



DOCTORAL THESIS

Manifold Optimization for Robotic Perception

Submitted by:

Shin-Fang Chng

Supervised by:

Prof. Tat-Jun Chin

Dr. Yasir Latif

*A thesis submitted in total fulfillment for the
degree of Doctor of Philosophy*

in the

Faculty of Engineering, Computer and Mathematical Sciences
School of Computer Science

August 2021

Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I acknowledge that copyright of published works contained within this thesis resides with the copyright holder(s) of those works.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

I acknowledge the support I have received for my research throughout the provision of a scholarship from the ARC Centre of Excellence on Robotic Vision CE140100016 and the Mawson Lakes Fellowship Program.

Signed: _____

Date: 18-Dec-2020

THE UNIVERSITY OF ADELAIDE

Abstract

School of Computer Science

Doctor of Philosophy

Manifold Optimization for Robotic Perception

by Shin-Fang Chng

Robotic perception plays a crucial role in endowing a robot with human-like perception. This entails the ability to perceive and understand about the unstructured world from the sensor modalities, which would allow it to navigate autonomously through the environment to accomplish a task. Recent years have witnessed an unprecedented enthusiasm in robotic perception research as it promises a vast variety of compelling applications such as self-driving cars, drone technology, domestic robots, virtual and augmented reality.

An essential task in robotic perception is state estimation. Generally, the task is concerned with inferring the state, such as the pose of an entity from observations in the form of inertial and/or visual measurements. Such an inverse problem can usually be formulated as an optimization problem, that seeks to select the best model from the imperfect sensor data. This thesis falls under the paradigm of state estimation, which aims to address the pose estimation and Simultaneous Localisation and Mapping (SLAM) problems.

Solving pose estimation and SLAM problems typically involve estimating rotations. However, they naturally reside in the manifold space, i.e., the special orthogonal group $SO(3)$, where Euclidean geometry with which we are familiar is no longer applicable. To reliably and accurately deploy state estimation algorithms for real-world applications, the underlying optimization problems must be able to properly address the inherent non-convexity of the manifold constraints, which is the main contribution of this thesis.

Despite previous developments in state estimation, there remain unsatisfactorily solved problems, specifically, problems associated with outliers and large-scale input observations. This thesis is devoted to developing novel techniques to address these problems, in a manner that respects the manifold structure.

The first part of the thesis is concerned with the sensor fusion problem in the context of INS/GPS fusion. While a ‘de-facto’ standard for the sensor fusion problem is the filtering technique, it is highly susceptible to outlier measurements. This thesis proposes a method to address the outlier-prone sensor fusion problem with a robust nonlinear optimization framework, underpinned by a novel pre-integration theory.

An influential optimisation strategy in SLAM is rotation averaging, which aims to estimate the absolute orientation, given a set of relative orientations that are in general incompatible. It stems from the fact that if the rotations containing non-convex constraints were solved first, then the remaining problem involving structure and translation would be easier to deal with. Inspired by Lagrangian duality, this thesis contributes a globally-optimal rotation averaging algorithm which is capable of handling large-scale input measurements much more efficiently.

Finally, a specialised rotation averaging algorithm underpinned by a novel lifting technique, is proposed to resolve the fundamental ambiguity problem in marker-based SLAM. We demonstrate how to resolve the ambiguity problem by exploiting the special problem structure, which is then able to achieve a more accurate and/or complete marker-based SLAM.

Acknowledgements

There are many important people whom I would like to thank for helping me to navigate through the ups and downs of my Ph.D. journey.

First, I would like to thank my supervisor, Prof. Tat-Jun Chin, who has been so dedicated in molding me from a confused student to an independent researcher. I am extremely inspired by his research philosophy, passion, and vision in science. When I poorly explained an idea, he carefully pinpointed my flawed statement or reasoning. When I produced terrible paper drafts, he patiently provided me with constructive suggestions. I am immensely grateful to TJ not only for his technical guidance or his words of wisdom, but also for teaching me how to think critically and how to convey my research ideas effectively.

I would also like to thank my co-supervisor, Dr. Yasir Latif, for his valuable guidance, support and encouragement.

I would like to thank Dr. Alireza Khosravian, Dr. Alvaro Parra Bustos, and Dr. Pulak Purkait, who provided me with critical comments, feedback, and advice during our collaboration.

I would like to thank Dr. Huu Le for his general advice and encouragement.

I would like to thank the incredible people whom I have been fortunate to meet and become friends with throughout my Ph.D, for the unforgettable moments and fruitful research discussions. A special shout-out to Huangying Zhan, Peishen Liu, Kejie Li, Michelle Liu and Ming Cai who make me feel like home.

I would like to thank my friends who traveled 5,1000km to visit me. A big thank you to Lixian Chang, Shu Hua Lee, Kelvin Chan, Yunna Chong and Ryan Ch'ng for the amazing memories.

I am extremely grateful to Chee Kheng Ch'ng and Manlin Loh, for their relentless support, encouragement, and always pushing me to challenge myself. Thanks for helping me to get through the difficult times in my Ph.D. journey.

Finally, I would like to dedicate my sincere thanks to my family for their unconditional love and support. I hope I have made them proud.

Contents

Declaration of Authorship	ii
Abstract	iv
Acknowledgements	vi
Publications	xi
1 Introduction	1
1.1 State Estimation	2
1.2 Front-End for Robotic Perception	3
1.2.1 IMU	3
1.2.2 GPS	4
1.2.3 Magnetometer	4
1.2.4 Monocular Camera	5
1.3 Inertial Navigation System (INS)/GPS for Pose Estimation	7
1.4 Visual SLAM/SfM	8
1.4.1 Marker-based SLAM	10
1.5 Manifold	11
1.6 Back-End	12
1.6.1 Filtering for INS/GPS Fusion	12
1.6.2 Bundle Adjustment for Visual SLAM	13
1.6.2.1 Pose Averaging	15
1.6.2.2 Rotation Averaging	16
1.7 Summary of Contributions	17
1.8 Thesis Outline	17
2 Literature Review	19
2.1 Rotation Representation	20
2.1.1 The Matrix Lie Group	20
The Lie bracket	21
Baker-Campbell-Hausdorff	22
2.1.2 Quaternion	22

2.2	Distance Metrics on $SO(3)$	23
2.2.1	Angular Distance / Geodesic Distance	23
2.2.2	Chordal Distance	23
2.2.3	Quaternion Distance	24
2.3	Algorithms for the INS/GPS fusion	24
2.3.1	Standard filter	24
2.3.1.1	KF	24
2.3.1.2	EKF	26
2.3.2	Robust Filtering	27
2.3.2.1	Ad-hoc methods	27
2.3.2.2	Weighted-based filtering methods	28
2.3.3	Filtering on Manifolds	29
2.4	Algorithms for Multiple Rotation Averaging	29
2.4.1	Extrinsic-averaging-based algorithms	29
2.4.1.1	Quaternion Relaxation	30
2.4.1.2	Chordal Relaxation	30
2.4.2	Intrinsic-averaging-based algorithm	31
2.4.2.1	Least Squares	33
2.4.2.2	M-estimators	33
2.4.2.3	Weighted Least Squares	34
2.4.2.4	L1 Weiszfeld algorithm	35
2.4.3	Preprocessing	36
2.4.3.1	Random Sampling Method	36
2.4.3.2	Bayesian Method	37
2.4.4	Duality-based algorithms	37
2.4.4.1	Riemannian Staircase Method	39
2.4.4.2	Shonan Method	40
2.4.4.3	Block Coordinate Descent Method (BCD)	41
3	Outlier-Robust Manifold Pre-Integration for INS/GPS Fusion	42
4	Rotation Coordinate Descent for Fast Globally Optimal Rotation Averaging	53
5	Resolving Marker Pose Ambiguity by Robust Rotation Averaging with Clique Constraints	69
6	Conclusions and Future Work	79
6.1	Future Work	79
6.1.1	The Outlier-Robust INS/GPS Fusion	79
6.1.2	The Rotation Coordinate Descent Algorithm	80
6.1.3	The Clique-Constrained Rotation Averaging Algorithm	80

Bibliography

81

Publications

This thesis is in part result of the work presented in the following papers:

- Shin-Fang Chng, Alireza Khosravian, Anh-Dzung Doan and Tat-Jun Chin: Outlier-Robust Manifold Pre-Integration for INS/GPS Fusion. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2019.
- Alvaro Parra Bustos*, Shin-Fang Chng*, Tat-Jun Chin, Anders Eriksson and Ian Reid: Rotation Coordinate Descent for Fast Globally Optimal Rotation Averaging. Computer Vision and Pattern Recognition (CVPR) 2021. * denotes equal contribution.
- Shin-Fang Chng, Naoya Sogi, Pulak Purkait, Tat-Jun Chin and Kazuhiro Fukui: Resolving Marker Pose Ambiguity by Robust Rotation Averaging with Clique Constraints. IEEE International Conference on Robotics and Automation (ICRA) 2020.

