DOF

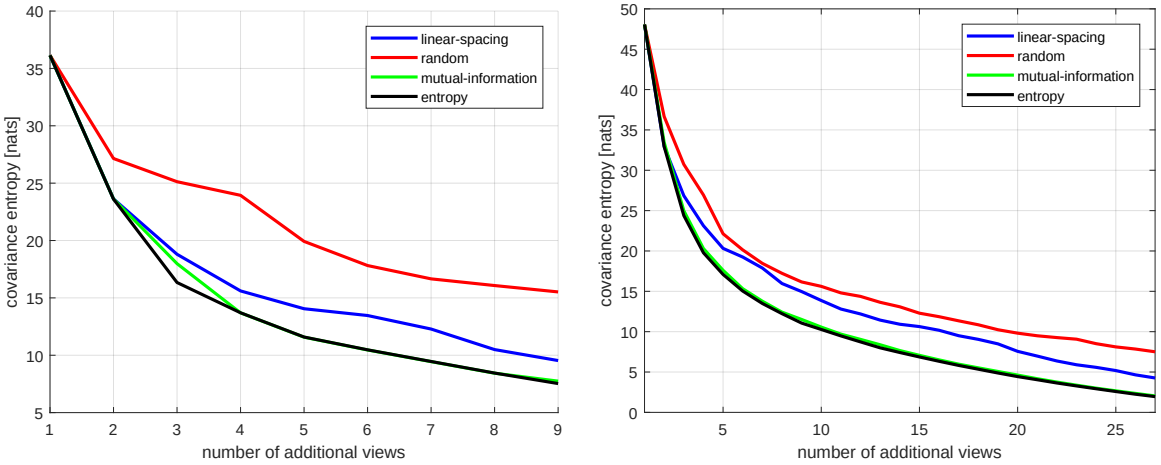
Figure 4.7: Calibration box: comparison of the parameter covariance entropy versus the number of views, for the random sampling, linear spacing, and next-best-view approaches. (a)-(c) present the results for the 2-,3-,and 5-DOF cases, respectively.

dom and linear spacing strategy, as the next-best-view method seeks the measurement configurations at each iteration that directly improve the uncertainty of the estimation parameters. Although the random sampling and linear spacing techniques are also able to provide a decrease in parameter uncertainty, those approaches do not evaluate the selected viewpoint, and therefore may include measurements which do not provide significant entropy reduction. This behaviour is evident in Figure 4.9(a), where between measurements 3 and 4, a random viewpoint is selected that provides little improvement of the covariance matrix trace score. The larger reduction in entropy for the random sampling approach compared to the linear spacing approach, as shown in Figures 4.8(c) and 4.9(c), is due to the random approach selecting a richer set of configurations from diverse viewpoints for the 5-DOF case.

From Figures 4.8 and 4.9, it is also evident that for a desired target covariance score, the next-best-view approaches are able to achieve the target using fewer viewpoints than the random and linear spacing strategies. For example, in Figure 4.8(c), a target covariance entropy of 25 nats is achieved by the next-best-view approach using 8 views, while the linear spacing and random strategies required 25 and 20 views, respectively, to achieve the same calibration quality. Since the next-best-view entropy and trace scores are less than the scores generated using the random and linear spacing methods, our results demonstrate that for any parameter uncertainty target, our next best view approaches will be able to achieve the target with fewer collected measurement views compared to the competing strategies.

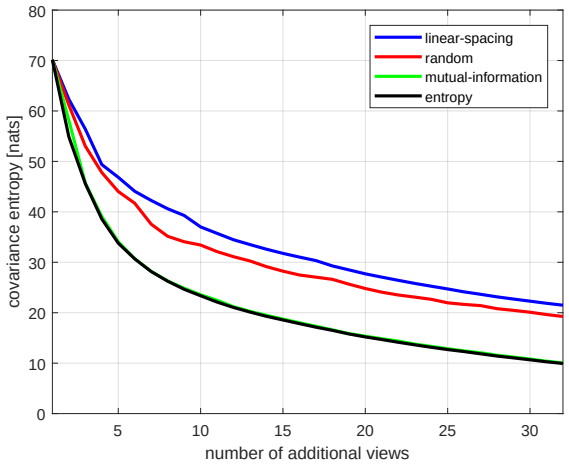
4.6.2 Physical Hardware Experiments

The next-best-view approaches are also validated using a physical, 3-DOF gimbal based DCC, that allows for yaw, pitch, and roll motions of the dynamic camera, and a 5-DOF manipulator based DCC, as depicted in Figure 3.17. We perform a similar experiment to



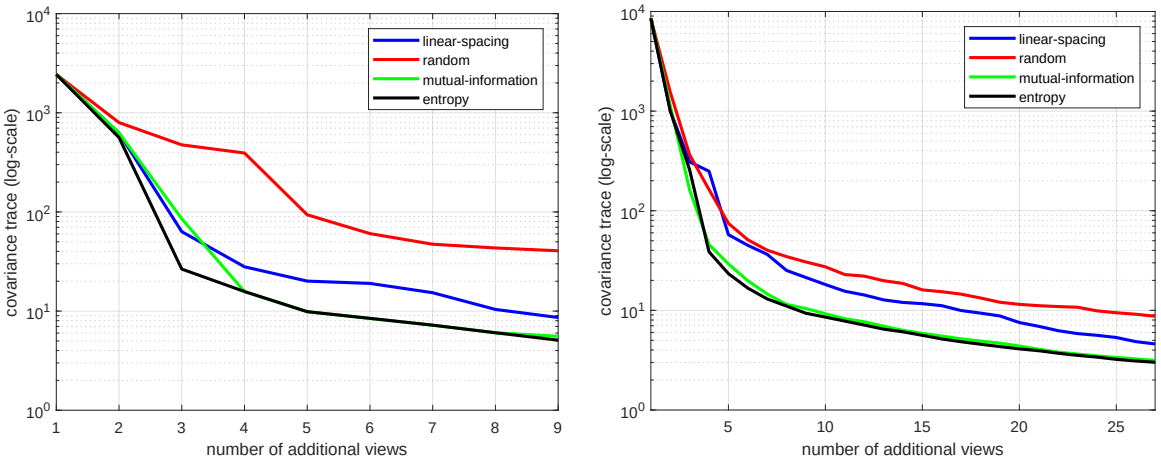
(a) 2-DOF

(b) 3-DOF



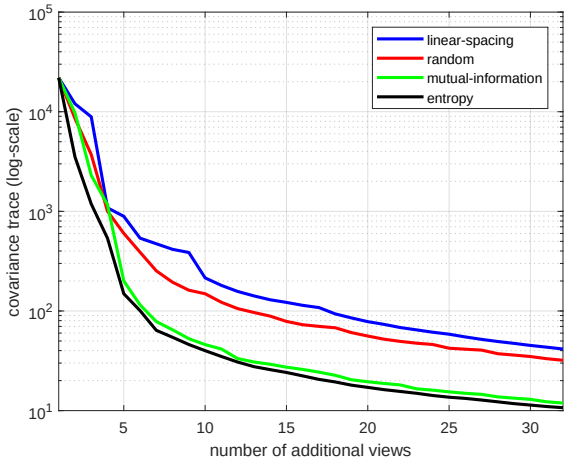
(c) 5-DOF

Figure 4.8: Calibration plane: comparison of the parameter covariance entropy score versus the number of views, for the random sampling, linear spacing, and next-best-view approaches. (a)-(c) present the results for the 2-,3-,and 5-DOF cases, respectively.



(a) 2-DOF

(b) 3-DOF



(c) 5-DOF

Figure 4.9: Calibration plane: comparison of the parameter covariance trace score versus the number of views, for the random sampling, linear spacing, and next-best-view approaches. (a)-(c) present the results for the 2-,3-,and 5-DOF cases, respectively.

Table 4.3: Summary of the calibration plane entropy scores at the final view point addition, for the tested DCC configurations. Note that the entropy and mutual information methods achieve the best score.

	2-DOF	**3-DOF**	**5-DOF**
Linear Spacing	9.54	4.27	21.48
Random	15.51	7.50	19.25
Entropy	7.53	1.95	9.93
Mutual Information	7.74	2.04	10.05

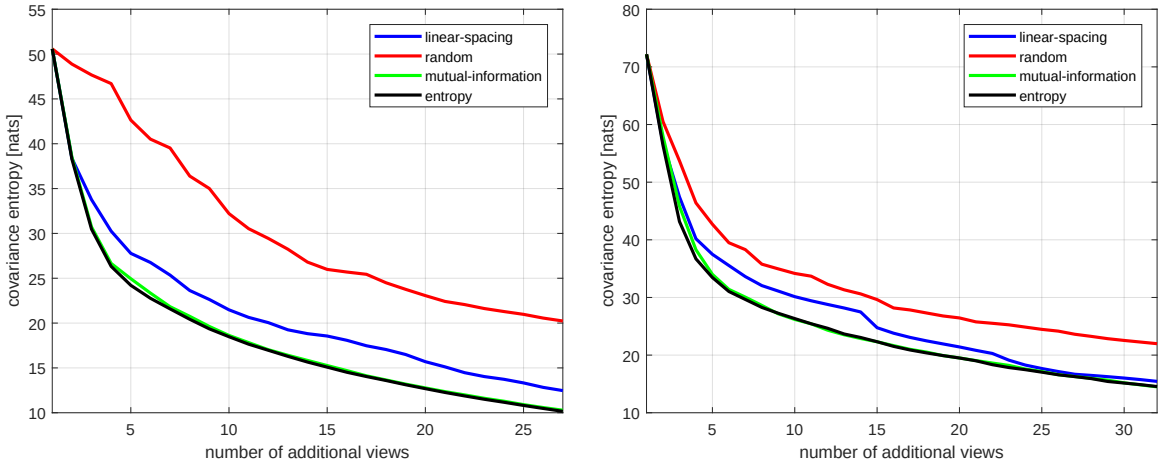
Table 4.4: Summary of the calibration plane trace scores at the final view point addition, for the tested DCC configurations. Note that the entropy and mutual information methods achieve the best score.

	2-DOF	**3-DOF**	**5-DOF**
Linear Spacing	8.61	4.60	41.54
Random	40.56	8.73	31.99
Entropy	5.09	3.01	10.69
Mutual Information	5.56	3.14	11.90

that presented in Section 4.6.1, where we compare the next-best-view approaches to the linear spacing and random sampling viewpoint selection methods. Our cameras are fitted with 120 degree field-of-view lenses, which allow for sufficient overlapping view between the static and dynamic camera, even when the joint angles are excited over a wide range.

Figures 4.10 and 4.11 plot the parameter covariance matrix entropy and trace scores, respectively, versus the number of collected views for each of the strategies tested. Tables 4.1 and 4.2 present the entropy and trace scores, for the final view-point addition, of the tested mechanism configurations. We see that the next-best-view approaches maintain the lowest entropy over all of the collected viewpoints, and exhibit a rapid rate of decreasing entropy over the first few collected images.

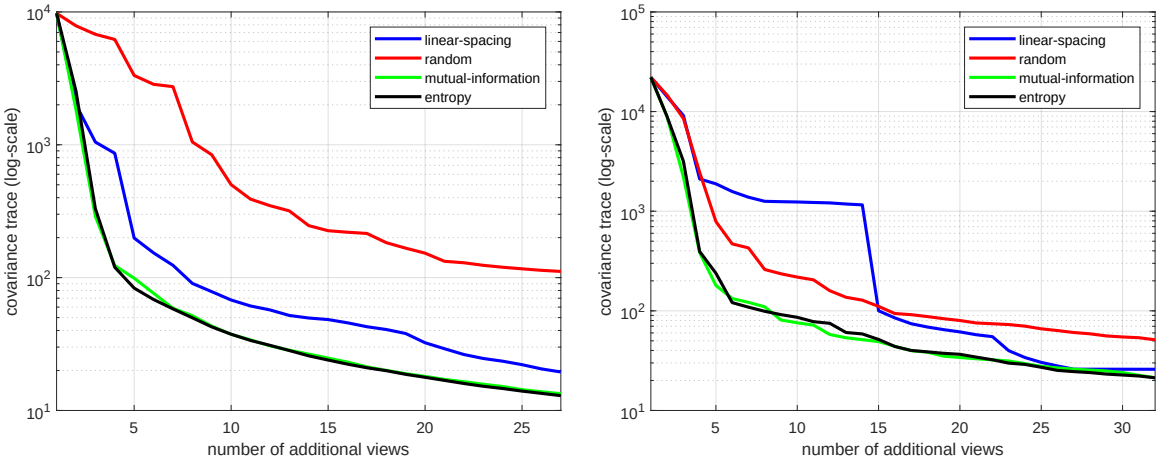
In order to validate the quality of the next-best-view approaches, the competing methods are used to generate calibrations and are evaluated against an independently collected validation data set, as described in Chapter 3. The RMSE pixel error of the validation



(a) 3-DOF (b) 5-DOF

Figure 4.10: Comparison of the parameter covariance entropy score versus the number of views, for the random sampling, linear spacing, and next-best-view approaches. (a)-(b) present the results for the 3-,and 5-DOF cases, respectively, as performed using the physical set-up shown in Figure 3.17.

set are plotted against the number of viewpoints used for the calibration, as presented in Figure 4.12. It is evident that for both the 3-DOF and 5-DOF cases, the mutual information and entropy based NBV approaches select measurements which result in high quality calibrations, as the RMSE pixel error of the validation set converge rapidly to low values, compared to the random and linear spacing based approaches. For the 3-DOF case, we see that the NBV selection methods produce a high quality calibration using just seven viewpoints, and the linear spacing and random methods require 16 and 27 viewpoints, respectively, to achieve a calibration of comparable quality. Similarly, for the 5-DOF case, a high quality calibration is achieved after five viewpoints, and the random and linear spacing techniques require 15 and 22 viewpoints, respectively, to generate calibrations of similar quality.