Chapter 1

Introduction

Robotics is an interdisciplinary discipline, which integrates computer science and engineering, aiming to develop machines that are capable of performing a complex series of actions, similar to humans. For instance, a machine can be a self-driving car, a spacecraft, a warehouse robot, a submarine and so on. Such intelligent systems have huge potential to dramatically enhance every aspect of human life, including increasing work productivity, being an excellent substitute for an unhealthy or hazardous environment and so on. A key example is that using robots instead of human to clear up the radioactive debris in the Chernobyl disaster could have significantly reduced the number of casualties.

Three decades ago, the idea of having robots performing purposeful tasks in any given specific environment was an absurd thought; whilst we remain far from achieving widespread advanced artificial intelligence (AI), sensing and computing advances have enabled deployment of robotic technology in a wide range of areas, such as self-driving cars, agriculture, space, domestic applications and so on. Today, Amazon has more than 200,000 mobile robots working inside its warehouses.

Robotic perception plays a central role in the realization of robots that can achieve human-like perception. Similar to the way we rely on our senses to relate to the world around us, robots has to be able to perceive and infer the unstructured world where they operate based on the noisy sensor data and make informed decisions about their tasks.

1.1 State Estimation

State estimation is a fundamental problem in robotic perception, which is concerned with inferring the mathematical quantities from measurements collected through sensors. Such an inverse problem can be formulated as an optimization problem, which seeks to select the estimates that best correlate with the observed measurements.

A typical state estimation pipeline can be divided into the *front-end* and the *back-end*. Raw sensor data e.g., visual images, inertial measurements or lidar point clouds are fed into the front-end, where measurements are extracted and processed. The back-end then uses nonlinear estimation techniques to determine the quantities of interest, e.g., the *pose* of the robot or the sensor bias.

State estimation plays a crucial role in enabling many computer vision and robotic applications. An example is a pose estimation problem where one wishes to estimate the position and orientation of a robot or object relative to some coordinate system [30]. Another example is object tracking which involves estimating the velocity and acceleration of a moving object in addition to the object pose. Often, the robot's perception capabilities can benefit from acquiring a geometric representation of the environment through which it is navigating. The concurrent estimation of pose and the 3D structure of the scene is referred as Simultaneous Localisation and Mapping (SLAM) [10, 22] or Structure from Motion (SfM) [63]. Other examples of applications that entail state estimation include shape reconstruction [12], 3D reconstruction [31], virtual reality and augmented reality [35, 36].

State estimation is a challenging task. As the perception front-end is generally imperfect, measurements derived from the sensors are often noisy and potentially corrupted with bias and/or outliers, which sets up the requirement that the underlying state estimation methods must be robust. Take, for example, the task of estimating pose from bias-contaminated inertial measurements. Due to the inherent errors of IMU, practitioners often rely on filtering-based sensor fusion techniques to improve the reliability of the estimations.

Another key challenge is that the computational efficiency of state estimation problems is often exacerbated by the huge amount of sensory data available. Take, for example, the task of conducting long-term SLAM in a large-scale environment for

surveillance using a robot. For such a task, the size of the state can grow enormously. Due to the constrained resources of the robot in practice, it is essential to design efficient SLAM algorithms. In this thesis, we aim to address these challenges of state estimation problems, with a particular focus on cases that can be formulated as having a Lie group structure of rotations (manifold), specifically pose estimation and SLAM.

1.2 Front-End for Robotic Perception

The solutions to pose estimation and SLAM problems rely on the sensors employed in the front-end to perceive the world. The sensors can be divided into two categories: exteroceptive and proprioceptive sensors. Proprioceptive sensors, such as inertial measurement units (IMU), measure the internal values arising from the robot platform. Exteroceptive sensors extract quantities related to the robot's environment, such as global positioning systems (GPS), cameras and lidars. This section briefly introduces some of the sensors we used to tackle the pose estimation and SLAM problems.

1.2.1 IMU

An IMU typically consists of a 3-axis gyroscope and a 3-axis accelerometer [1]. A gyroscope gives the angular velocity measurement $\tilde{\boldsymbol{\omega}}_B$ whereas an accelerometer measures the acceleration $\tilde{\mathbf{a}}_B$, whose reference is denoted as B, at regular intervals Δt . As IMUs have low cost, weight and power consumption, they are commonly used in an exceptionally broad range of applications, such as unmanned aerial vehicles (UAVs), spacecraft, and self-driving cars.

In practice, both measurements suffer from the slowly varying biases \mathbf{b}_g and \mathbf{b}_a of the gyroscope and accelerometer, respectively, in addition to sensor noise. As in [37], using Euler integration, the pose $(\mathbf{R}_{B,W}, {}_W\mathbf{p}_B)$ and velocity ${}_W\mathbf{v}_B$, in the world

reference W can be computed as

$$\begin{aligned} \mathbf{R}_{B,W}^{(t+1)} &= \mathbf{R}_{B,W}^{(t)} \exp \left((\tilde{\boldsymbol{\omega}}_B^{(t)} - \mathbf{b}_g^{(t)})_{\times} \Delta t \right) \\ {}_W \mathbf{v}_B^{(t+1)} &= {}_W \mathbf{v}_B^{(t)} + \mathbf{g}_W \Delta t + \mathbf{R}_{B,W}^{(t)} (\tilde{\mathbf{a}}_B^{(t)} - \mathbf{b}_a^{(t)}) \Delta t \\ {}_W \mathbf{p}_B^{(t+1)} &= {}_W \mathbf{p}_B^{(t)} + {}_W \mathbf{v}_B^{(t)} \Delta t + \frac{1}{2} \mathbf{g}_W \Delta t^2 + \frac{1}{2} \mathbf{R}_{B,W}^{(t)} (\tilde{\mathbf{a}}_B^{(t)} - \mathbf{b}_a^{(t)}) \Delta t^2, \end{aligned} \quad (1.1)$$

where \mathbf{g}_W is the gravity and $\exp(\cdot)$ denotes the exponential mapping from $\mathfrak{so}(3)$ to $\text{SO}(3)$; see Section 2.1.

Ideally, if IMUs were to give perfect measurements, the estimated pose would be perfect. However, in practice, IMUs suffer from slowly varying biases, as described previously. Observe that in (1.1), the orientation $\mathbf{R}_{B,W}$ and position ${}_W \mathbf{p}_B$ are recursively estimated through onefold and twofold integration of the inertial measurements, respectively. In addition, the estimated position ${}_W \mathbf{p}_B$ is inherently linked to the estimated orientation $\mathbf{R}_{B,W}$. As a result, *dead reckoning*, which recursively computes the current pose using a previously determined pose, is susceptible to drift over time.

1.2.2 GPS

A GPS unit provides absolute position and velocity data, which is especially useful for outdoor localization. Generally, it performs well in most of the cases where there are unobstructed lines of sight to four or more GPS satellites. However, the signals are extremely vulnerable to blockage from tall buildings and terrain. Therefore, in the obstructed scenarios when multipath blocking occurs, the accuracy of the GPS receiver dramatically degrades and tends to provide inaccurate measurements [21]. Moreover, GPS has a low update frequency, which happens at typically 1-10Hz. As a result, the low sampling rate poses a significant limitation for high-dynamic applications, especially UAVs where onboard navigation algorithms usually run as fast as 100Hz.

1.2.3 Magnetometer

A 3-axis magnetometer is generally useful for aircraft attitude (orientation) estimation. It measures the magnetic field of the earth in a body-fixed frame. The ideal

magnetometer output \mathbf{m}_B gives partial information of the orientation $\mathbf{R}_{B,W}$ as:

$$\mathbf{m}_B = \mathbf{R}_{B,W}^T \mathbf{m}_W, \quad (1.2)$$

where \mathbf{m}_W is the (approximately constant) magnetic field of the earth at the position of the rigid body expressed in the world frame W .

A major drawback of magnetometers is being vulnerable to outliers as magnetometers are susceptible to magnetic interference. Example of sources of unanticipated magnetic disturbances include smartphones or motors, which are commonly available [48].

1.2.4 Monocular Camera

As cameras are small, inexpensive, and ubiquitous, the last two decades have witnessed an unprecedented deployment of cameras in robotic applications. A camera model plays an important role when encapsulating the geometry between the 3D world and the 2D image mathematically. A commonly used camera model is the *pinhole camera model*. As seen in Figure 1.1, this model assumes that a point $\mathbf{X} = [X_1, X_2, X_3]^T$ in 3D space is back-projected to the point $\mathbf{u} = [u_1, u_2]^T$ on the image plane \mathcal{I} , where a line joining the point \mathbf{u} to the camera centre \mathbf{C} coincides with the image plane. Such a perspective projection function $g(\cdot) : \mathbb{R}^3 \mapsto \mathbb{R}^2$ can be described as:

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \frac{fX_1}{X_3} \\ \frac{fX_2}{X_3} \end{bmatrix}, \quad (1.3)$$

where f is the focal length of the camera.

Let $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{u}}$ be denoted as the homogeneous coordinates of \mathbf{X} and \mathbf{u} respectively [46], (1.3) can be rewritten as

$$\tilde{\mathbf{u}} = \begin{bmatrix} fX_1 \\ fX_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{\mathbf{X}}. \quad (1.4)$$

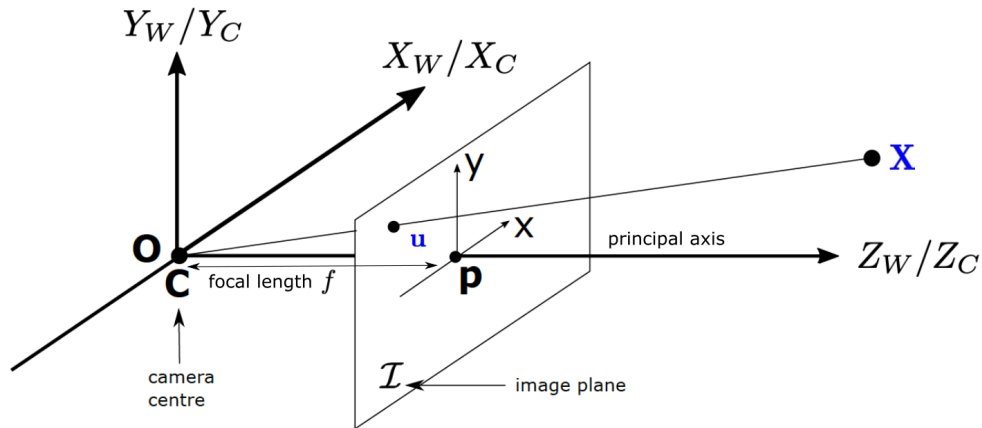


FIGURE 1.1: Pinhole camera model where \mathbf{X} is a 3D scene point with its corresponding projection \mathbf{u} on the image \mathcal{I} . The world coordinate system $[X_W, Y_W, Z_W]$ coincides with the camera coordinate system $[X_C, Y_C, Z_C]$. Observe that the camera centre \mathbf{C} lies at the origin \mathbf{O} of the world coordinate system and both the camera axes and world axes are aligned with each other.

We further recast (1.4) as

$$\tilde{\mathbf{u}} = P\tilde{\mathbf{X}}, \quad (1.5)$$

where the *camera projection matrix* $P \in \mathbb{R}^{3 \times 4}$ can be further decomposed as

$$P = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I_3 & | & 0 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{R} & | & \mathbf{t} \end{bmatrix}, \quad (1.6)$$

where \mathbf{K} is the *camera calibration matrix*, the *camera pose* consisting \mathbf{R} and \mathbf{t} define the orientation and translation from the *world coordinate system* to the *camera coordinate system*, respectively.

Since we assume the *world coordinate system* coincides with the *camera coordinate system* for ease of exposition in this example (see Figure 1.1), observe that \mathbf{R} is an identity matrix and \mathbf{t} is a zero vector. Often, the points in space are available in a different Euclidean coordinate frame. Therefore, \mathbf{R} can be any valid 3×3 rotation matrix and \mathbf{t} can be computed as

$$\mathbf{t} = -\mathbf{R}\tilde{\mathbf{C}}, \quad (1.7)$$

where $\tilde{\mathbf{C}}$ represents the 3D coordinates of the camera centre \mathbf{C} in the world coordinate system [46].

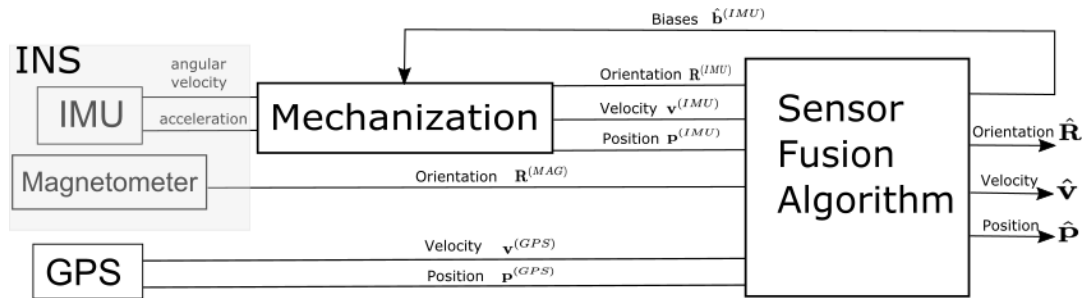


FIGURE 1.2: A typical INS/GPS fusion pipeline.

Generally, the principle point \mathbf{p} does not lie at the origin of the camera coordinate system on the image plane \mathcal{I} . Therefore, we define a more general camera calibration matrix \mathbf{K} as

$$\mathbf{K} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (1.8)$$

where $\mathbf{p} = \begin{bmatrix} p_x \\ p_y \end{bmatrix} \in \mathbb{R}^2$ is the 2D coordinates of \mathbf{p} on \mathcal{I} .

1.3 Inertial Navigation System (INS)/GPS for Pose Estimation

Recent advances in micro-electromechanical systems have led to a considerable amount of interest in developing low cost pose estimation solutions based on IMU, especially for unmanned aerial vehicle (UAV) navigation, which plays an important role in aviation [2, 49, 50, 60]. Inertial odometry operates by incrementally estimating the pose of the robot relative to the initial pose. Such an approach functions by integrating the angular velocity and acceleration obtained from the IMU; see Section 1.2.1.

A major drawback of inertial odometry is that the pose estimates derived from the highly sampled IMU measurements are susceptible to inevitable drift over time due to the inherent bias in the measurements, as described in Section 1.2.1. Therefore, using IMU individually cannot provide reliable pose estimates.

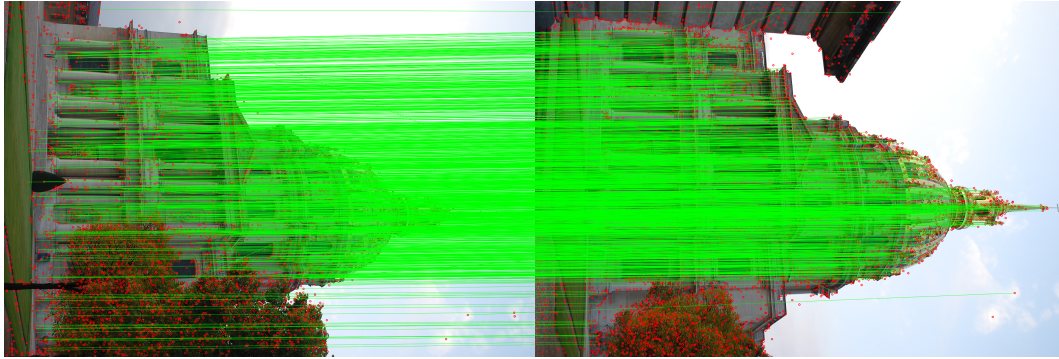


FIGURE 1.3: Feature-extraction-and-matching on two images, where each “*red dot*” denotes the extracted feature point; the left image is matched with the corresponding ones on the right image indicated by the “*green line*”.

Multi-sensor fusion is a well-known technique for combining multiple disparate and/or identical sensory data, such that the resulting information can be more accurate, reliable or complete compared to when they are used individually [52, 53, 50]. To mitigate IMU drift, a popular fusion strategy is to fuse the IMU with a low-sampling rate GPS, which provides drift-free absolute position measurements due to their complementary natures [21]. Another commonly used method is to fuse the IMU with a magnetometer, which provides partial pose information to realise the *Inertial Navigation System* (INS) [42]. This thesis focuses on integrating the IMU with both GPS and a magnetometer as the front-end for pose estimation, in a setting where the IMU, magnetometer and GPS are functioning at different frequencies. For brevity, INS will be used to denote IMU/Magnetometer.