(a) 3-DOF (b) 5-DOF

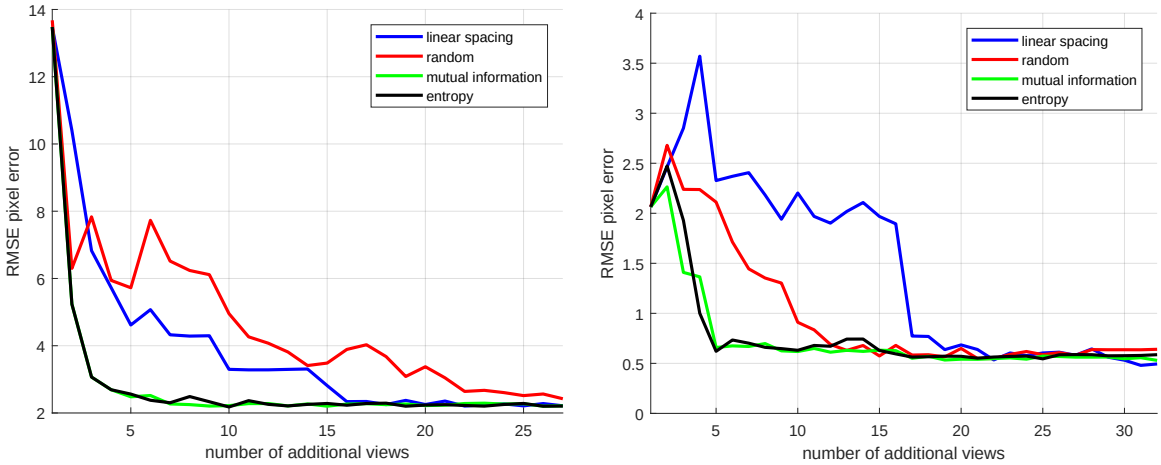
Figure 4.11: Comparison of the parameter covariance trace score versus the number of views, for the random sampling, linear spacing, and next-best-view approaches. (a)-(b) present the results for the 3-,and 5-DOF cases, respectively, as performed using the physical set-up shown in Figure 3.17.

	3-DOF	**5-DOF**
Linear Spacing	12.47	15.44
Random	20.23	21.98
Entropy	10.15	14.45
Mutual Information	10.28	14.49

Table 4.5: Summary of the covariance entropy scores at the final view point addition, for the physical DCC set-ups.

	3-DOF	**5-DOF**
Linear Spacing	19.48	25.90
Random	111.31	51.20
Entropy	12.92	21.27
Mutual Information	13.36	21.27

Table 4.6: Summary of the covariance trace scores at the final view point addition, for the physical DCC set-ups.



(a) 3-DOF (b) 5-DOF

Figure 4.12: RMSE pixel error of the validation set, plotted against the number of views used to generate the calibration parameters. (a) and (b) show the reprojection error results for the 3-DOF Aeryon gimbal DCC, and 5-DOF Fanuc manipulator DCC, respectively. Note that the entropy and mutual information based approaches converge rapidly to low RMSE pixel values, compared to the linear spacing and random method.

Chapter 5

Keyframe Selection using Point Entropy Methods

In this chapter, we discuss the informed selection of key-frames for inclusion in the SLAM back-end, which also contributes significantly to the overall performance of visual SLAM methods by improving the quality of the global map that results from performing bundle adjustment. The accuracy of the map point parameter estimation is heavily dependent on the triangulation baseline between the measurement viewpoints. If the point is initially triangulated using narrow baseline observations, considerable uncertainty in point depth estimates result. However, if the point is re-observed with a wider baseline, the depth estimate can be resolved with increased accuracy. Many visual SLAM techniques use heuristics based on point triangulation baseline to perform key-frame insertion. However no existing approaches attempt to perform key-frame selection through direct minimization of the point estimate covariance. To determine the best multi-keyframe for insertion, this chapter will propose two main selection metrics: the cumulative point entropy reduction (CPER), which maximizes the expected entropy reduction in the existing map points, and the point pixel flow discrepancy (PPFD), which assesses the expected future features that

can be added to the map.

5.1 Selection Based on Cumulative Point Entropy Reduction (CPER)

The most accurate way to update the point covariance would be to integrate all the measurements from each observing key-frame, effectively eliminating the benefit of key-frame selection by performing bundle adjustment on every tracked image. However, such an approach for point covariance evaluation is very computationally expensive, and is not suitable for a real-time mapping and tracking solution. Instead, we propose to select a key-frame by evaluating the expected information gain from each potential candidate image set since the previous multi-keyframe insertion. The expected information gain will be computed by comparing the expected entropy reduction over the map points, with the assumption that the key-frame candidate's location is known and fixed. Although the key-frame's pose parameters are in-fact updated through bundle adjustment once inserted into the map, the fixed key-frame parameter assumption allows for rapid evaluation of the point's entropy reduction, and is reasonable so long as the tracker pose estimate is sufficiently accurate.

In order to determine when a key-frame should be inserted into the map, we inspect the uncertainty of the camera cluster provided by the tracking process at current time k. The covariance of the tracking pose parameters is given by $\boldsymbol{\Sigma}_c^k = (\boldsymbol{J}_{c^k}^T \Omega \boldsymbol{J}_{c^k})^{-1}$, where $\boldsymbol{J}_{c^k}$ is the Jacobian of the map reprojection error with respect to the cluster state, and Ω is the measurement information matrix. Note that $\boldsymbol{J}_{c^k}$ has m rows, representing m stacked measurements collected across all of the cluster cameras. To assess the current tracking performance, we extract the diagonal elements of covariance matrix $\boldsymbol{\Sigma}_c^k$, which reflect the individual uncertainty of each of the tracker state parameters, σ_{t_x}, σ_{t_y}, σ_{t_z}, σ_{ϕ_x}, σ_{ϕ_y}, and

σ_{ϕ_z}, where t_x, t_y, and t_z are the translational perturbations according to the x, y, and z directions, and ϕ_x, ϕ_y, and ϕ_z denote the rotation perturbations in the tangent space, for the current linearization point. Finally, a key-frame is added when any element of the positional entropy is above a user defined threshold, ϵ, or

$$\max(h_e(\sigma_{t_x}), h_e(\sigma_{t_y}), h_e(\sigma_{t_z}), h_e(\sigma_{\phi_x}), h_e(\sigma_{\phi_y}), h_e(\sigma_{\phi_z})) > \epsilon, \tag{5.1}$$

where $h_e(\cdot)$ is computed using Equation (2.56). When a multi-keyframe addition is triggered, the next step is to determine which multi-keyframe should be added. For this, multi-keyframe candidates are maintained in a buffer and scored based on the expected reduction in point entropy if added to the map through a bundle adjustment process.

Suppose the tracking thread is currently operating at time k, and the last multi-keyframe insertion occurred at time k_0. Denote the set of multi-keyframe candidates which are buffered between times k_0 and k as

$$\Phi = \{M_{k_0}, M_{k_0+1}, M_{k_0+2}, \ldots, M_k\}. \tag{5.2}$$

Since each of the multi-keyframe candidates are saved from the tracking thread, an estimate of the global pose of each candidate is available from the tracking solution. Denote the set of map points from $\boldsymbol{P}$, visible in $K_l \in M_i$, as $\tilde{\boldsymbol{P}}^{il} \subset \boldsymbol{P}$.

In the bundle adjustment process, the point parameters are modeled as a Gaussian distribution with an associated mean and covariance. Denote the estimate for the point in the world frame, ${}^{\mathtt{w}}\boldsymbol{p}$ as ${}^{\mathtt{w}}\hat{\boldsymbol{p}}_j$, and the associated covariance matrix $\boldsymbol{\Sigma}_j \in \mathbb{R}^{3\times3}$. Suppose point ${}^{\mathtt{w}}\boldsymbol{p}_j \in \tilde{\boldsymbol{P}}^{il}$ is measured in key-frame $K_l \in M_i$. Our method seeks to determine the updated covariance of point ${}^{\mathtt{w}}\boldsymbol{p}_j$, if triangulated using an additional measurement from key-frame K_j. This is accomplished using a covariance update step, similar to the Extended Kalman Filter.

Denote the Jacobian of the reprojection function, with respect to the point parameters, evaluated at point ${}^{\mathtt{w}}\hat{\boldsymbol{p}}_j$, as

$$\boldsymbol{J}_{\boldsymbol{p}_j} = \frac{\partial \Psi(\mathbf{T}_k^{\mathtt{c_b}:\mathtt{w}}\, {}^{\mathtt{w}}\boldsymbol{p}_j)}{\partial\, {}^{\mathtt{w}}\boldsymbol{p}_j}|_{{}^{\mathtt{w}}\hat{\boldsymbol{p}}_j}. \tag{5.3}$$

The Jacobian, $\boldsymbol{J}_{\boldsymbol{p}_j}$, describes how perturbations in the point parameters for ${}^{\mathtt{w}}\hat{\boldsymbol{p}}_j$ map to perturbations in the image reprojections. Using the Jacobian, $\boldsymbol{J}_{\boldsymbol{p}_j}$, and the prior point covariance $\boldsymbol{\Sigma}_j$, the predicted point covariance is given as

$$\bar{\boldsymbol{\Sigma}}_j = (\mathbf{I} - \boldsymbol{\Sigma}_j \boldsymbol{J}_{\boldsymbol{p}_j}^T (\boldsymbol{J}_{\boldsymbol{p}_j} \boldsymbol{\Sigma}_j \boldsymbol{J}_{\boldsymbol{p}_j}^T + \mathbf{R})^{-1} \boldsymbol{J}_{\boldsymbol{p}_j}) \boldsymbol{\Sigma}_j. \tag{5.4}$$

Note that the prior point covariance, $\boldsymbol{\Sigma}_j$, is available from the most recent bundle adjustment. The predicted covariance, $\bar{\boldsymbol{\Sigma}}_j$, provides an estimate of the covariance for point, ${}^{\mathtt{w}}\boldsymbol{p}_j$, if the observing key-frame was inserted into the bundle adjustment process. Equation (5.4) can be evaluated rapidly for each point, as the computational bottleneck is the inversion of a 3×3 matrix. To asses the reduction in uncertainty, we evaluate the entropy of the predicted covariance matrix from Equation (5.4). Similar to the strategy used in Chapter 4 to perform next-best-view data collection, we use the entropy of the covariance matrix as a smooth cost which quantifies the overall uncertainty of the parameter estimate. Furthermore, the point entropy provides a convenient way to assess the uncertainty reduction across all of the points observed in the multi-keyframe, as the entropy score from individual points can be easily summed.

Denote the entropy corresponding to the point's prior and predicted covariance as $h_e({}^{\mathtt{w}}\hat{\boldsymbol{p}}_j)$ and $\bar{h}_e({}^{\mathtt{w}}\hat{\boldsymbol{p}}_j)$, respectively. The reduction in entropy for point ${}^{\mathtt{w}}\boldsymbol{p}_j$ is given as

$$\Upsilon({}^{\mathtt{w}}\boldsymbol{p}_j) = h_e({}^{\mathtt{w}}\hat{\boldsymbol{p}}_j) - \bar{h}_e({}^{\mathtt{w}}\hat{\boldsymbol{p}}_j). \tag{5.5}$$

Using Equation (5.5), the expected entropy reduction,Λ_c, for all map points measured in

multi-keyframe M_i is

$$\Lambda_c(M_i) = \sum_{K_l \in M_i} \sum_{\boldsymbol{p} \in \tilde{\boldsymbol{P}}^{il}} \Upsilon(\boldsymbol{p}). \tag{5.6}$$

Finally, when a multi-keyframe insertion is requested, all multi-keyframes within the buffer Φ are scored using Equation (5.6). The multi-keyframe selected for insertion, M^*, is the one from the buffer which maximizes the point entropy reduction:

$$M^* = \underset{M_i \in \Phi}{\operatorname{argmax}} \ \Lambda_c(M_i). \tag{5.7}$$

Once selected, M^* is inserted into the map through bundle adjustment, and the multi-keyframe buffer, Φ, is reset.

Note that our proposed approach treats each point independently, as the covariance matrix from Equation (5.4), is only for the parameters of point ${}^{\mathtt{w}}\hat{\boldsymbol{p}}_j$, and ignores the correlation the point may have with other points in the map. Consider that if two points are correlated, then improvement of the point estimate for one of the points would improve the point estimate of the other point as well. This additional correlation is captured in the *mutual information* [85] between points. The independence approximation is suitable for our application, as it provides a lower bound on the overall expected entropy reduction of the map, using the known camera motion assumption. The mutual information which exists between points will only further reduce the map entropy after the key-frame is inserted and processed with bundle adjustment.

The entropy reduction for a point described in Equation (5.5), is also dependent on the camera model, as the point projection Jacobian given by Equation (5.3), is dependent on the underlying camera projection equations. For example, wide FOV lenses using the Taylor model spatially compress points near the boundaries of the image plane, while a typical pinhole projection preserves the relative spatial distribution of 3D points. This comparison is illustrated in Figures 5.1(a) and 5.1(b), which display the image projection of

a planar grid of 3D points positioned in-front of the camera. As a result of the compression, points projected using the Taylor model which fall near the boundaries of the image are less sensitive to perturbations of the 3D point location. This amounts to the magnitude of the Jacobian described in Equation (5.3), being attenuated for points which project near the edge of the image. This insight is illustrated in Figures 5.1(c) and 5.1(d), which show the norm of the projection Jacobian with respect to perturbations in the x point parameter. It is evident that the pinhole model maintains uniform sensitivity to point parameter perturbations across the image plane, as the norm of the Jacobian is constant over the entire image, while the Taylor camera model has reduced sensitivity as the points are projected farther from the image center.

Thus, by scoring multi-keyframes based on the expected point entropy reduction, our proposed approach is able to select key-frames based on the triangulation baselines between observations, as well as take into account the camera model and assess key-frames based on the sensitivity of point measurements as well. It should be noted that even though we derived the proposed key-frame selection approach for points parameterized with Euclidean co-ordinates in the global frame, our method can be formulated for a wide range of point parameterizations, such as spherical, relative, etc.

5.2 Selection Based on Point Pixel Flow Discrepancy (PPFD)

Although it is important to strengthen the parameter estimates for the points which already exist within the map, an equally important function for multi-keyframe insertion is to incorporate new map points to track against in the future. Feature matches which exist between the inserted multi-keyframe and the existing multi-keyframes result in new map points which can be used for localization. However, the effectiveness of the newly created

points for tracking is dependent on their location, as well as the motion of the vehicle. For example, the creation of new map points which are not in the current view of the camera cluster, as well as the creation of new map points in a region with a high density of existing map points, are generally not desired. From a tracking perspective, map points should be distributed throughout the image [94, 95], and should be trackable given the current camera motion.