Figure 1.2 demonstrates a typical INS/GPS fusion pipeline. The *mechanization* procedure processes raw measurements from the IMU to obtain the pose estimates. Once GPS data is available, the absolute position and velocity estimates from the GPS receiver are merged with the INS solution through a sensor-fusion algorithm in the back-end module. The error states containing the IMU biases, which have been estimated by the back-end, are then fed back to the mechanization procedure to compensate for the inherent IMU bias.

1.4 Visual SLAM/SfM

SLAM aims at building a globally consistent geometric representation of the environment by returning to a previously visited location (*loop closure*). In contrast to



FIGURE 1.4: A demonstration of SfM, where each “*red prism*” denotes the position of the camera centre, and each “*dot*” denotes the reconstructed 3D points.

odometry, which only estimates the poses (motion), SLAM provides the 3D geometry (structure) of the unknown scene in addition to the poses. Therefore, the main factor that distinguishes odometry and SLAM is *mapping*.

The advantage of having a map of the environment is twofold. First, the map can benefit other applications such as path planning, augmented reality and virtual reality. Second, the map can alleviate the drift problem in localization by loop closure.

Many different types of sensors have been applied to SLAM, such as Lidars, inertial sensors, GPS, and cameras; we refer the reader to [10, 69] for an excellent survey of such algorithms. This thesis tackles SLAM using a monocular camera, more specifically a feature-based visual SLAM. Visual SLAM is a special case of SfM, which considers measurements received in a *sequential* manner.

The input to the front-end module is a stream of images with overlapping views. Distinctive points in the image are extracted and matched across the images to generate a set of feature correspondences; see Figure 1.3. The feature correspondences are then used to compute the camera poses and the 3D structure of the scene, which are refined by the back-end optimisation; see Figure 1.4 for an output of SfM.

Mathematically, visual SLAM/SfM is an optimization problem. Formally, let $\mathbf{u}_{i,k}$ be the 2D measured image coordinates of the i th scene point, as observed by the

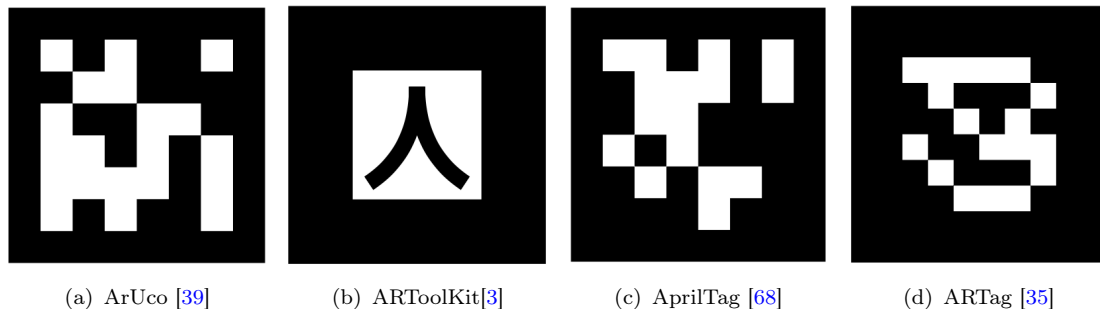


FIGURE 1.5: Examples of binary-squared fiducial markers

k th camera C_k ; SLAM estimates the 3D coordinates $\{\mathbf{X}_i\}_{i=1}^N$ of the scene points and the 6DOF poses $\{(\mathbf{R}_k, \mathbf{t}_k)\}_{k=1}^M$, which are consistent with the image observations, as

$$\min_{\{\mathbf{X}_i\}_{i=1}^N, \{(\mathbf{R}_k, \mathbf{t}_k)\}_{k=1}^M} \sum_{i=1}^N \sum_{k=1}^M \mathbb{I}_{i,k} \left(\left\| \mathbf{u}_{i,k} - g(\mathbf{X}_i | \mathbf{R}_k, \mathbf{t}_k) \right\|_2^2 \right), \quad (1.9)$$

where $g(\mathbf{X}_i | \mathbf{R}_k, \mathbf{t}_k)$ is the projection of \mathbf{X}_i onto C_k (assuming calibrated cameras). $\mathbb{I}_{i,k}$ is an indicator function, which returns 1 if the 3D point i is visible in image k and 0 otherwise.

As (1.9) concurrently estimates the 3D structure of the scene and the camera poses, solving (1.9) is generally difficult. Being a non-linear least squares problem, (1.9) is commonly solved using Levenberg-Marquardt, which necessitates a good initialization for \mathbf{X}_i and $(\mathbf{R}_k, \mathbf{t}_k)$ to avoid convergence to bad local optima.

1.4.1 Marker-based SLAM

Marker-based SLAM is a special case of visual SLAM, which commonly employs binary square *fiducial markers* to simplify the front-end module, as it can be easily detected and associated across images. The fiducial marker is comprised of an external black border, which facilitates its fast detection in the image, and an inner binary code that defines the size of the marker, as well as uniquely distinguishes one from another [39, 68]; see Figure 1.5 for an example of different fiducial markers. In addition, the marker is a convenient way of providing real-scale information about the scene. Although marker-based SLAM entails affixing fiducial markers to the scene, the effort is negligible and finds practical use in a constrained environment such as factories, warehouses, mines, and so on [57, 58].

The front-end module of the marker-based SLAM receives a set of images containing markers. Let $\mathbf{m}_{i,k}^c$ be the 2D measured coordinates of each corner c of the i th marker extracted from a standard marker detection and identification algorithm [20], as observed by the k th camera; marker-based SLAM aims to estimate the camera pose \mathbf{P}_k and the marker pose \mathbf{P}_i that agree with the detected marker corners. The optimization problem of marker-based SLAM can be established as:

$$\min_{\{\mathbf{P}_i\}_{i=1}^N, \{\mathbf{P}_k\}_{k=1}^M} \sum_{k=1}^M \sum_{i=1}^N \sum_{c=1}^4 \left\| \mathbf{m}_{i,k}^c - g(\mathbf{P}_k, \mathbf{P}_i, \mathbf{X}^c) \right\|_2^2, \quad (1.10)$$

where $g(\mathbf{P}_k, \mathbf{P}_i, \mathbf{X}^c)$ is the projection of \mathbf{X}^c of the i th marker onto the k th camera (assuming calibrated cameras). Since the size s of the fiducial marker is known beforehand, its four corners $\{\mathbf{X}^c\}_{c=1}^4 \in \mathbb{R}^3$ can be expressed relative to the marker centre as

$$\mathbf{X}^1 = [0, 0, 0], \quad \mathbf{X}^2 = [0, s, 0], \quad \mathbf{X}^3 = [s, s, 0], \quad \mathbf{X}^4 = [s, 0, 0], \quad (1.11)$$

where $c = \{1, \dots, 4\}$ indexes the 4 corners of the marker.

Observe that there is a resemblance between (1.10) and (1.9); however, a key difference is that, in contrast to (1.9) which assumes the points to be independent from each other, (1.10) encapsulates the 3D points of the marker in its pose \mathbf{P}_i to well-constrain the size of the marker to be s .

1.5 Manifold

Before embarking on a discussion of different algorithms to address the state estimation problem in the back-end, we first introduce the concept of the *manifold*. Solving pose estimation and SLAM problems typically involves estimating the rotations, which lie on the manifold.

Generally, a manifold is a space where each point has a neighbourhood, which locally resembles Euclidean space [4]. An intuitive example to understand the concept is a globe's surface, which can be described by an atlas.

A Lie group is a group that has a differentiable manifold, whose product and inverse operations are smoothly differentiable [5]. A 3D rotation group, often denoted as a Special Orthogonal Group $\text{SO}(3)$, is a Lie group.

Formally, $\text{SO}(3)$ is defined as

$$\text{SO}(3) = \{\mathbf{R} \in \mathbb{R}^{3 \times 3} \mid \mathbf{R}^T \mathbf{R} = \mathbf{I}_3, \det(\mathbf{R}) = 1\}, \quad (1.12)$$

where \mathbf{R}^T denotes the *transpose* of \mathbf{R} and \mathbf{I}_3 is the 3×3 identity matrix.

By definition, $\text{SO}(3)$ in (1.12) must fulfill two constraints, i.e.,

$$\text{Orthogonality constraint : } \mathbf{R}^T \mathbf{R} = \mathbf{I}_3 \quad (1.13)$$

$$\text{Determinant constraint : } \det(\mathbf{R}) = 1 \quad (1.14)$$

Both the orthogonality condition (1.13) and positive determinant constraint (1.14) play important roles in preserving the angle, length, and orientation. Such properties are crucial in state estimation problems, as rotations are often employed to represent orientation (rigid motion) of the rigid body in \mathbb{R}^3 .

A compelling benefit of the manifold theory is that instead of working directly on $\text{SO}(3)$, it enables many operations to be performed on its associated vector space, which is geometrically intuitive and simpler than $\text{SO}(3)$; see Section 2.1.1 for further details.

1.6 Back-End

This section introduces the back-end optimization employed in this thesis to address the pose estimation and Visual SLAM/SfM described in Sections 1.3 and 1.4.

1.6.1 Filtering for INS/GPS Fusion

Stochastic filtering techniques, especially the Kalman filter (KF), which was first introduced in 1960 by Rudolf E. Kalman., is the de-facto standard for multi-sensor fusion problems. KF operates on a series of Gaussian distributed sequential measurements to *recursively* estimate the underlying states that tend to be more accurate than those based on a single measurement alone. The main assumption of KF is that the probability distribution of the dynamic system can be sufficiently modeled by the mean and covariance of a Gaussian distribution. The optimality of KF assumes the errors are Gaussian.

KF operates in two main stages, which are *predict* and *update*. During the prediction step, KF estimates the current states along with the covariances. Whenever a new measurement is observed, the posterior estimates are then computed using a Kalman gain, which specifies the relative weight given to the measurement and the current state estimates. If the gain is low, KF places a higher emphasis on the predicted states. On the other hand, if the gain is high, KF permits a higher weight on the measurements.

Due to its simplicity and well-understood mathematical theory, KF has been developed and deployed in a broad range of applications [56, 21, 48]. The popularity of KF has inspired numerous extensions. Among the KF variants, the extended Kalman filter (EKF) is a notable extension of KF, which can be applied on nonlinear systems by linearizing the current estimates.

However, since the optimality of standard EKF and its variants assume the errors are Gaussian, they are highly susceptible to outliers [67]. Data acquisition systems are not error-proof, hence they are prone to giving erroneous measurements in practice; see Section 1.2. As a result, EKF's accuracy will degrade dramatically in the presence of anomalies. Moreover, the standard EKF does not intrinsically exploit the geometry of the Lie group structure of the INS/GPS fusion problem. Dealing with such a constrained INS/GPS fusion problem naively via Euclidean geometry tools may lead to ill-posed problems and affect the stability of the filter. This thesis makes progress towards addressing these two issues; see Chapter 3.

1.6.2 Bundle Adjustment for Visual SLAM

There are mainly two prevalent techniques for the back-end of Visual SLAM: a filtering-based approach and bundle adjustment (BA). This thesis focuses on the BA-based approach, as BA has proven to be more efficient and accurate than the filtering approach, given the equivalent computing resources [64]; we refer the reader to [10, 69, 65] for details.

BA is the task of jointly adjusting the 3D structure and the camera poses together, according to a criterion entailing the corresponding image projections of the points [46]. Assume that N 3D points are observed in M views, as depicted in Figure 1.6. More formally, given $\mathbf{u}_{i,k}$ which denotes the projection of the i th 3D point observed on image k , BA minimizes the total reprojection error with respect

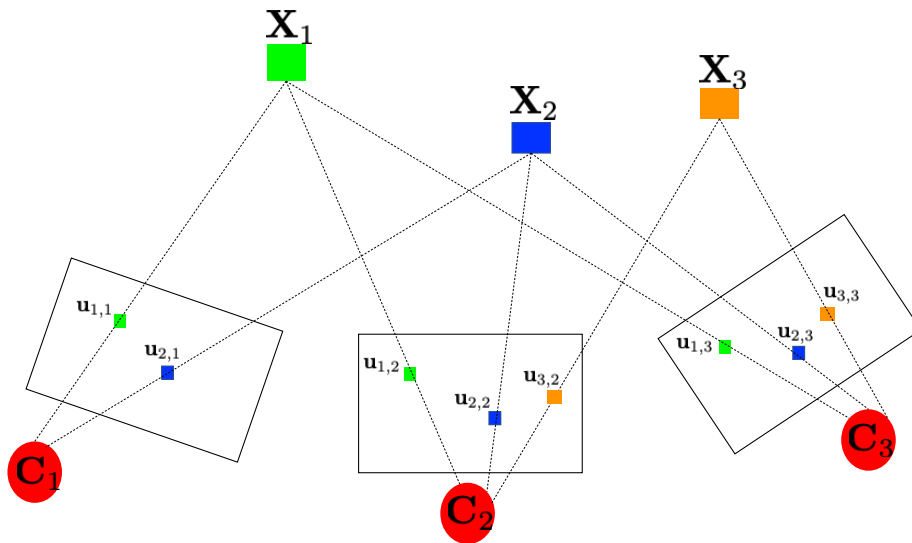


FIGURE 1.6: A bundle adjustment instance, where 3D point \mathbf{X}_i and its corresponding projection $\mathbf{u}_{i,k}$ which are observed by camera \mathbf{C}_k .

to all 3D points $\{\mathbf{X}_i\}_{i=1}^N$ and the 6DOF poses $\{(\mathbf{R}_k, \mathbf{t}_k)\}_{k=1}^M$ as

$$\min_{\{\mathbf{X}_i\}_{i=1}^N, \{(\mathbf{R}_k, \mathbf{t}_k)\}_{k=1}^M} \sum_{i=1}^N \sum_{k=1}^M \mathbb{I}_{i,k} \left(\rho \left(\left\| \mathbf{u}_{i,k} - g(\mathbf{X}_i | \mathbf{R}_k, \mathbf{t}_k) \right\|_2^2 \right) \right). \quad (1.15)$$

Observe that there is a resemblance between (1.9) and (1.15), except for the fact that (1.15) is a more robust cost function due to $\rho(\cdot)$, whose role is to downweight the influence of outlier measurements; in fact visual SLAM is essentially solving a bundle adjustment problem. The minimization problem (1.9) is commonly solved using a nonlinear least-squares algorithm e.g., Levenberg-Marguardt, which enables convergence up to local optimality.

Solving such a non-convex problem is notoriously challenging [28]. First, BA reduces to minimizing the sum of the squares of an *enormous* number of *complicated* nonlinear functions. Second, incorrect and/or spurious measurements tend to exist in visual data due to the imperfect front-end. Therefore, like other iterative algorithms that iteratively search the neighbourhood of the current estimate for a lower cost solution from an initial solution, it is vital to well-initialise the estimated variables in (1.15) to avoid convergence to poor solutions.

1.6.2.1 Pose Averaging

Another popular variation of this paradigm is *pose averaging*, which significantly reduces the number of variables in the optimization problem (1.15) by factoring out the structure, leading to a cost function that involves only camera poses [25]. Having the camera poses fixed, a structure-only BA is then solved to refine the 3D points.

Formally, we compactly rewrite pose (\mathbf{R}, \mathbf{t}) as $\mathbf{M} \in \text{SE}(3)$ as

$$\mathbf{M} = \left[\begin{array}{c|c} \mathbf{R} & \mathbf{t} \\ \hline 0_{1 \times 3} & 1 \end{array} \right]. \quad (1.16)$$

Consider a viewgraph $\mathcal{G}_M = (\mathcal{V}_M, \mathcal{E}_M)$, where each vertex denotes the unknown pose \mathbf{M}_i and each edge $(i, j) \in \mathcal{E}_M$ corresponds to a relative pose $\mathbf{M}_{i,j}$ between vertices i and j . Under the ideal condition, this entails finding $N = |\mathcal{V}_M|$ poses, which obey the relationship in (1.17).

$$\mathbf{M}_{i,j} = \mathbf{M}_j \mathbf{M}_i^{-1}, \forall (i, j) \in \mathcal{E}_M \quad (1.17)$$

The obvious gauge freedom can be easily eliminated by fixing any of the poses \mathbf{M}_i to the I_4 [45]. Specifically, if the pose corresponding to the first node is fixed, i.e., $\mathbf{M}_1 = I_4$, (1.17) admits unique solutions for $\{\mathbf{M}_i\}_{i=2}^N$.

However, in practice, when noise is inevitable, a solution to (1.17) is not guaranteed to exist. Thus, the pose averaging problem is often tackled as an optimisation problem, which minimizes the discrepancies with respect to the measurements $\tilde{\mathbf{M}}_{i,j}$, i.e.,

$$\min_{\{\mathbf{M}_i\}_{i=1}^N \in \text{SE}(3)} \sum_{(i,j) \in \mathcal{E}_M} \rho(d_{\text{SE}(3)}(\tilde{\mathbf{M}}_{i,j}, \mathbf{M}_j \mathbf{M}_i^T)), \quad (1.18)$$

where $d_{\text{SE}(3)}$ is the distance function between two poses in $\text{SE}(3)$ [25] and $\rho(\cdot)$ is a loss function defined over the distance.