The difficulty in assessing a potentially new map point is that a correspondence is required from an existing key-frame in order to perform the 3D point triangulation. The process of feature matching across key-frames is computationally expensive, thus in order to maintain real-time performance, MCPTAM searches for feature correspondences between key-frames in the mapping thread, prior to bundle adjustment. Such a division allows the tracking thread to localize against the existing map, while the mapping thread searches for new features and performs bundle adjustment. Although it would be possible to develop a heuristic to postulate the effectiveness of an image feature *without* searching for correspondences, performing feature matching in the key-frame evaluation process presents many benefits. First, seeking feature matches allows us to determine the 3D location of the point, which can be used to predict the movement of the point from a camera fixed observation frame. Second, locating matches allows us to directly detect when measurements to a feature are occluded due to obstacles.

We propose a three step method to evaluate multi-keyframes from the multi-keyframe buffer, Φ, for potential map points. In the first step, we evaluate the map points which currently exist in the map and create a probability distribution on the image plane, which captures how those points will move over a specified time horizon and vehicle motion. In step two, we take features from the multi-keyframe candidates and match them against a subset of the existing multi-keyframes to determine the set of *predicted* map points that would appear if the multi-keyframe was inserted into the map. Given the predicted point

locations, we generate a similar probability distribution to step one, except now capturing how we expect the predicted points to move in the image frame. In step three, we compare the two probability distributions, and identify the multi-keyframe from the buffer which best introduces new map points such that the distribution of map points over the image is maximized.

Evaluation of Existing Map Points: In order to evaluate the distribution of the existing map points, we first project them into the image frame corresponding to the current tracker position. We are interested in understanding how the map points will move, or *flow*, in the image, given a time horizon and a predicted camera cluster motion. In some applications, the path of the vehicle is determined using a path planning method, thus the expected motion of the vehicle is known in advance. For this derivation, we shall not assume any known motion, but instead use a constant velocity model. However, the approach is general, and can also be formulated using more complex motion models.

The flow of a map point, ${}^{\mathtt{w}}\boldsymbol{p}_i$, expressed in the frame of observing camera j, ${}^{\mathtt{c_j}}\boldsymbol{p}_i$, will begin at image co-ordinates corresponding to the point reprojection, $\boldsymbol{u}_i = \Psi({}^{\mathtt{c_j}}\boldsymbol{p}_i)$. To determine the flow of the map point for a given motion, we require relationships between the pixel reprojection motion and camera motion. The Jacobian of the image reprojection for point i, with respect to the cluster state at time k, is given as

$$\boldsymbol{J}^i_{c^k} = \frac{\partial \Psi(\mathbf{T}^{\mathtt{c_j:c_b}}\mathbf{T}^{\mathtt{c_b:w}}_k\,{}^{\mathtt{w}}\boldsymbol{p}_j)}{\partial \mathbf{T}^{\mathtt{c_b:w}}_k}. \tag{5.8}$$

If the motion of the camera cluster with respect to time is denoted as $\frac{\partial \mathbf{T}^{\mathtt{c_b:w}}_k}{\partial t}$, then the change of the point projection with respect to time is

$$\frac{\partial \Psi(\mathbf{T}^{\mathtt{c_j:c_b}}\mathbf{T}^{\mathtt{c_b:w}}_k\,{}^{\mathtt{w}}\boldsymbol{p}_j)}{\partial t} = \frac{\partial \Psi(\mathbf{T}^{\mathtt{c_j:c_b}}\mathbf{T}^{\mathtt{c_b:w}}_k\,{}^{\mathtt{w}}\boldsymbol{p}_j)}{\partial \mathbf{T}^{\mathtt{c_b:w}}_k}\frac{\partial \mathbf{T}^{\mathtt{c_b:w}}_k}{\partial t}. \tag{5.9}$$

Using the constant velocity model over a specified time horizon, δ_t, the ending image

point, $\bar{\boldsymbol{u}}_i$, for the reprojection flow, given the motion of the cluster is

$$\bar{\boldsymbol{u}}_i = \frac{\partial \Psi(\mathbf{T}^{\mathtt{c_j:c_b}} \mathbf{T}_k^{\mathtt{c_b:w}} {}^{\mathtt{w}}\boldsymbol{p}_j)}{\partial t} \delta_t. \tag{5.10}$$

Using the pair of start and end points for the reprojection flow $(\boldsymbol{u}_i, \bar{\boldsymbol{u}}_i)$, a discrete probability distribution can be constructed on the image plane, which we will call the *existing point flow PDF (E-PFP)*. The E-PFP captures the probability of *not* measuring existing map points in the image, so areas of high probability denote locations where we would like to *add* new map points. To construct the E-PFP, we begin with a 2D uniform distribution, where the discretization of the PDF is analogous to the pixel resolution of the image. Straight lines are drawn between the reprojection flow pairs, and the 2D locations on the PDF which intersect the flow lines are assigned a value of 0. The flow lines are then dilated and blurred in order to account for motion uncertainty of the camera cluster, and to avoid adding new points in the map which are needlessly close to existing points. The amount of dilation and blurring, as well as the lookahead horizon, are a tunable parameters dependent on the relative point density desired by the user. Finally, the E-PFP is normalized to make it valid probability distribution, such that the sum of all probabilities is equal to 1. A depiction of the E-PFP is given in Figure 5.2(b).

Evaluation of Predicted Map Points: We now wish to find the multi-keyframe from the buffer which will add new image points in high probability areas of the E-PFP. To do so requires the matching of image features from the candidate key-frame to features located in key-frames in the existing map. In order to reduce the computational complexity of the feature matching operation, we first perform a binning operation, similar to what is described in [94, 95]. The images from the candidate multi-keyframe are divided into rectilinear bins, and only the feature with the best score is matched against existing key-frames, which greatly reduces the number of feature matches required. Second, the features are only matched against the key-frames from the N closest multi-keyframes which exist

in the map.

To determine how the predicted map points will flow in the image plane, we construct the *future point flow PDF (F-PFP)*, which is generated in a similar manner to the E-PFP. For each candidate multi-keyframe, we perform feature matching and generate the set of predicted map points, $\bar{\boldsymbol{P}}$. We then compute the pixel flow for the predicted points in $\bar{\boldsymbol{P}}$ using Equation (5.10). The F-PFP is also initialized as a discrete uniform distribution, and regions of the predicted points' pixel flow are assigned a value of 1. Similar to the E-PFP, the flow lines for the predicted points are dilated and blurred in order to account for motion uncertainty of the camera cluster. Finally, the F-PFP is normalized to make it a valid discrete probability distribution. An example F-PFP is shown in Figure 5.2(c).

Scoring the pixel flow PDF: Given the E-PFP for the current multi-keyframe, and the F-PFP's from the buffered multi-keyframes, we now wish to determine which F-PFP is the most similar to the E-PFP. Doing so determines the multi-keyframe from the buffer which best distributes new features in regions of the image plane not already containing existing features, while taking into account the vehicle motion and expected time horizon between key-frame additions. To compare the probability distributions, we compute the relative entropy between the E-PFP and F-PFP using Equation (2.52), as described in Section 2. Recall that the relative entropy is a measure of the difference between PDF's, thus if the two PDFs are identical, the relative entropy will be zero. If we denote the E-PFP as τ, and the set of all F-PFP's from multi-keyframes buffered since time k_0 as

$$\Gamma = \{\beta_{k_0+1},\, \beta_{k_0+2},\, \ldots\, , \beta_k\},\; k = |\Phi|, \tag{5.11}$$

then the optimal F-PFP, β^*, which minimizes the relative entropy, is selected as

$$\beta^* = \underset{\beta \in \Gamma}{\operatorname{argmin}}\; \mathcal{D}(\tau \| \beta). \tag{5.12}$$

5.3 Experimental Results

To validate our proposed key-frame selection methods, we performed two sets of experiments. In the first set of experiments, we simulate a surveillance task, and compare the CPER approach to other common key-frame selection heuristics based on movement thresholds and map point overlap. In the second set of experiments, we demonstrate the effectiveness of the PPFD approach in exploration scenarios involving occlusions and aggressive camera motion.

For both cases, the key-frame selection methods are implemented within the MCPTAM framework. The camera rig was equipped with three Ximea xIQ cameras, arranged in a rigid cluster, with one camera looking forwards, and the others facing off to the left and right sides. The cameras were fitted with wide angle lenses, with 160° field of view, and images were captured at 30 frames per second, at a resolution of 900x600 pixels. The ground truth of the camera motion was collected using an Optitrack Indoor Positioning System (IPS), which is capable of tracking motion at a rate of 100 Hz, with sub centimeter translational accuracy, and sub degree rotational accuracy. The extrinsic calibration between the camera co-ordinate frame and the IPS was determined using an off-line optimization method, as detailed in [5, 11]. Note that although experiments are shown using the MCPTAM algorithm, our key-frame selection methods can be applied to any key-frame based visual SLAM which uses point features.

5.3.1 Evaluation of CPER in a Surveillance Task

We simulate a surveillance task by moving the camera rig in an environment where there are few occlusions, and such that many features can be re-observed throughout the trajectory. The test trajectory is visualized in Figure 5.4. The proposed CPER approach is compared against a distance threshold heuristic and a point overlap heuristic. The former adds a

key-frame when the distance between the current tracking position and the closest key-frame in the map is greater than a user defined threshold, while the latter adds a key-frame when the ratio of measured image features to measured map points falls below a threshold. We tested the CPER method to distance threshold of 0.5m (0.5mt), 1m (1mt), 2m (2mt), and to point percent overlap threshold of 25% (25pt), 50% (50pt) and 60% (60pt). For the CPER method, we set the entropy threshold, ϵ, to -6.0 nats, through empirical evaluation of our approach. The results of the experiment are shown in Figure 5.3 and summarized in Table 5.1.

	Translation MSE [m^2]	**rotation MSE [rad^2]**	**num multi-kf**	**num map points**	**average tracker entropy [nats]**
0.5mt	0.0024	4.33e-05	13	2431	-6.59
1mt	0.0027	5.50e-05	5	1390	-6.22
2mt	0.0039	1.75e-04	4	459	-5.73
25pt	0.0105	0.0019	3	104	-4.7
50pt	0.0023	1.00e-04	11	1886	-6.11
60pt	7.02e-04	1.33e-04	44	3657	-6.55
CPER	1.04e-04	4.12e-05	8	949	-6.56

Table 5.1: Summary of results for surveillance motion sequence.

In general, all of the tested approaches were able to provide precise localization, due to the small workspace and high visibility of features. However, the CPER approach demonstrated an improvement in average translation error of roughly one order of magnitude, compared to the other methods. The rotation errors are approximately equal for all of the tested methods, except for 25pt, which also performed the least well in terms of translation error, as well. This is likely because the 25pt threshold only added 3 key-frames to the map, resulting in few points to track against.

Conversely, the CPER approach aggressively added key-frames at the beginning of the sequence in order to rapidly reduce the tracking entropy. Figure 5.3(d) shows that CPER added approximately 4 key-frames within the first two seconds of motion. Although

the decrease in entropy, seen in Figure 5.3(c), is also exhibited by the 60pt and 0.5mt approaches, both achieve the same entropy reduction approximately 3 seconds after CPER. This is because CPER actively selects key-frames in order to reduce tracking entropy, which allows for consistent localization throughout the trajectory.

We further see the CPER approach is not wasteful in adding key-frames. Although 0.5mt and 60pt achieved comparable accuracy and tracker entropy reduction, each approach added 13 and 44 key-frames respectively, while CPER only added 8. Similarly CPER added 74% and 60% fewer map points compared to 60pt and 0.5mt, respectively. This is because our approach only adds new key-frames when required by the tracker, and seeks to improve the points which exist in the map. As a result, fewer multi-keyframes are added, and fewer points are required to maintain suitable tracking integrity. Since the point percent threshold methods add key-frames based on the ratio of measured image features to visible map points, image noise and temporary tracking loss of map points will cause these approaches to rapidly insert key-frames into the map, displayed by the 60pt approach in Figure 5.3(d).

5.3.2 Evaluation of PPFD in an Exploration Task

We identify three important situations where the PPFD approach is effective in key-frame selection. In case 1, we illustrate how PPFD is well suited to exploratory movements when there are few existing points in the map. Case 2 demonstrates how PPFD is able to effectively add new key-frames when the view points are partially occluded, and finally, case 3 shows how PPFD is able to successfully initialize new map points when the camera cluster is undergoing aggressive motions. Results from the three cases are highlighted in Figures 5.5 - 5.7, while the relative entropy score (RES) over the motion sequences are displayed in Figure 5.8. In order to isolate the results of key-frame triggering from key-frame selection, the triggering was performed after a fixed time horizon in the sequence.

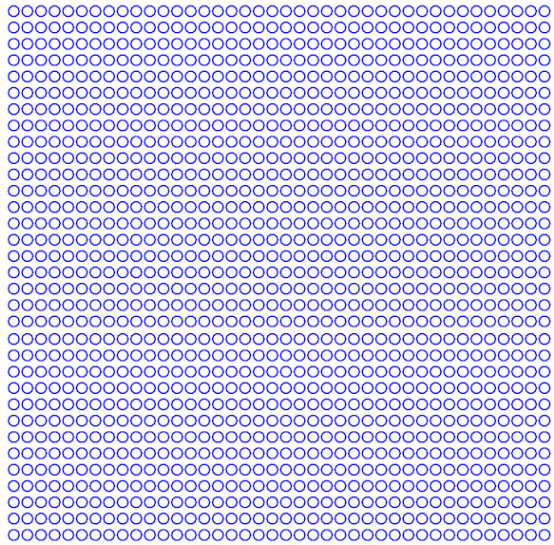
After a new key-frame was requested, the key-frame for insertion was selected from the buffer using the PPFD method.