1.6.2.2 Rotation Averaging

A main hurdle in pose averaging (1.18) is due to the rotation that resides in the nonlinear manifold $\text{SO}(3)$. Therefore, an alternative is to first solve a rotation averaging problem to obtain a good rotation estimation [40, 22, 45, 25], which will then be used to bootstrap the pose averaging (1.18).

The strength of this approach is two-fold: first, if the rotations were known and kept constant in the pose averaging problem, the resulting optimization problem would be a linear problem, whose translation can be computed efficiently; second, in certain cases, multiple rotation averaging can be solved up to global optimality [33, 34].

The input to rotation averaging is a set of noisy relative rotations $\{\tilde{\mathbf{R}}_{ij}\}$, where each $\{\tilde{\mathbf{R}}_{ij}\}$ is a measurement of the relative orientation between cameras i and j . Given the relative rotations, rotation averaging is concerned with estimating the absolute rotations $\{\mathbf{R}_i\}_{i=1}^N$. In an ideal case where the noise is absent in the relative rotations $\{\mathbf{R}_{ij}\}$, the compatibility constraint (1.19) holds.

$$\mathbf{R}_{ij} = \mathbf{R}_j \mathbf{R}_i^T. \quad (1.19)$$

The input relative rotation $\{\tilde{\mathbf{R}}_{ij}\}$ defines a camera graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ which encapsulates the geometric relationship between the cameras in the scene. $\mathcal{V} = \{1, \dots, n\}$ is the set of cameras and $(i, j) \in \mathcal{E}$ is an edge in \mathcal{G} if the relative rotation $\tilde{\mathbf{R}}_{ij}$ between cameras i and j can be measured using epipolar geometry [46].

As the input relative rotations $\{\tilde{\mathbf{R}}_{ij}\}$ are noisy in practice, there exist multiple paths between two vertices i, j with the aggregated relative rotations being inconsistent along different paths. Therefore, rotation averaging is usually posed as a nonlinear optimisation problem, whose goal is to find the average solution based on the inputs, i.e.

$$\min_{\{\mathbf{R}_i\}_{i=1}^N \in \text{SO}(3)} \sum_{(i,j) \in \mathcal{E}} \rho(d(\tilde{\mathbf{R}}_{ij}, \mathbf{R}_j \mathbf{R}_i^T)), \quad (1.20)$$

where $d : \text{SO}(3) \times \text{SO}(3) \mapsto \mathbb{R}$ is a distance function between two rotations in $\text{SO}(3)$ (see Section 2.2) and $\rho(\cdot)$ is a loss function defined over the chosen distance measure.

Solving (1.20) can be challenging [45]. As there is no closed-form solution for (1.20), the minimization problem is usually solved iteratively [28, 27, 44].

1.7 Summary of Contributions

In Chapter 3, an efficient and robust algorithm for an outlier-prone INS/GPS fusion problem is proposed. Aiming to offer a fresh insight into the long-standing INS/GPS fusion problem, which has been traditionally addressed with an EKF, we propose a novel non-linear optimization approach that fuses IMU and magnetometer measurements with GPS, that function at different frequencies, in a manner that respects the manifold structure of state space; and supports the usage of an M-estimator to mitigate the effects of outliers effectively.

Recent advancements in globally optimal rotation averaging have demonstrated some optimistic results by exploiting Lagrangian duality theory. Under mild conditions on the noise level of the measurements, rotation averaging satisfies the strong duality, which permits global solutions to be obtained by solving the semidefinite programming (SDP) relaxation. Unfortunately, generic solvers for the relaxed problem do not scale well to large input instances. Chapter 4 proposes a new algorithm that can efficiently find globally optimal rotations for large input instances.

While existing algorithms are developed for the general rotation averaging problem, we characterise and exploit the special problem structure to customise an efficient rotation averaging algorithm. Chapter 5 proposes such a 'bespoke' algorithm to resolve the fundamental pose ambiguity problem in marker-based SLAM.

1.8 Thesis Outline

The upcoming chapters are organized as follows:

- Chapter 2 provides a discussion of some of the elementary concepts of a rotation manifold, along with the distance metrics to provide foundations for some of the methods described in this thesis. The rest of the chapter is then devoted to existing algorithms for INS/GPS fusion and multiple rotation averaging problems, all of which involve rotational variables in the state estimation.
- Chapter 3 introduces a method to address the outlier-prone sensor fusion problem in the context of INS/GPS fusion. By extending pre-integration theory,

the algorithm efficiently and robustly fuses disparate sensors to function at different frequencies. This reveals the huge potential of nonlinear optimization techniques for long-term autonomous INS/GPS navigation.

- Chapter 4 contributes to the improvement of efficient and globally optimal rotation averaging algorithms. It proposes a new technique which can significantly accelerate the Lagrangian dual optimisation routine of the multiple rotation averaging problem.
- Chapter 5 presents a specialised rotation averaging algorithm to resolve a fundamental rotational ambiguity problem efficiently in marker-based SLAM. Unlike the existing approach, which relies on a heuristic criterion for disambiguation, we formalise the problem into a clique-constrained rotation averaging problem and develop a lifted algorithm for effective marker disambiguation.
- Chapter 6 concludes and discusses future work.

Chapter 2

Literature Review

This chapter surveys existing algorithms for INS/GPS fusion and multiple rotation averaging problems. As giving an exhaustive coverage of all existing works would be immensely challenging, this chapter aims to discuss the representative algorithms that are closely related to this thesis.

This chapter is organized as follows:

- Section 2.1 provides some elementary discussions of different rotation representations and their mutual relationships.
- Section 2.2 introduces the distance measures that are commonly employed in existing algorithms for rotation averaging described in Section 2.4.
- Section 2.3 reviews existing filtering methods for INS/GPS fusion problem. These include the popular standard Kalman filter and its variants, as well as more recent extensions that aim to make the filter more robust or systematically address the underlying geometric structure of the state estimation problem for better stability.
- Section 2.4 describes existing algorithms for multiple rotation averaging. These include the standard, robust, as well as more recent duality-based algorithms which aim to obtain globally optimal rotations by exploiting Lagrangian duality theory.

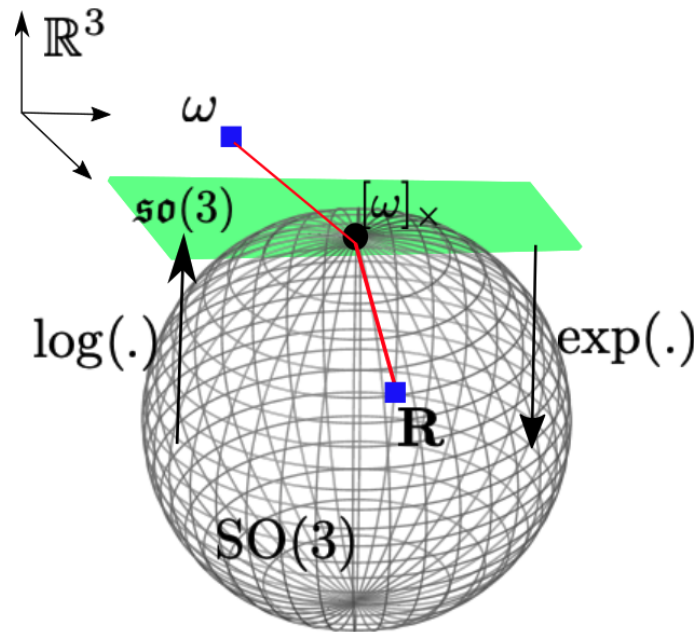


FIGURE 2.1: An intuitive way to understand the relationship between the linear tangent space $\mathfrak{so}(3)$ and the non-linear Lie group $SO(3)$

2.1 Rotation Representation

This section briefly introduces different representations of rotations, including the matrix Lie group, angle-axis and quaternion.

2.1.1 The Matrix Lie Group

Rotation space $SO(3)$ (1.12) naturally forms a matrix Lie group, which has a manifold structure. Being a manifold, $SO(3)$ inherits the properties for which the matrix multiplication and inverse operations are smoothly differentiable [5].

An appealing advantage of the manifold theory is that the *local* neighbourhood of a point on the Lie group can be sufficiently described by its associated tangent space, i.e., Lie algebra, $\mathfrak{so}(3)$. In contrast to the corresponding Lie group, dealing with the linear Lie algebra is often easier. In addition, the mappings between the Lie algebra and the Lie group can be described conveniently using the exponential and logarithm functions, respectively; see Figure 2.1.

Every rotation can be defined using the angle-axis representation to obtain $\omega = \theta \hat{\omega} \in \mathbb{R}^3$, where θ is the angle of rotation about a unit norm axis $\hat{\omega}$. We can map a vector in \mathbb{R}^3 to the space of a 3×3 skew symmetric matrix that coincides with

$\mathfrak{so}(3)$ using

$$[\boldsymbol{\omega}]_{\times} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \quad (2.1)$$

The correspondence between the Lie algebra $\mathfrak{so}(3)$ and the associated Lie Group $SO(3)$ is related by the exponential mapping as

$$\mathbf{R} = \exp([\boldsymbol{\omega}]_{\times}) : \mathfrak{so}(3) \mapsto SO(3). \quad (2.2)$$

Similarly, the inverse mapping $\mathfrak{so}(3) \mapsto SO(3)$ exists and is related by logarithm function, i.e.,

$$[\boldsymbol{\omega}]_{\times} = \log(\mathbf{R}) \quad : \quad SO(3) \mapsto \mathfrak{so}(3) \quad (2.3)$$

The exponential and logarithm functions in (2.2) and (2.3) can be computed using Rodrigues' formula [45]

$$\exp(\theta \hat{\boldsymbol{\omega}}) = \mathbf{I} + \sin(\theta)[\hat{\boldsymbol{\omega}}]_{\times} + (1 - \cos(\theta))([\hat{\boldsymbol{\omega}}]_{\times})^2. \quad (2.4)$$

$$\log(\mathbf{R}) = \frac{\theta(\mathbf{R} - \mathbf{R}^T)}{2 \sin(\theta)}, \quad (2.5)$$

where $\theta = \cos^{-1}\left(\frac{\text{tr}(\mathbf{R})-1}{2}\right)$.

Note that the exponential map (2.2) and logarithmic map (2.3) are often conveyed with some abuse of notation. Specifically, $\boldsymbol{\omega} \in \mathbb{R}^3$ is confounded with $[\boldsymbol{\omega}]_{\times} \in \mathfrak{so}(3)$. For clarity, we define $\mathbb{R}^3 \mapsto SO(3)$ with a capitalised Exp such that

$$\mathbf{R} = \text{Exp}(\boldsymbol{\omega}) : \mathbb{R}^3 \mapsto SO(3), \quad (2.6)$$

where $\text{Exp}(\boldsymbol{\omega}) \triangleq \exp([\boldsymbol{\omega}]_{\times})$; we use similar notation for the logarithmic mapping.

The Lie bracket The Lie algebra $\mathfrak{so}(3)$, which is the tangent space at the identity element of a Lie-group, is equipped with a Lie bracket. Where $X, Y \in \mathfrak{so}(3)$ and

$\lambda \in \mathbb{R}$, the Lie bracket is defined as

$$[X, Y] = XY - YX \quad (2.7)$$

which satisfies

- Anti-commutativity, $[X, Y] = -[Y, X]$.
- Bilinearity, $[\lambda X, Y] = [X, \lambda Y] = \lambda[X, Y]$
- Jacobi identity, $[X, [Y, Z]] + [Z, [X, Y]] + [Y, [Z, X]] = 0$.

Baker-Campbell-Hausdorff The Lie bracket is particularly useful for concatenating non-infinitesimal elements of Lie algebra. Given $X, Y \in \mathfrak{so}(3)$, the usual exponential relationship where $\exp^X \exp^Y = \exp^{X+Y}$ does not hold. Instead, the mapping is defined by BCH [43] as

$$\exp^X \exp^Y = \exp^{BCH(X,Y)}, \quad (2.8)$$

where $BCH(.,.)$ is defined by the Baker-Campbell-Hausdorff series

$$BCH(X, Y) = X + Y + \frac{1}{2}[X, Y] + \frac{1}{12}[X, [X, Y]] + \frac{1}{12}[[X, Y], Y] + \dots, \quad (2.9)$$

and "..." implies higher order terms.

2.1.2 Quaternion

A rotation \mathbf{R} can be parameterized in terms of a unit quaternion \mathbf{q} , which has the form

$$\mathbf{q} = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}, \quad (2.10)$$

where $a, b, c, d \in \mathbb{R}$ and $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are the fundamental *quaternion units* [6].

Let $\hat{\boldsymbol{\omega}} = [\hat{\omega}_x, \hat{\omega}_y, \hat{\omega}_z]^T$ be the unit vector axis of \mathbf{R} and the angle θ , the unit quaternion \mathbf{q} is expressed formally as

$$\mathbf{q} = \cos(\theta/2) + (\hat{\omega}_x\mathbf{i} + \hat{\omega}_y\mathbf{j} + \hat{\omega}_z\mathbf{k}) \sin(\theta/2), \quad (2.11)$$

whose norm must satisfy the unit length, i.e. $\|\mathbf{q}\|_2 = 1$. Note that both \mathbf{q} and $-\mathbf{q}$ constitute the same rotation \mathbf{R} .

2.2 Distance Metrics on $\text{SO}(3)$

Here, we show some common choices of bi-invariant distance metrics $d(.,.)$ for the cost function, such that they satisfy

$$d(\mathbf{R}_1, \mathbf{R}_2) = d(\mathbf{T}\mathbf{R}_1, \mathbf{T}\mathbf{R}_2), \quad (2.12)$$

for all rotations $\mathbf{T}, \mathbf{R}_i \in \text{SO}(3)$.

2.2.1 Angular Distance / Geodesic Distance

Any rotation in $\text{SO}(3)$ can be defined using the angle-axis representation, i.e., a rotation through an angle θ about an axis. Naturally, given two rotations \mathbf{R}_1 and $\mathbf{R}_2 \in \text{SO}(3)$, we can establish their distance as the angular distance, i.e., the angle of the relative rotation between them. Therefore, the angular distance $d_{\angle}(\mathbf{R}_1, \mathbf{R}_2)$ is defined as

$$d_{\angle}(\mathbf{R}_1, \mathbf{R}_2) = \|\log(\mathbf{R}_1\mathbf{R}_2^T)\|_2, \quad (2.13)$$

where $\|\cdot\|_2$ is the Euclidean norm of the vector. By definition, the rotation angle between \mathbf{R}_1 and \mathbf{R}_2 lies in the range $[0, \pi]$.

2.2.2 Chordal Distance

Another commonly used distance metric is the *chordal distance* $d_{chord}(.,.)$, which is derived as the Euclidean distance between two rotations in the embedding space \mathbb{R}^9 . The chordal distance between two rotations $\mathbf{R}_1, \mathbf{R}_2 \in \text{SO}(3)$ is equal to

$$d_{chord}(\mathbf{R}_1, \mathbf{R}_2) = \|\mathbf{R}_1 - \mathbf{R}_2\|_F, \quad (2.14)$$

where $\|\cdot\|_F$ is the Frobenius norm of the matrix. Let the geodesic distance between $\mathbf{R}_1, \mathbf{R}_2$ be noted as θ , the chordal distance is related to the geodesic distance defined

in Section 2.2.1 by

$$\|\mathbf{R}_1 - \mathbf{R}_2\|_F = 2\sqrt{2} \sin \frac{\theta}{2}. \quad (2.15)$$

2.2.3 Quaternion Distance

The quaternion metric d_{quat} between \mathbf{R}_1 and \mathbf{R}_2 is defined as

$$d_{quat}(\mathbf{R}_1, \mathbf{R}_2) = \min\{\|\mathbf{q}_1 - \mathbf{q}_2\|_2, \|\mathbf{q}_1 + \mathbf{q}_2\|_2\}, \quad (2.16)$$

where \mathbf{q}_1 and \mathbf{q}_2 are the quaternion representation of \mathbf{R}_1 and \mathbf{R}_2 , respectively, and the norm $\|\cdot\|$ is the Euclidean norm $\in \mathbb{R}^4$. Let the geodesic distance between $\mathbf{R}_1, \mathbf{R}_2$ be noted as θ , the quaternion distance can be related to the geodesic distance defined in Section 2.2.1 by

$$d_{quat}(\mathbf{R}_1, \mathbf{R}_2) = 2 \sin\left(\frac{\theta}{4}\right). \quad (2.17)$$

2.3 Algorithms for the INS/GPS fusion

In this section, we describe filtering methods for the INS/GPS fusion problem. We begin with the classic KF, which is the main driver that led to the widespread deployment of hybrid inertial navigation systems in the control literature. The rest of this section discusses the different variants or extensions of KF.