Exploration: Figure 5.5(a) shows an image with the map points visible immediately after initialization, while Figure 5.5(b) shows the map points after the key-frame was inserted using PPFD. The respective E-PFP and F-PFP are visualized in Figures 5.5(c) and 5.5(d). The E-PFP clearly shows areas in the image frame which are lacking points, and the F-PFP corresponds to the key-frame from the buffer which resulted in the lowest RES, displayed in Figure 5.8(a).

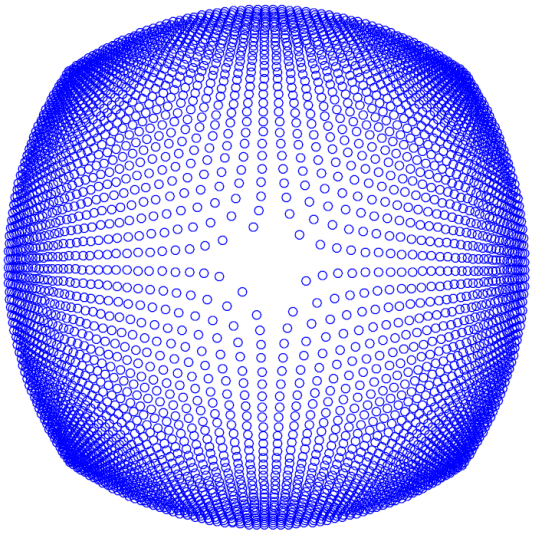
Occlusion: The map points pre and post key-frame insertion are depicted in Figure 5.6(a) and 5.6(c), respectively. There are few map points at initialization, and as shown in Figure 5.6(b), there are few predicted map points due to the occlusion. This behaviour is reflected by the RES in Figure 5.8(b), where the occlusion results in a high RES between key-frames 10 and 85. Finally, once the camera has passed the occlusion, the RES drops sharply at key-frame 90, which is selected as the key-frame for insertion. The respective E-PFP and F-PFP are visualized in Figures 5.6(d) and 5.6(e).

Aggressive Motion: Due to the aggressive movement of the camera, the image exhibits motion blur, as seen in Figure 5.7(a). In this case, the camera was pitched back and forth aggressively, which resulted in a diminished ability to robustly track features. The behaviour is reflected the RES, seen in Figure 5.8(c), where the score exhibits oscillatory behaviour and intermittent regions where there were no feature matches with which to compute the F-PFP score. However, the PPFD method chooses the key-frame from the buffer which is able to best initialize new map points. In this case, key-frame 10, where the image was momentarily sharp due to a direction change in the rotation, was selected for insertion, resulting in the initialization of new map points seen in Figure 5.7(b). The associated E-PFP and F-PFP are also visualized in Figures 5.7(c) and 5.7(d), respectively. Figure 5.7(e) depicts a F-PFP from key-frame 110, which had a relatively high RES. Note

the lower number of features detected on Figure 5.7(e) when compared to Figure 5.7(d).

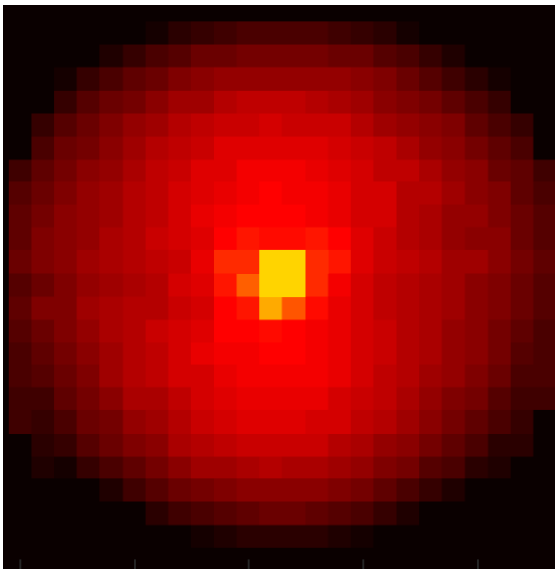


(a) Pinhole



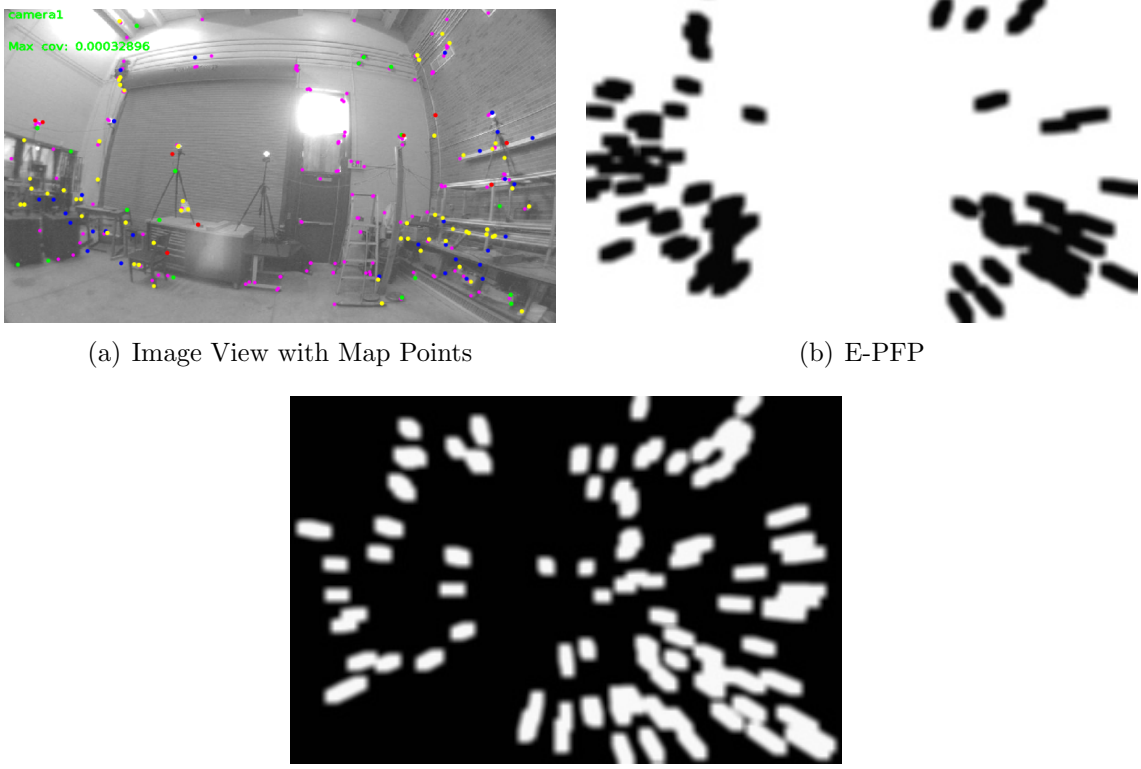
(b) Taylor

(c) Jacobian (pinhole)



(d) Jacobian (Taylor)

Figure 5.1: Comparison of image reprojection sensitivity between pinhole and Taylor camera models, with respect to the x point parameter. The image compression around the edges results in reduced sensitivity of image projection Jacobian in the outer edge areas, as seen in (b), where as the pinhole camera model displays uniform strength in the image reprojection Jacobian, as seen in (a).

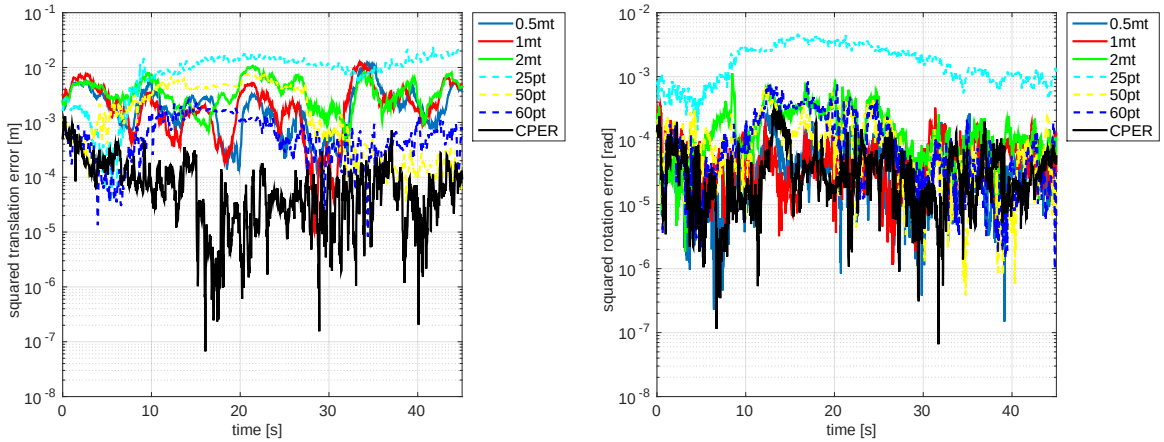


(a) Image View with Map Points

(b) E-PFP

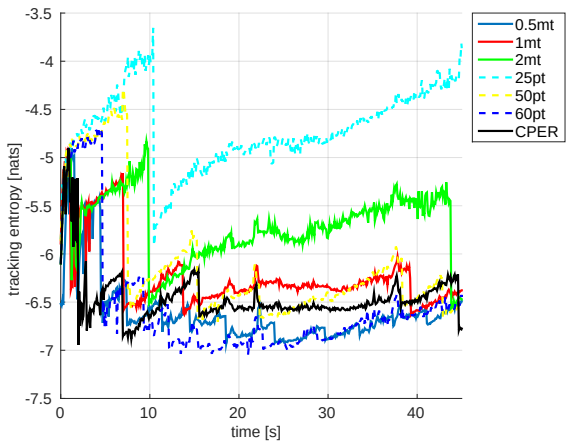
(c) F-PFP

Figure 5.2: Example E-PFP and P-PFP with corresponding image view and map points. (a) shows the camera view. The red, green, and yellow markers denote existing map points, while the magenta markers denote predicted point locations triangulated between current key-frame and existing multi-keyframes in the map. The E-PFP and F-PFP, shown in (b) and (c), respectively, are generated assuming a 1 m/s velocity in the forward direction of the cluster, and for a time horizon of 1 second. Light regions denote areas of high probability, while darker areas indicate low probability.

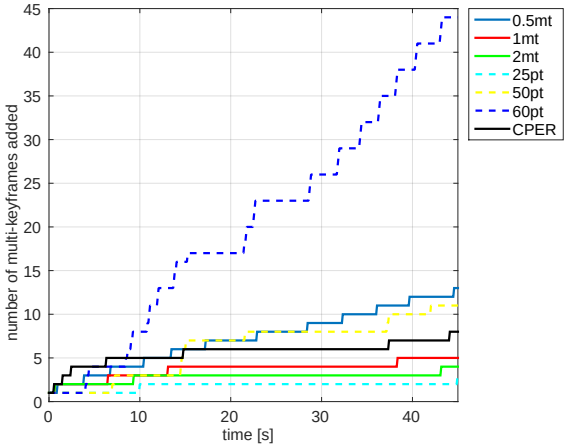


(a) Translational Error

(b) Rotation Error



(c) Tracker Entropy



(d) Keyframe additions

Figure 5.3: (a) and (b) show the translation and rotation errors, respectively, of the camera trajectories compared to ground truth. Note that our proposed CPER approach is able to maintain very precise localization throughout the sequence. Plot (c) illustrates the tracker entropy over time, while (d) shows the key-frame additions over time. Note that the CPER method aggressively adds key-frames in the first few seconds of motion in order to rapidly lower the tracking entropy.

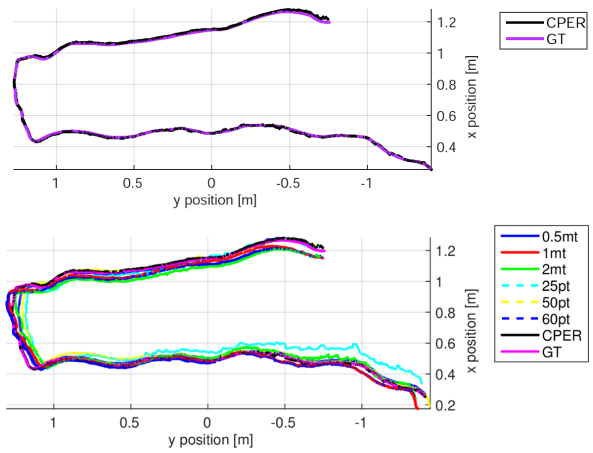
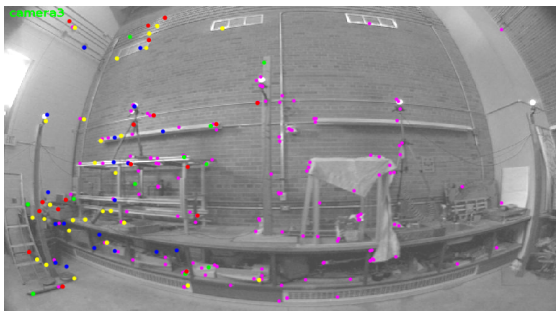
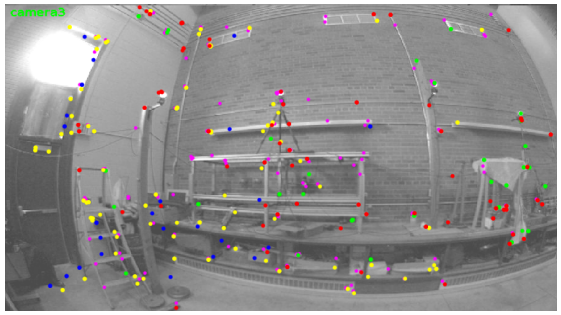


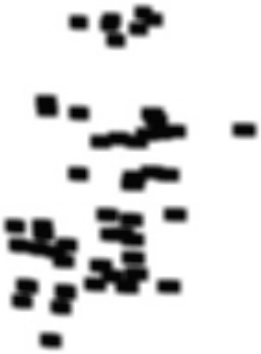
Figure 5.4: The recovered camera cluster motion for the tested methods, with the ground truth overlayed



(a) exp. pre-insertion



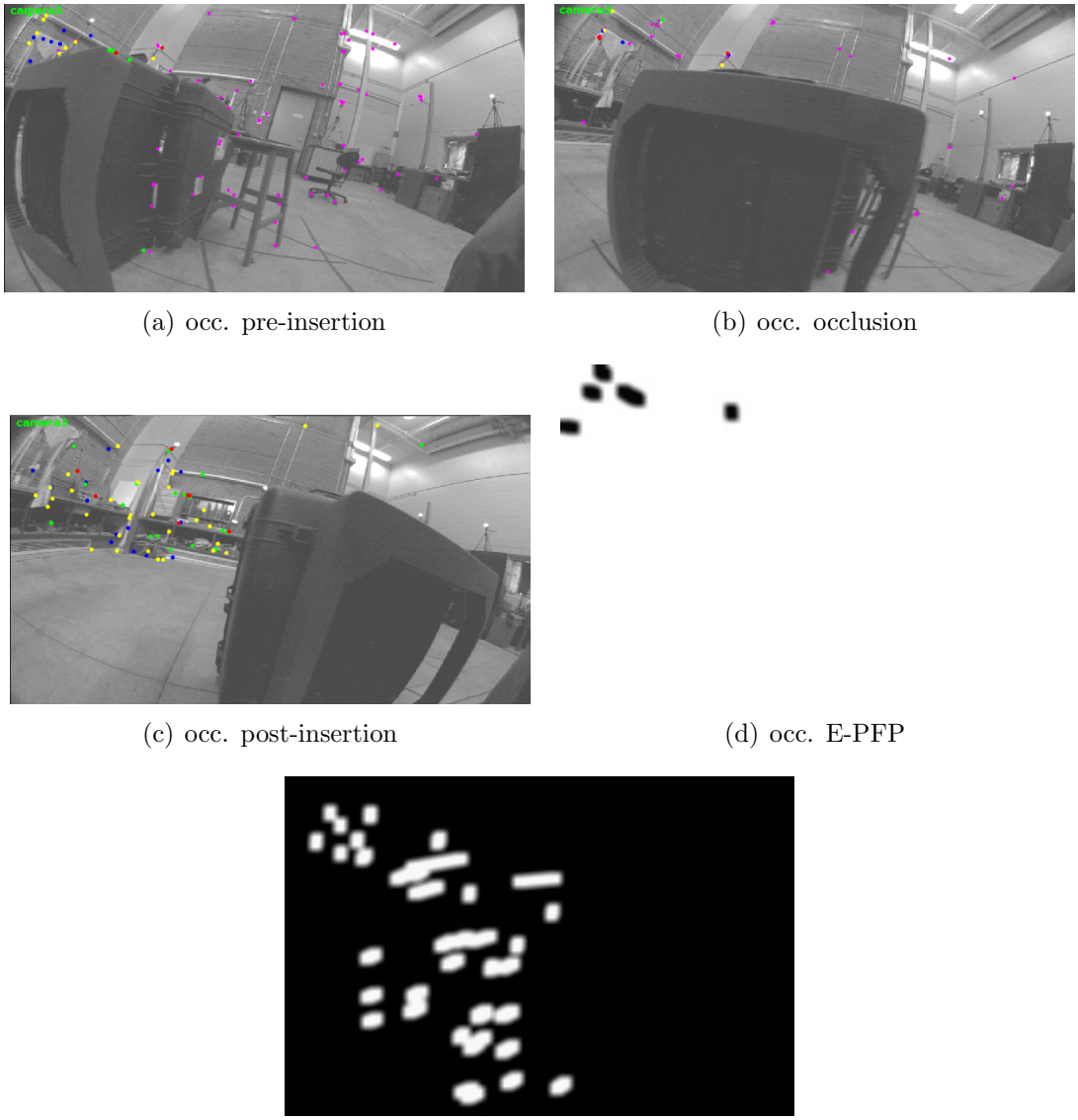
(b) exp. pos-insertion



(c) exp. E-PFP

(d) exp. F-PFP

Figure 5.5: PPFD results for a typical exploration task. Subfigures (a) and (b) show the map points overlayed in the camera images, while (c) and (d) visualize the E-PFPs and F-PFPs used to select the inserted keyframe.



(a) occ. pre-insertion

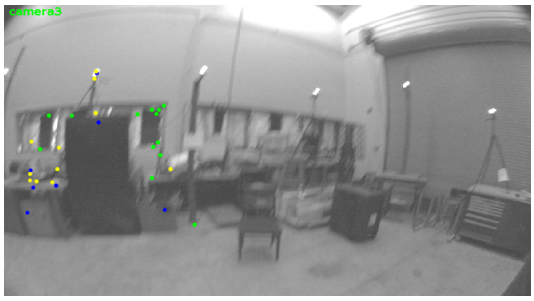
(b) occ. occlusion

(c) occ. post-insertion

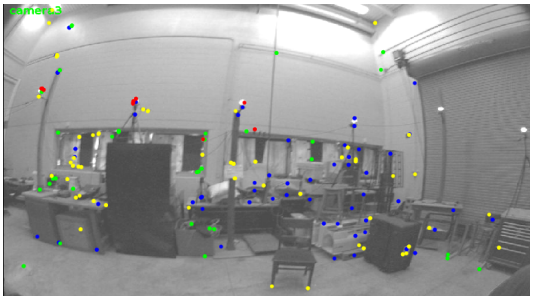
(d) occ. E-PFP

(e) occ. F-PFP

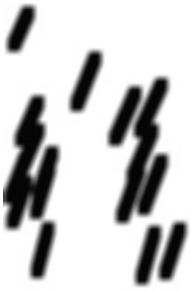
Figure 5.6: PPFD results with occlusion present over a portion of the camera motion. Subfigures (a),(b), and (c) show the map points overlayed in the camera images, while (d) and (e) visualize the E-PFPs and F-PFPs used to select the inserted key-frame. Note the small number of predicted features in (b) due to the occlusion.



(a) agg. pre-insertion



(b) agg. post-insertion

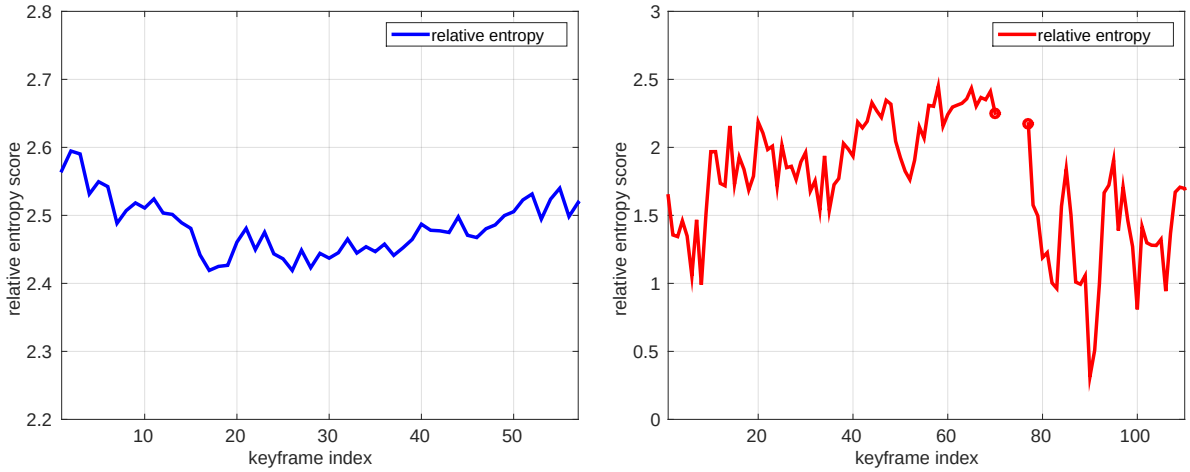


(c) agg. E-PFP

(d) agg. F-PFP

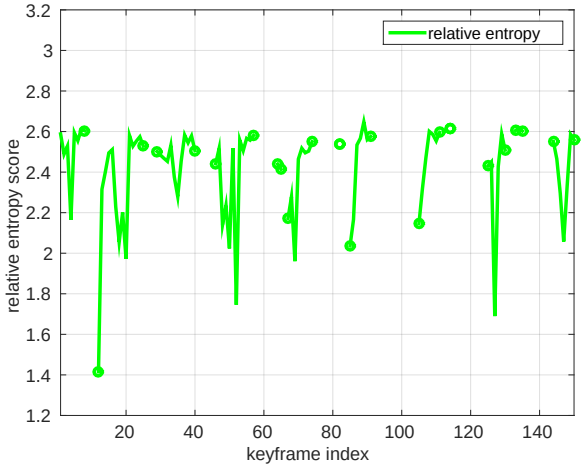
(e) agg. failed F-PFP

Figure 5.7: PPFD results with aggressive motion of the camera cluster. Subfigures (a) and (b) show the map points overlayed in the camera images, while (c) and (d) visualize the E-PFPs and F-PFPs used to select the inserted key-frame, and (e) depicts the F-PFP of a key-frame where few feature matches are found due to motion blur.



(a) relative entropy (exploration)

(b) relative entropy (occlusion)



(c) relative entropy (aggressive motion)

Figure 5.8: Plots (a),(b), and (c) show the relative entropy over the sequence for the forward exploration, occlusion, and aggressive motion cases, respectively. Note that discontinuities in the plots are due to indeterminate relative entropy scores that result when there are no feature matches in the key-frame Such is especially the case in (c), where rapid motion results in many key-frames where points cannot be matched successfully.

Chapter 6

Conclusion

The work presented in this thesis discusses the design and analysis of systems which seek to perform information-theoretic data selection of the incoming camera streams for multi-camera visual calibration and navigation. In this work, informed data selection is performed using active vision to manipulate the viewpoint of the cameras, and by analyzing the information content in the image streams to determine optimal viewpoints for keyframe selection.

Traditional multi-camera systems require a fixed calibration between cameras to provide the solution at the correct scale. The fixed extrinsic calibration of the cluster places many limitations on multi-camera cluster performance, such as increasing the difficulty of in-field calibration, coupling the collected image viewpoints with the vehicle motion and possibly leading to degenerate motions, and precluding the use of actuated cameras, such as UAV gimbals, with state-of-the-art visual navigation systems. This thesis investigates the calibration of dynamic camera clusters, or DCCs, where one or more of the cluster cameras is mounted to an actuated mechanism, such as a gimbal or robotic manipulator. Our novel calibration approach parametrizes the actuated mechanism using the Denavit-Hartenberg convention, then determines the calibration parameters which allow for the

estimation of the time varying extrinsic transformations between the static and dynamic camera frames.

The proposed dynamic-to-static camera transformation chain is parameterized using the DH convention, as described in Chapter 2, which allows us to parameterize the extrinsic calibration between the optical centers of a static and dynamic camera, as a function of the actuated mechanism input, and perform the DCC calibration for any system that can be described as a serial manipulator.

The DCC calibration can be formulated either using a reprojection error, or a poseloop error, as presented in Chapter 3, both of which can perform the estimation of the unknown calibration parameters through successive observations of a fiducial target of known scale. We also presented an extension of the dynamic-to-static calibration case which performs DCC calibration between two dynamic cameras. The proposed calibration approaches were evaluated both in simulation, and using physical hardware. High quality calibrations were achieved using a 5-DOF Fanuc arm based DCC, and a 3-DOF Aeryon gimbal based DCC, which exhibited approximately 0.58 and 2.13 RMSE pixel error, respectively, using a validation data set.

The proposed DCC calibration was analyzed for system degeneracies, in Chapter 3, illustrating an over-parameterization of the system. By investigating the system Jacobian matrix, it was shown that a subset of the columns could be expressed as linear combinations of each other, identifying a rank deficiency in the Jacobian matrix. As a result, we showed that the d_M , a_1, α_1, and d_1 calibration parameters are redundant, a degeneracy occurs when two subsequent joint axes are parallel, and that the special case where there is one joint axis in the DCC presents two additional redundant parameters.

Manual measurement collection for DCC calibration is increasingly onerous as the number of degrees of freedom of the manipulator increase, as it becomes difficult to ensure sufficient measurement excitation through the mechanism's configuration space. In order to

automate the calibration process, Chapter 4 presented two information theoretic methods which select the optimal calibration viewpoints using a next-best-view strategy. The first strategy looked at minimizing the entropy of the calibration parameters, while the second method selected the viewpoints which maximize the mutual information between the joint angle input and calibration parameters. The NBV methods were evaluated using the 5-DOF Fanuc arm based DCC and a 3-DOF Aeryon gimbal DCC, and we demonstrated that both the entropy and mutual information based NBV approaches are able to actively select the viewpoints which result in the lowest parameter uncertainty.

The effective selection of key-frames is also an essential aspect of robust visual navigation algorithms, as it ensures metrically consistent mapping solutions while reducing the computational complexity of the bundle adjustment process. Both the accuracy and computational complexity of visual SLAM solutions can be significantly improved by selectively including only the most informative new measurements at each stage of the algorithm. Existing approaches generally insert key-frames based on point triangulation baselines, co-visibility of features, or heuristics which insert key-frames whenever the number of currently tracked points falls below a user defined threshold. Overall these existing key-frame insertion strategy remain ad-hoc and requires manual tuning.

To that end, this thesis also presented two novel approaches for key-frame selection, cumulative point entropy reduction (CPER) and predicted pixel flow discrepancy (PPFD), which systematically evaluate candidate key-frames and selects them based on the expected improvement of the map. The approaches are complementary, as CPER improves the existing map, and PPFD predicts and improves the future map. Our experimental results demonstrated that the CPER approach allows for accurate location using fewer key-frames compared to ad-hoc insertion methods, and that the PPFD approach is well suited to exploration tasks and is able to achieve good performance even in the presence of occlusion and aggressive motion.

In summary, the main contributions of this thesis are listed as follows:

- The novel formulation of the calibration process for a dynamic camera cluster, and using the Denavit-Hartenberg convention to parameterize the extrinsic calibration between the optical centers of a static and dynamic camera, as a function of the actuated mechanism input.
- An analytical degeneracy study leading to the identification of redundant parameters of the DCC calibration system.
- Two novel information theoretic next-best-view measurement approaches, which seek to select actuator configurations which directly reduce the uncertainty of the calibration parameters. The approaches investigate two formulations: one which seeks to reduce the entropy of the estimated parameter covariance matrix, and another which aims to optimally find the actuator configurations which maximize the mutual information between the extrinsic calibration parameters and mechanism joint angles.
- Two novel approaches for key-frame selection which systematically determine the best key-frames for insertion into a point-feature based map. The first approach, CPER, inserts key-frames based on the cumulative point entropy reduction in the existing map, while the second approach, PPFD, uses the predicted point flow discrepancy to select key-frames which best initialize new features for the camera to track against in the future.

6.1 Future Extensions

Given the versatility of the discussed methods in this work, a wide range of future extensions can be considered. In this section, the most promising of them are summarized.

- A calibrated DCC can be mounted to a vehicle, and used for localization or SLAM. It would be interesting to look at viewpoint selection strategies, based on the ideas presented in this thesis, which point the cameras in order to maximize localization quality.

- Active viewpoint selection using a calibrated DCC could also be formulated and applied to a wide range of tasks outside of SLAM and localization. For example, the calibrated DCC can be used for inspection, or a viewpoint policy could be formulated in order to reconfigure the cluster to best perform dense reconstruction for obstacle avoidance.

- As seen in this work, the quality of the DCC calibration is directly impacted by the accuracy of the joint angle encoders. An *encoderless* approach, which does not require any joint angle feedback, but instead estimates the joint angle configurations as part of the calibration state, facilitates DCC calibration on systems which either posses poor quality encoders, or no encoders at all. Although such a system has been formulated and tested on a 2-DOF gimbal [87], the degeneracy analysis of the encoderless system is yet to be investigated. This degeneracy analysis would be an interesting problem to pursue, as the inclusion of the joint angle states in the estimation makes this a rich problem.

- The next-best-view formulations can be extended to include the intrinsic lens parameters. The NBV approaches can further be modified in order to establish repeatable trajectories for more generic calibrations, such as IMU to Camera, LIDAR to Camera, etc. These repeatable trajectories would be valuable in a manufacturing setting, where the approach could be used to automatically calibrate large volumes of devices.

- Remove the fiducial target for the DCC calibration, and attempt to perform the calibration using a pre-built map, natural features in the environment, or additional

sensing that provides scale information, such as LIDAR, GPS, or IMU.

- An interesting extension to the key-frame selection strategies presented in this thesis would be to apply them to other variants of SLAM and localization which use other sensors and map representations, such as point-cloud, surfel, or TSDF.

References

[1] A. Kim and R. Eustice, "Real-time visual SLAM for autonomous underwater hull inspection using visual saliency," *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 719–733, 2013.

[2] L. Doitsidis, A. Renzaglia, S. Weiss, E. Kosmatopoulos, D. Scaramuzza, and R. Siegwart, "3D surveillance coverage using maps extracted by a monocular SLAM algorithm," in *International Conference on Intelligent Robots and Systems (IROS)*, Sept 2011, pp. 1661–1667.

[3] T. Howard, A. Morfopoulos, J. Morrison, Y. Kuwata, C. Villalpando, L. Matthies, and M. McHenry, "Enabling continuous planetary rover navigation through FPGA stereo and visual odometry," in *IEEE Aerospace Conference*, Big Sky, MT, 2012, pp. 1–9.

[4] A. Das, D. Kumar, A. El Bably, and S. L. Waslander, "Taming the north: Multi-camera parallel tracking and mapping in snow-laden environments," in *Field and Service Robotics (FSR)*, June 2015.

[5] M. J. Tribou, D. W. Wang, and S. L. Waslander, "Degenerate motions in multicamera cluster SLAM with non-overlapping fields of view," *Image and Vision Computing*, vol. 50, pp. 27–41, 2016.

[6] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision.* Cambridge University Press, 2003.

[7] I. Cvisic and I. Petrovic, "Stereo odometry based on careful feature selection and tracking," in *European Conference on Mobile Robots (ECMR)*, Lincoln, UK, Sept 2015, pp. 1–6.

[8] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Vision-based state estimation for autonomous rotorcraft mavs in complex environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2013, pp. 1758–1764.

[9] A. Tribou, Michael J. and. Harmat, D. Wang, I. Sharf, and S. L. Waslander, "Tracking and mapping with non-overlapping fields of view," *International Journal of Robotics Research*, vol. 34, no. 12, pp. 1480–1500, December 2015.

[10] A. Harmat, M. Trentini, and I. Sharf, "Multi-camera tracking and mapping for unmanned aerial vehicles in unstructured environments," *Journal of Intelligent & Robotic Systems*, vol. 78, no. 2, pp. 291–317, 2014.

[11] M. J. Tribou, A. Harmat, D. Wang, I. Sharf, and S. L. Waslander, "Multi-camera parallel tracking and mapping with non-overlapping fields of view," *International Journal of Robotics Research*, pp. 1–43, December 2014.

[12] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-SLAM: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[13] R. Mur-Artal and J. D. Tardós, "Orb-SLAM2: An open-source slam system for monocular, stereo, and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[14] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Computer Vision (ECCV)*, Sept. 2014.

[15] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

[16] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 15–22.

[17] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *2007 IEEE international conference on Robotics and Automation (ICRA)*. IEEE, 2007, pp. 3565–3572.

[18] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[19] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, Nov 2013, pp. 1280–1286.

[20] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *The International Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, 2011.

[21] C. I. Connolly, "The determination of next best views," in *IEEE International Conference on Robotics and Automation (ICRA)*, 1985.

[22] J. Levinson and S. Thrun, "Unsupervised calibration for multi-beam lasers," in *International Symposium on Experimental Robotics*, Delhi, India, 2014, pp. 179–193.

[23] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, "Automatic targetless extrinsic calibration of a 3D lidar and camera by maximizing mutual information." in *AAAI National Conference on Artificial Intelligence*, Toronto, Canada, July 2012, pp. 2053–2059.

[24] P. Lbraly, E. Royer, O. Ait-Aider, C. Deymier, and M. Dhome, "Fast calibration of embedded non-overlapping cameras," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 2011, pp. 221–227.

[25] B. Li, L. Heng, K. Koser, and M. Pollefeys, "A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, Nov 2013, pp. 1301–1307.

[26] L. Heng, M. Burki, G. H. Lee, P. Furgale, R. Siegwart, and M. Pollefeys, "Infrastructure-based calibration of a multi-camera rig," in *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, 2014, pp. 4912–4919.

[27] K. Daniilidis, "Hand-eye calibration using dual quaternions," *The International Journal of Robotics Research*, vol. 18, no. 3, pp. 286–298, 1999.

[28] A. Censi, A. Franchi, L. Marchionni, and G. Oriolo, "Simultaneous calibration of odometry and sensor parameters for mobile robots," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 475–492, 2013.

[29] M.-S. Kim, H.-S. Yoo, S.-W. Cho, H.-S. Chang, and G. Spur, "A new calibration method," *CIRP Annals-Manufacturing Technology*, vol. 39, no. 1, pp. 421–424, 1990.

[30] D. Whitney, C. Lozinski, and J. M. Rourke, "Industrial robot forward calibration method and results," *Journal of dynamic systems, measurement, and control*, vol. 108, no. 1, pp. 1–8, 1986.

[31] D. C.-C. Lu, "Kinematic calibration of serial manipulators using relative measurements," Master's thesis, Ottawa-Carleton Institute for Mechanical and Aerospace Engineering Department, Jan 2014.

[32] C. S. Gatla, R. Lumia, J. Wood, and G. Starr, "Calibrating pan-tilt cameras in robot hand-eye systems using a single point," in *IEEE International Conference on Robotics and Automation (ICRA)*, Roma, Italy, April 2007, pp. 3186–3191.

[33] V. Pradeep, K. Konolige, and E. Berger, "Calibrating a multi-arm multi-sensor robot: A bundle adjustment approach," in *International Symposium on Experimental Robotics (ISER)*, 12/2010 2010.

[34] R. Hermann and A. Krener, "Nonlinear controllability and observability," *IEEE Transactions on Automatic Control*, vol. 22, no. 5, pp. 728–740, 1977.

[35] F. M. Mirzaei and S. I. Roumeliotis, "A kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1143–1156, 2008.

[36] J. H. Kim, M. J. Chung, and B. T. Choi, "Recursive estimation of motion and a scene model with a two-camera system of divergent view," *Pattern Recognition*, vol. 43, no. 6, pp. 2265–2280, 2010.

[37] B. Clipp, J. H. Kim, J. M. Frahm, M. Pollefeys, and R. Hartley, "Robust 6DOF motion estimation for non-overlapping, multi-camera systems," in *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV)*, Jan. 2008, pp. 1–8.

[38] J. S. Kim and T. Kanade, "Degeneracy of the linear seventeen-point algorithm for generalized essential matrix," *Journal of Mathematical Imaging and Vision*, vol. 37, no. 1, pp. 40–48, May 2010.

[39] G. H. Lee, F. Fraundorfer, and M. Pollefeys, "Motion estimation for self-driving cars with a generalized camera," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2013.

[40] H. Stewenius and K. Astrom, "Structure and motion problems for multiple rigidly moving cameras," *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 252–263, 2004.

[41] A. J. Krener and K. Ide, "Measures of unobservability," in *Conference on Decision and Control (CDC.* IEEE, 2009, pp. 6401–6406.

[42] K. Hausman, J. Preiss, G. S. Sukhatme, and S. Weiss, "Observability-aware trajectory optimization for self-calibration with application to uavs," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1770–1777, 2017.

[43] R. Minguez and A. J. Conejo, "State estimation sensitivity analysis," *IEEE Transactions on Power Systems*, vol. 22, no. 3, pp. 1080–1091, 2007.

[44] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola, *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008.

[45] Y.-J. Huoh, *Sensitivity analysis of stochastic simulators with information theory*. University of California, Berkeley, 2013.

[46] W. Burgard, D. Fox, and S. Thrun, "Active mobile robot localization," 1997.

[47] A. J. Davison and D. W. Murray, "Mobile robot localisation using active vision," in *European Conference on Computer Vision*. Springer, 1998, pp. 809–825.

[48] H. J. S. Feder, J. J. Leonard, and C. M. Smith, "Adaptive mobile robot navigation and mapping," *The International Journal of Robotics Research*, vol. 18, no. 7, pp. 650–668, 1999.

[49] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte, "Information based adaptive robotic exploration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2002, pp. 540–545.

[50] A. Handa, M. Chli, H. Strasdat, and A. Davison, "Scalable active matching," in *Computer Vision and Pattern Recognition (CVPR)*, June 2010, pp. 1546–1553.

[51] M. Chli and A. Davison, "Automatically and efficiently inferring the hierarchical structure of visual maps," in *International Conference on Robotics and Automation (ICRA)*, May 2009, pp. 387–394.

[52] M. Chli and A. J. Davison, "Active matching," in *European conference on computer vision*. Springer, 2008, pp. 72–85.

[53] M. Chli, "Applying information theory to efficient slam," Ph.D. dissertation, Department of Computing, Imperial College London, 2010.

[54] R. Bajcsy, "Active perception," in *Proceedings of IEEE*, vol. 76, no. 8, 1988.

[55] S. Frintrop and P. Jensfelt, "Attentional landmarks and active gaze control for visual SLAM," *IEEE Transactions on Robotics, special Issue on Visual SLAM*, vol. 24, no. 5, Oct. 2008.

[56] A. Richardson, J. Strom, and E. Olson, "AprilCal: Assisted and repeatable camera calibration," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013.

[57] Y. Kuniyoshi, N. Kita, T. Suehiro, and S. Rougeaux, "Active stereo vision system with foveated wide angle lenses," in *Asian Conference on computer vision*. Springer, 1995, pp. 191–200.

[58] K. Kuhnlenz, G. Lidoris, D. Wollherr, and M. Buss, "On foveated gaze control and combined gaze and locomotion planning," in *Humanoid Robots: New Developments*. InTech, 2007.

[59] T. Manderson, F. Shkurti, and G. Dudek, "Texture-aware SLAM using stereo imagery and inertial information," in *Conference on Computer and Robot Vision (CRV)*. IEEE, 2016, pp. 456–463.

[60] R. B. Gomes, B. M. de Carvalho, and L. M. G. Gonçalves, "Visual attention guided features selection with foveated images," *Neurocomputing*, vol. 120, pp. 34–44, 2013.

[61] Y. Sun and R. Fisher, "Object-based visual attention for computer vision," *Artificial intelligence*, vol. 146, no. 1, pp. 77–123, 2003.

[62] S. Wenhardt, B. Deutsch, J. Hornegger, H. Niemann, and J. Denzler, "An Information Theoretic Approach for Next Best View Planning in 3-D Reconstruction," in *The 18th International Conference on Pattern Recognition*, vol. 1, 2006, pp. 103–106.

[63] E. Dunn and J.-M. Frahm, "Next best view planning for active model improvement." in *British Machine Vision Conference (BMVC)*, 2009, pp. 1–11.

[64] S. Foix, G. Alenyà, and C. Torras, "Towards plant monitoring through next best view," in *International Conference of Catalan Association for Artificial Intelligence (CCIA)*. Citeseer, 2011.

[65] C. Potthast and G. S. Sukhatme, "A probabilistic framework for next best view estimation in a cluttered environment." *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 148–164, 2014.

[66] Y.-J. C. Jwu-Sheng Hu, "Automatic calibration of hand-eye-workspace and camera using hand-mounted line laser," *IEEE/ASME Transactions on Mechatronics*, vol. 18, pp. 1778–1786, 2013.

[67] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3d robotics hand/eye calibration," in *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, 1989, pp. 345–358.

[68] S. D. Ma, "A self-calibration technique for active vision systems," in *IEEE Transactions on Robotics and Automation*, vol. 12, 1996, pp. 114–120.

[69] Y. F. Z. Chichyang Chen, "A new robotic hand/eye calibration method by active viewing of a checkerboard pattern," in *IEEE Transactions on Robotics and Automation*. IEEE, 1993.

[70] K. Nickels, E. Huber, and M. Dicicco, "Hand-eye calibration using active vision," in *IEEE Aerospace Conference*, 2007.

[71] D. Bennett, D. Geiger, and J. Hollerbach, "Autonomous robot calibration for hand-eye coordination," *International Journal of Robotics Research*, vol. 10, no. 5, pp. 550–559, 10 1991.

[72] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, Nara, Japan, November 2007, pp. 225–234.

[73] J. Stalbaum and J.-B. Song, "Keyframe and inlier selection for visual SLAM," in *Ubiquitous Robots and Ambient Intelligence (URAI)*, Oct 2013, pp. 391–396.

[74] S. Leutenegger, P. T. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart, "Keyframe-based visual-inertial SLAM using nonlinear optimization," in *Robotics: Science and Systems (RSS)*, June 2013.

[75] D. Gallup, J.-M. Frahm, P. Mordohai, and M. Pollefeys, "Variable baseline/resolution stereo," in *Computer Vision and Pattern Recognition (CPVR)*. IEEE, June 2008, pp. 1–8.

[76] Z. Dong, G. Zhang, J. Jia, and H. Bao, "Keyframe-based real-time camera tracking," in *International Conference on Computer Vision (ICCV)*. IEEE, Sept. 2009, pp. 1538–1545.

[77] M. Bloesch, H. Sommer, T. Laidlow, M. Burri, G. Nuetzi, P. Fankhauser, D. Bellicoso, C. Gehring, S. Leutenegger, M. Hutter *et al.*, "A primer on the differential calculus of 3d orientations," *arXiv preprint arXiv:1606.05285*, 2016.

[78] T. D. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2017.

[79] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder, "Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds," *Information Fusion*, vol. 14, no. 1, pp. 57–77, 2013.

[80] M. W. Spong and M. Vidyasagar, *Robot dynamics and control*. John Wiley & Sons, 2008.

[81] R. S. Hartenberg and J. Denavit, *Kinematic Synthesis of Linkages*. McGraw-Hill, 1964.

[82] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A flexible technique for accurate omnidirectional camera calibration and structure from motion," in *IEEE International Conference on Computer Vision Systems (ICVS)*. New York, NY: IEEE, Jan. 2006, pp. 45–45.

[83] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1335–1340, 2006.

[84] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 2011.

[85] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY: Wiley-Interscience, 2006.

[86] D. J. MacKay, *Information theory, inference and learning algorithms*. Cambridge University Press, 2003.

[87] C. Choi, J. Rebello, L. Koppel, P. Ganti, A. Das, and S. L. Waslander, "Encoderless gimbal calibration of dynamic multi-camera clusters," in *International Conference on Robotics and Automation (ICRA)*, May 2018.

[88] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 2011, pp. 3400–3407.

[89] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPNP: An accurate O(n) solution to the PNP problem," *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009.

[90] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, 1990.

[91] H. Goldstein, *Classical mechanics*. Pearson Education, 2011.

[92] S. Thompson, *Sampling*, ser. CourseSmart. Wiley, 2012.

[93] D. Simon, *Optimal state estimation: Kalman, H_∞ and nonlinear approaches.* John Wiley and Sons, 2006.

[94] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3D reconstruction in real-time," in *Intelligent Vehicles Symposium (IV)*, June 2011.

[95] B. Kitt, A. Geiger, and H. Lategahn, "Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme," in *Intelligent Vehicles Symposium (IV)*, June 2010.

APPENDICES

Appendix A

Skew-Symmetric Matrix Identities

The degeneracy analysis performed in Chapter 3 requires the manipulation of vectors expressed in skew-symmetric form. This appendix will outline some important identities related to skew-symmetric matrices, which are described in greater detail in [77].

Suppose we have a vector $v \in \mathbb{R}^3$. A skew-symmetric (also called an *anti-symmetric*) matrix satisfies

$$([v]^\wedge)^T = -[v]^\wedge. \tag{A.1}$$

Additionally, given an orthogonal matrix $\mathbf{R}$, it can be shown that the following identity holds

$$[\mathbf{R}v]^\wedge = \mathbf{R}[v]^\wedge\mathbf{R}^T. \tag{A.2}$$

Finally, though simple manipulation, Equation (A.2) can be expressed as

$$\mathbf{R}[v]^\wedge = [\mathbf{R}v]^\wedge\mathbf{R} \tag{A.3}$$

Appendix B

$\mathbb{SE}(3)$ Composition and Box-minus Jacobian Derivation

The degeneracy analysis performed in Chapter 3 makes extensive use of the transformation matrix composition and box-minus functions, expressed in Equations (3.17) and (3.20), respectively. In this appendix, we shall derive these Jacobian matrices, primarily using the process described in [77].

In order to derive the composition and box-minus Jacobian matrices for elements in $\mathbb{SE}(3)$, we must first derive some important Jacobian quantities relating to elements in $\mathbb{SO}(3)$.

B.1 Important $\mathbb{SO}(3)$ Jacobians

Suppose we have two rotation matrices, $\mathbf{R}^{\mathsf{b:c}} \in \mathbb{SO}(3)$ and $\mathbf{R}^{\mathsf{a:b}} \in \mathbb{SO}(3)$, with their resulting composition

$$\mathbf{R}^{\mathsf{a:c}} = \mathbf{R}^{\mathsf{a:b}}\mathbf{R}^{\mathsf{b:c}}. \tag{B.1}$$

As derived in [77], the derivatives of the composition function from Equation (B.1), are given as

$$\frac{\partial \mathbf{R}^{\mathsf{a:b}}\mathbf{R}^{\mathsf{b:c}}}{\partial \mathbf{R}^{\mathsf{a:b}}} = I \tag{B.2}$$

$$\frac{\partial \mathbf{R}^{\mathsf{a:b}}\mathbf{R}^{\mathsf{b:c}}}{\partial \mathbf{R}^{\mathsf{b:c}}} = \mathbf{R}^{\mathsf{a:b}}, \tag{B.3}$$

and the derivatives of the inverse and log functions are given as

$$\frac{\partial (\mathbf{R}^{\mathsf{a:b}})^{-1}}{\partial \mathbf{R}^{\mathsf{a:b}}} = -(\mathbf{R}^{\mathsf{a:b}})^{T} \tag{B.4}$$

and

$$\frac{\partial \log(\mathbf{R}^{\mathsf{a:b}})}{\partial \mathbf{R}^{\mathsf{a:b}}} = [\Gamma(\log(\mathbf{R}^{\mathsf{a:b}}))]^{-1}, \tag{B.5}$$

respectively. Here, $\Gamma \in \mathbb{R}^{3\times 3}$ is the Jacobian of the exponential map function, and is given by

$$\Gamma(\phi^{\mathsf{a:b}}) = I + \frac{(1-\cos(\|\phi^{\mathsf{a:b}}\|))[\phi^{\mathsf{a:b}}]^{\wedge}}{\|\phi^{\mathsf{a:b}}\|^2} + \frac{(\|\phi^{\mathsf{a:b}}\| - \sin(\|\phi^{\mathsf{a:b}}\|))[[\phi^{\mathsf{a:b}}]^{\wedge}]^2}{\|\phi^{\mathsf{a:b}}\|^3} \tag{B.6}$$

$$\Gamma(\phi^{\mathsf{a:b}}) \approx I + \frac{1}{2}[\phi^{\mathsf{a:b}}]^{\wedge}, \quad (\|\phi^{\mathsf{a:b}}\| \approx 0). \tag{B.7}$$

Finally, the derivative of the co-ordinate mapping through a rotation matrix, is given as

$$\frac{\partial (\mathbf{R}^{\mathsf{a:b}}\,{}^{\mathsf{b}}t^{\mathsf{b:c}})}{\partial \mathbf{R}^{\mathsf{a:b}}} = -[\mathbf{R}^{\mathsf{a:b}}\,{}^{\mathsf{b}}t^{\mathsf{b:c}}]^{\wedge}. \tag{B.8}$$

B.2 Box-minus Jacobian for $\mathbb{SO}(3)$

Suppose we have two rotation matrices, $\mathbf{R}^{\mathsf{a:c}} \in \mathbb{SO}(3)$ and $\mathbf{R}^{\mathsf{b:c}} \in \mathbb{SO}(3)$, with their resulting box-minus difference

$$\phi^{\mathsf{a:b}} = \mathbf{R}^{\mathsf{a:c}} \boxminus \mathbf{R}^{\mathsf{b:c}} \tag{B.9}$$

$$= \log(\mathbf{R}^{\mathsf{a:c}}(\mathbf{R}^{\mathsf{b:c}})^{-1}) \tag{B.10}$$

$$\tag{B.11}$$

where $\phi^{\mathsf{a:b}} \in \mathfrak{so}(3)$ is a member of the Lie Algebra of $\mathbb{SO}(3)$. Using chain rule, the derivative of the expression from Equation (B.9) with respect to the left rotation, $\mathbf{R}^{\mathsf{a:c}}$, is given as

$$\frac{\partial(\mathbf{R}^{\mathsf{a:c}} \boxminus \mathbf{R}^{\mathsf{b:c}})}{\partial \mathbf{R}^{\mathsf{a:c}}} = \frac{\partial \log(\mathbf{R}^{\mathsf{a:c}}(\mathbf{R}^{\mathsf{b:c}})^{\text{-}1})}{\partial(\mathbf{R}^{\mathsf{a:c}}(\mathbf{R}^{\mathsf{b:c}})^{-1})} \frac{\partial(\mathbf{R}^{\mathsf{a:c}}(\mathbf{R}^{\mathsf{b:c}})^{-1})}{\partial \mathbf{R}^{\mathsf{a:c}}}. \tag{B.12}$$

By Equation (B.2),

$$\frac{\partial(\mathbf{R}^{\mathsf{a:c}}(\mathbf{R}^{\mathsf{b:c}})^{-1})}{\partial \mathbf{R}^{\mathsf{a:c}}} = I, \tag{B.13}$$

and

$$\frac{\partial \log(\mathbf{R}^{\mathsf{a:c}}(\mathbf{R}^{\mathsf{b:c}})^{\text{-}1})}{\partial(\mathbf{R}^{\mathsf{a:c}}(\mathbf{R}^{\mathsf{b:c}})^{-1})} \tag{B.14}$$

is evaluated using Equation (B.5). Noting that $\mathbf{R}^{\mathsf{a:c}}(\mathbf{R}^{\mathsf{b:c}})^{-1} = \mathbf{R}^{\mathsf{a:b}}$, the derivative of Equation (B.12) can be simplified as

$$\frac{\partial(\mathbf{R}^{\mathsf{a:c}} \boxminus \mathbf{R}^{\mathsf{b:c}})}{\partial \mathbf{R}^{\mathsf{a:c}}} = \frac{\partial \log(\mathbf{R}^{\mathsf{a:b}})}{\partial \mathbf{R}^{\mathsf{a:b}}} \tag{B.15}$$

$$= [\Gamma(\log(\mathbf{R}^{\mathsf{a:b}}))]^{-1}. \tag{B.16}$$

Similarly, the derivative of the expression from Equation (B.9) with respect to the right

rotation, $\mathbf{R}^{\text{b:c}}$, is given as

$$\frac{\partial(\mathbf{R}^{\text{a:c}} \boxminus \mathbf{R}^{\text{b:c}})}{\partial \mathbf{R}^{\text{b:c}}} = \frac{\partial \log(\mathbf{R}^{\text{a:c}}(\mathbf{R}^{\text{b:c}})^{\text{-1}})}{\partial(\mathbf{R}^{\text{a:c}}(\mathbf{R}^{\text{b:c}})^{-1})} \frac{\partial(\mathbf{R}^{\text{a:c}}(\mathbf{R}^{\text{b:c}})^{-1})}{\partial(\mathbf{R}^{\text{b:c}})^{-1}} \frac{\partial(\mathbf{R}^{\text{b:c}})^{-1}}{\partial \mathbf{R}^{\text{b:c}}}. \tag{B.17}$$

By Equation (B.3),

$$\frac{\partial(\mathbf{R}^{\text{a:c}}(\mathbf{R}^{\text{b:c}})^{-1})}{\partial(\mathbf{R}^{\text{b:c}})^{-1}} = \mathbf{R}^{\text{a:c}}, \tag{B.18}$$

and by Equation (B.4)

$$\frac{\partial(\mathbf{R}^{\text{b:c}})^{-1}}{\partial \mathbf{R}^{\text{b:c}}} = -(\mathbf{R}^{\text{b:c}})^{T}. \tag{B.19}$$

Finally, the derivative of Equation (B.17) can be simplified as

$$\frac{\partial(\mathbf{R}^{\text{a:c}} \boxminus \mathbf{R}^{\text{b:c}})}{\partial \mathbf{R}^{\text{b:c}}} = -[\Gamma(\log(\mathbf{R}^{\text{a:b}}))]^{-1} \mathbf{R}^{\text{a:c}}(\mathbf{R}^{\text{b:c}})^{T} \tag{B.20}$$

$$= \mathcal{J}_{\boxminus}^{\text{b:c}}, \tag{B.21}$$

where $\mathcal{J}_{\boxminus}^{\text{b:c}}$ is the shorthand expression for Equation (B.20), as it is used quite frequently in the degeneracy analysis conducted in Chapter 3.

B.3 Composition Jacobian for $\mathbb{SE}(3)$

Suppose we have two Transformation matrices,

$$\mathbf{T}^{\text{a:b}} = \left[\begin{array}{c|c} \mathbf{R}^{\text{a:b}} & {}^{\text{a}}t^{\text{a:b}} \\ \hline 0 & 1 \end{array}\right], \tag{B.22}$$

$$\mathbf{T}^{\text{b:c}} = \left[\begin{array}{c|c} \mathbf{R}^{\text{b:c}} & {}^{\text{b}}t^{\text{b:c}} \\ \hline 0 & 1 \end{array}\right], \tag{B.23}$$

and their resulting composition

$$\mathbf{T}^{\mathrm{a:b}}\mathbf{T}^{\mathrm{b:c}} = \left[\begin{array}{c|c} \mathbf{R}^{\mathrm{a:b}} & {}^{\mathrm{a}}t^{\mathrm{a:b}} \\ \hline 0 & 1 \end{array}\right] \left[\begin{array}{c|c} \mathbf{R}^{\mathrm{b:c}} & {}^{\mathrm{b}}t^{\mathrm{b:c}} \\ \hline 0 & 1 \end{array}\right] \tag{B.24}$$

$$= \left[\begin{array}{c|c} \mathbf{R}^{\mathrm{a:b}}\mathbf{R}^{\mathrm{b:c}} & \mathbf{R}^{\mathrm{a:b}}\,{}^{\mathrm{b}}t^{\mathrm{b:c}} + {}^{\mathrm{a}}t^{\mathrm{a:b}} \\ \hline 0 & 1 \end{array}\right]. \tag{B.25}$$

As described in Chapter 2, in this work, we treat perturbations to the rotation and translation components of a transformation separately. To that end, the Jacobian of the composition from Equation (B.24) with respect to the left transformation,$\mathbf{T}^{\mathrm{a:b}}$,has the form

$$\frac{\partial \mathbf{T}^{\mathrm{a:b}}\mathbf{T}^{\mathrm{b:c}}}{\partial \mathbf{T}^{\mathrm{a:b}}} = \left[\begin{array}{c|c} \frac{\partial(\mathbf{R}^{\mathrm{a:b}}\mathbf{R}^{\mathrm{b:c}})}{\partial \mathbf{R}^{\mathrm{a:b}}} & \frac{\partial(\mathbf{R}^{\mathrm{a:b}}\mathbf{R}^{\mathrm{b:c}})}{\partial\, {}^{\mathrm{a}}t^{\mathrm{a:b}}} \\ \hline \frac{\partial(\mathbf{R}^{\mathrm{a:b}}\,{}^{\mathrm{b}}t^{\mathrm{b:c}} + {}^{\mathrm{a}}t^{\mathrm{a:b}})}{\partial \mathbf{R}^{\mathrm{a:b}}} & \frac{\partial(\mathbf{R}^{\mathrm{a:b}}\,{}^{\mathrm{b}}t^{\mathrm{b:c}} + {}^{\mathrm{a}}t^{\mathrm{a:b}})}{\partial\, {}^{\mathrm{a}}t^{\mathrm{a:b}}} \end{array}\right], \tag{B.26}$$

where by Equation (B.2),

$$\frac{\partial(\mathbf{R}^{\mathrm{a:b}}\mathbf{R}^{\mathrm{b:c}})}{\partial \mathbf{R}^{\mathrm{a:b}}} = I. \tag{B.27}$$

By inspection, it is clear that

$$\frac{\partial(\mathbf{R}^{\mathrm{a:b}}\mathbf{R}^{\mathrm{b:c}})}{\partial\, {}^{\mathrm{a}}t^{\mathrm{a:b}}} = 0 \tag{B.28}$$

and

$$\frac{\partial(\mathbf{R}^{\mathrm{a:b}}\,{}^{\mathrm{b}}t^{\mathrm{b:c}} + {}^{\mathrm{a}}t^{\mathrm{a:b}})}{\partial\, {}^{\mathrm{a}}t^{\mathrm{a:b}}} = I. \tag{B.29}$$

Finally, the term $\mathbf{R}^{\mathrm{a:b}}\,{}^{\mathrm{b}}t^{\mathrm{b:c}}$ describes a co-ordinate mapping, so by Equation (B.8),

$$\frac{\partial(\mathbf{R}^{\mathrm{a:b}}\,{}^{\mathrm{b}}t^{\mathrm{b:c}} + {}^{\mathrm{a}}t^{\mathrm{a:b}})}{\partial \mathbf{R}^{\mathrm{a:b}}} = -[\mathbf{R}^{\mathrm{a:b}}\,{}^{\mathrm{b}}t^{\mathrm{b:c}}]^{\wedge}. \tag{B.30}$$

Therefore, the derivative of the composition function from Equation (B.26) is computed as

$$\frac{\partial \mathbf{T}^{\mathsf{a:b}}\mathbf{T}^{\mathsf{b:c}}}{\partial \mathbf{T}^{\mathsf{a:b}}} = \left[\begin{array}{c|c} I & 0 \\ \hline -[\mathbf{R}^{\mathsf{a:b}}\,{}^{\mathsf{b}}t^{\mathsf{b:c}}]^{\wedge} & I \end{array}\right] \tag{B.31}$$

$$= \left[\begin{array}{c|c} I & 0 \\ \hline -[{}^{\mathsf{a}}t^{\mathsf{b:c}}]^{\wedge} & I \end{array}\right]. \tag{B.32}$$

Similarly, the Jacobian of the composition from Equation (B.24) with respect to the right transformation, $\mathbf{T}^{\mathsf{b:c}}$, has the form

$$\frac{\partial \mathbf{T}^{\mathsf{a:b}}\mathbf{T}^{\mathsf{b:c}}}{\partial \mathbf{T}^{\mathsf{b:c}}} = \left[\begin{array}{c|c} \frac{\partial(\mathbf{R}^{\mathsf{a:b}}\mathbf{R}^{\mathsf{b:c}})}{\partial \mathbf{R}^{\mathsf{b:c}}} & \frac{\partial(\mathbf{R}^{\mathsf{a:b}}\mathbf{R}^{\mathsf{b:c}})}{\partial\,{}^{\mathsf{b}}t^{\mathsf{b:c}}} \\ \hline \frac{\partial(\mathbf{R}^{\mathsf{a:b}}\,{}^{\mathsf{b}}t^{\mathsf{b:c}}+{}^{\mathsf{a}}t^{\mathsf{a:b}})}{\partial \mathbf{R}^{\mathsf{b:c}}} & \frac{\partial(\mathbf{R}^{\mathsf{a:b}}\,{}^{\mathsf{b}}t^{\mathsf{b:c}}+{}^{\mathsf{a}}t^{\mathsf{a:b}})}{\partial\,{}^{\mathsf{b}}t^{\mathsf{b:c}}} \end{array}\right], \tag{B.33}$$

where by Equation (B.3),

$$\frac{\partial(\mathbf{R}^{\mathsf{a:b}}\mathbf{R}^{\mathsf{b:c}})}{\partial \mathbf{R}^{\mathsf{b:c}}} = \mathbf{R}^{\mathsf{a:b}}, \tag{B.34}$$

and by Inspection

$$\frac{\partial(\mathbf{R}^{\mathsf{a:b}}\mathbf{R}^{\mathsf{b:c}})}{\partial\,{}^{\mathsf{b}}t^{\mathsf{b:c}}} = 0, \tag{B.35}$$

$$\frac{\partial(\mathbf{R}^{\mathsf{a:b}}\,{}^{\mathsf{b}}t^{\mathsf{b:c}} + {}^{\mathsf{a}}t^{\mathsf{a:b}})}{\partial \mathbf{R}^{\mathsf{b:c}}} = 0, \tag{B.36}$$

and

$$\frac{\partial(\mathbf{R}^{\mathsf{a:b}}\,{}^{\mathsf{b}}t^{\mathsf{b:c}} + {}^{\mathsf{a}}t^{\mathsf{a:b}})}{\partial\,{}^{\mathsf{b}}t^{\mathsf{b:c}}} = \mathbf{R}^{\mathsf{a:b}}. \tag{B.37}$$

Therefore, the derivative of the composition function from Equation (B.33) is computed as

$$\frac{\partial \mathbf{T}^{\mathsf{a:b}}\mathbf{T}^{\mathsf{b:c}}}{\partial \mathbf{T}^{\mathsf{b:c}}} = \left[\begin{array}{c|c} \mathbf{R}^{\mathsf{a:b}} & 0 \\ \hline 0 & \mathbf{R}^{\mathsf{a:b}} \end{array}\right]. \tag{B.38}$$

B.4 Box-minus Jacobian for $\mathbb{SE}(3)$

Suppose we have two transformation matrices, $\mathbf{T}^{\mathsf{a:c}} \in \mathbb{SE}(3)$ and $\mathbf{T}^{\mathsf{b:c}} \in \mathbb{SE}(3)$, with their resulting box-minus difference

$$\rho^{\mathsf{a:b}} = \mathbf{T}^{\mathsf{a:c}} \boxminus \mathbf{T}^{\mathsf{b:c}} \tag{B.39}$$

$$= \begin{bmatrix} \mathbf{R}^{\mathsf{a:c}} \boxminus \mathbf{R}^{\mathsf{b:c}} \\ \hline {}^{\mathsf{a}}t^{\mathsf{a:c}} - {}^{\mathsf{b}}t^{\mathsf{b:c}} \end{bmatrix}. \tag{B.40}$$

The derivative of Equation (B.39) with respect to the left transformation, $\mathbf{T}^{\mathsf{a:c}}$, has the form

$$\frac{\partial \mathbf{T}^{\mathsf{a:c}} \boxminus \mathbf{T}^{\mathsf{b:c}}}{\partial \mathbf{T}^{\mathsf{a:c}}} = \left[\begin{array}{c|c} \frac{\partial(\mathbf{R}^{\mathsf{a:c}} \boxminus \mathbf{R}^{\mathsf{b:c}})}{\partial \mathbf{R}^{\mathsf{a:c}}} & \frac{\partial(\mathbf{R}^{\mathsf{a:c}} \boxminus \mathbf{R}^{\mathsf{b:c}})}{\partial\, {}^{\mathsf{a}}t^{\mathsf{a:c}}} \\ \hline \frac{\partial({}^{\mathsf{a}}t^{\mathsf{a:c}} - {}^{\mathsf{b}}t^{\mathsf{b:c}})}{\partial \mathbf{R}^{\mathsf{a:c}}} & \frac{\partial({}^{\mathsf{a}}t^{\mathsf{a:c}} - {}^{\mathsf{b}}t^{\mathsf{b:c}})}{\partial\, {}^{\mathsf{a}}t^{\mathsf{a:c}}} \end{array}\right], \tag{B.41}$$

where by Equation (B.15),

$$\frac{\partial(\mathbf{R}^{\mathsf{a:c}} \boxminus \mathbf{R}^{\mathsf{b:c}})}{\partial \mathbf{R}^{\mathsf{a:c}}} = [\Gamma(\log(\mathbf{R}^{\mathsf{a:b}}))]^{-1}, \tag{B.42}$$

and by inspection

$$\frac{\partial(\mathbf{R}^{\mathsf{a:c}} \boxminus \mathbf{R}^{\mathsf{b:c}})}{\partial\, {}^{\mathsf{a}}t^{\mathsf{a:c}}} = 0, \tag{B.43}$$

$$\frac{\partial({}^{\mathsf{a}}t^{\mathsf{a:c}} - {}^{\mathsf{b}}t^{\mathsf{b:c}})}{\partial \mathbf{R}^{\mathsf{a:c}}} = 0, \tag{B.44}$$

and

$$\frac{\partial({}^{\mathsf{a}}t^{\mathsf{a:c}} - {}^{\mathsf{b}}t^{\mathsf{b:c}})}{\partial\, {}^{\mathsf{a}}t^{\mathsf{a:c}}} = I, \tag{B.45}$$

and the derivative from Equation (B.41) has the form

$$\frac{\partial \mathbf{T}^{\mathsf{a:c}} \boxminus \mathbf{T}^{\mathsf{b:c}}}{\partial \mathbf{T}^{\mathsf{a:c}}} = \left[\begin{array}{c|c} [\Gamma(\log(\mathbf{R}^{\mathsf{a:b}}))]^{-1} & 0 \\ \hline 0 & I \end{array}\right]. \tag{B.46}$$

Similarly, the derivative of Equation (B.39) with respect to the right transformation, $\mathbf{T}^{\mathsf{b:c}}$, has the form

$$\frac{\partial \mathbf{T}^{\mathsf{a:c}} \boxminus \mathbf{T}^{\mathsf{b:c}}}{\partial \mathbf{T}^{\mathsf{b:c}}} = \left[\begin{array}{c|c} \frac{\partial(\mathbf{R}^{\mathsf{a:c}} \boxminus \mathbf{R}^{\mathsf{b:c}})}{\partial \mathbf{R}^{\mathsf{b:c}}} & \frac{\partial(\mathbf{R}^{\mathsf{a:c}} \boxminus \mathbf{R}^{\mathsf{b:c}})}{\partial\, {}^{\mathsf{b}}t^{\mathsf{b:c}}} \\ \hline \frac{\partial({}^{\mathsf{a}}t^{\mathsf{a:c}} - {}^{\mathsf{b}}t^{\mathsf{b:c}})}{\partial \mathbf{R}^{\mathsf{b:c}}} & \frac{\partial({}^{\mathsf{a}}t^{\mathsf{a:c}} - {}^{\mathsf{b}}t^{\mathsf{b:c}})}{\partial\, {}^{\mathsf{b}}t^{\mathsf{b:c}}} \end{array}\right], \tag{B.47}$$

where by Equation (B.20),

$$\frac{\partial(\mathbf{R}^{\mathsf{a:c}} \boxminus \mathbf{R}^{\mathsf{b:c}})}{\partial \mathbf{R}^{\mathsf{b:c}}} = \mathcal{J}_{\boxminus}^{\mathsf{b:c}}, \tag{B.48}$$

and by inspection

$$\frac{\partial(\mathbf{R}^{\mathsf{a:c}} \boxminus \mathbf{R}^{\mathsf{b:c}})}{\partial\, {}^{\mathsf{b}}t^{\mathsf{b:c}}} = 0, \tag{B.49}$$

$$\frac{\partial({}^{\mathsf{a}}t^{\mathsf{a:c}} - {}^{\mathsf{b}}t^{\mathsf{b:c}})}{\partial \mathbf{R}^{\mathsf{b:c}}} = 0, \tag{B.50}$$

and

$$\frac{\partial({}^{\mathsf{a}}t^{\mathsf{a:c}} - {}^{\mathsf{b}}t^{\mathsf{b:c}})}{\partial\, {}^{\mathsf{b}}t^{\mathsf{b:c}}} = -I, \tag{B.51}$$

and the derivative from Equation (B.47) is computed as,

$$\frac{\partial \mathbf{T}^{\mathsf{a:c}} \boxminus \mathbf{T}^{\mathsf{b:c}}}{\partial \mathbf{T}^{\mathsf{b:c}}} = \left[\begin{array}{c|c} \mathcal{J}_{\boxminus}^{\mathsf{b:c}} & 0 \\ \hline 0 & -I \end{array}\right]. \tag{B.52}$$