2.3.1 Standard filter

2.3.1.1 KF

A KF operates by combining a *state-transition model*, which describes the dynamic behavior of the state and a *measurement model*, which relates the states to the observed measurement to find the state estimates. The idea of KF is to consider both uncertainties of the models due to the inaccurate model assumption and noisy observations for the best state estimates.

Algorithm 1 Kalman filter.

Require: $\hat{\mathbf{x}}_{t-1}$, $\hat{\Sigma}_{t-1}$, \mathbf{u}_t and \mathbf{z}_t .

- 1: **Predict**
 - 2: $\bar{\mathbf{x}}_t = \mathbf{A}_t \hat{\mathbf{x}}_{t-1} + \mathbf{B}_t \mathbf{u}_t$
 - 3: $\bar{\Sigma}_t = \mathbf{A}_t \hat{\Sigma}_{t-1} \mathbf{A}_t^T + \mathbf{W}_t$
 - 4: **Update**
 - 5: $\tilde{\mathbf{y}}_t = \mathbf{z}_t - \mathbf{C}_t \bar{\mathbf{x}}_t$
 - 6: $\mathbf{S}_t = \mathbf{C}_t \bar{\Sigma}_t \mathbf{C}_t^T + \mathbf{Q}_t$
 - 7: $\mathbf{K}_t = \bar{\Sigma}_t \mathbf{C}_t^T \mathbf{S}_t^{-1}$
 - 8: $\hat{\mathbf{x}}_t = \bar{\mathbf{x}}_t + \mathbf{K}_t \tilde{\mathbf{y}}_t$
 - 9: $\hat{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t \mathbf{C}_t) \bar{\Sigma}_t$
 - 10: **return** $\hat{\mathbf{x}}_t, \hat{\Sigma}_t$.
-

Let the underlying dynamic system model be

$$\mathbf{x}_t = \mathbf{A}_t \mathbf{x}_{t-1} + \mathbf{B}_t \mathbf{u}_t + \epsilon_t \quad (2.18)$$

where $\mathbf{A}_t \in \mathbf{R}^{n \times n}$ is the state-transition matrix, $\mathbf{B}_t \in \mathbf{R}^{n \times m}$ is the control-input matrix, ϵ_t is the process noise, which assumes a zero mean and covariance \mathbf{W}_t ; and the measurement model be

$$\mathbf{z}_t = \mathbf{C}_t \mathbf{x}_t + \delta_t \quad (2.19)$$

where $\mathbf{C}_t \in \mathbf{R}^{k \times n}$ is the observation matrix, δ_t is the observation noise, which assumes a zero mean and covariance \mathbf{Q}_t .

Here, we will briefly describe the KF algorithm, summarised in Algorithm 1. To compute the mean $\hat{\mathbf{x}}_t$ and covariance $\hat{\Sigma}_t$, KF uses a two-step procedure: *predict* and *update*. During the prediction step, Lines 2 and 3 estimate the current states along with the covariance without incorporating measurement \mathbf{z}_t . A Kalman gain \mathbf{K}_t , which indicates the relative weight given to the current state estimates and the measurement, is then computed in Line 7. Lines 8 and 9 then update the posterior states and covariance ($\hat{\mathbf{x}}_t, \hat{\Sigma}_t$) based on the Kalman gain \mathbf{K}_t and the deviation between the actual measurements \mathbf{z}_t and the predicted measurement (2.19).

However, the basic KF is restricted to a linear assumption, which is unsuitable for nonlinear INS/GPS fusion problem.

Algorithm 2 Extended Kalman filter.

Require: $\hat{\mathbf{x}}_{t-1}$, $\hat{\Sigma}_{t-1}$, \mathbf{u}_t and \mathbf{z}_t .

- 1: **Predict**
 - 2: $\bar{\mathbf{x}}_t = f(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_t)$
 - 3: $\bar{\Sigma}_t = \mathbf{F}_t \hat{\Sigma}_{t-1} \mathbf{F}_t^T + \mathbf{W}_t$
 - 4: **Update**
 - 5: $\tilde{\mathbf{y}}_t = \mathbf{z}_t - h(\bar{\mathbf{x}}_t)$
 - 6: $\mathbf{S}_t = \mathbf{H}_t \bar{\Sigma}_t \mathbf{H}_t^T + \mathbf{Q}_t$
 - 7: $\mathbf{K}_t = \bar{\Sigma}_t \mathbf{H}_t^T \mathbf{S}_t^{-1}$
 - 8: $\hat{\mathbf{x}}_t = \bar{\mathbf{x}}_t + \mathbf{K}_t \tilde{\mathbf{y}}_t$
 - 9: $\hat{\Sigma}_t = (I - \mathbf{K}_t \mathbf{H}_t) \bar{\Sigma}_t$
 - 10: **return** $\hat{\mathbf{x}}_t, \hat{\Sigma}_t$.
-

2.3.1.2 EKF

The EKF extends the KF defined for a linear state-transition model (2.18) and an observation model (2.19) to the case of nonlinear functions as

$$\begin{aligned} \mathbf{x}_t &= f(\mathbf{x}_{t-1}, \mathbf{u}_t) + \epsilon_t \\ \mathbf{z}_t &= h(\mathbf{x}_t) + \delta_t, \end{aligned} \tag{2.20}$$

where $f(\cdot)$ and $h(\cdot)$ can cater for differentiable functions. Consequently, f and h can no longer be applied directly to the covariance. Therefore, the key idea of EKF is to linearize about the current state estimates through the first-order Taylor expansion of the nonlinear functions f and h .

Algorithm 2 summarizes the EKF algorithm. Observe that Lines 2 and 5 are substituted by their nonlinear generalizations. To define the state-transition and observation matrices, the EKF computes the Jacobians \mathbf{F}_t and \mathbf{H}_t as

$$\mathbf{F}_t = \frac{\delta f(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_t)}{\delta \mathbf{x}_{t-1}} \tag{2.21}$$

$$\mathbf{H}_t = \frac{\delta h(\bar{\mathbf{x}}_t)}{\delta \mathbf{x}_t}. \tag{2.22}$$

EKF has two main drawbacks. First, unlike classic KF, EKF does not guarantee optimality and risks to divergence, especially when the initial state estimates are incorrect due to its linearization [11, 29]. We refer readers to [51] for a detailed discussion of EKF's convergence. Second, since EKF operates on the assumption that

the noise follows a gaussian distribution, the accuracy of EKF's state estimations can be severely hampered by outliers.

2.3.2 Robust Filtering

As mentioned in the previous section, a major weakness of the standard EKF approach is being vulnerable to outliers. A robust mechanism is usually applied to the standard KF/EKF to mitigate the effect of outliers. Commonly used techniques include ad-hoc practices [62, 2], alternative noise models [55], weighted-based filtering [67], to name a few.

2.3.2.1 Ad-hoc methods

The ad-hoc method is one of the most widely used techniques for making EKF robust. The method is simple: discard any observations that differ from the predicted value by a predefined threshold. For instance, the posteriori state estimate $\hat{\mathbf{x}}_t$ would not be updated using observations \mathbf{z}_t (skip Lines 7-9 in Algorithm 2) if the ratio of the innovation term $\tilde{\mathbf{y}}_t$ that describes the deviation between the predicted value and observation (Algorithm 2 Line 5) and the innovation covariance \mathbf{S}_t (Algorithm 2 Line 6)

$$\mathbf{y}_t^T \mathbf{S}_t^{-1} \mathbf{y}_t > \beta, \quad (2.23)$$

exceeds a positive threshold β .

This simple heuristic works reasonably well and does not add any additional computational cost. Without the need to substantially modify the standard EKF, many practitioners often employ the ad-hoc strategy to robustify EKF for INS/GPS fusion applications in practice [2]. Although appealing, this ad-hoc practice has two main drawbacks. First, there is no theoretical justification for the choice of the thresholds β in (2.23) (typically two standard deviations are used). Second, such a heuristic is prone to false negatives, which can lead to a false build-up of estimation variances and eventually poor state estimates; see Chapter 3.

Algorithm 3 Weighted-based Extended Kalman filter.

Require: $\hat{\mathbf{x}}_{t-1}$, $\hat{\Sigma}_{t-1}$, \mathbf{u}_t and \mathbf{z}_t .

- 1: **Predict**
 - 2: $\bar{\mathbf{x}}_t = f(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_t)$
 - 3: $\bar{\Sigma}_t = \mathbf{W}_t$
 - 4: **Update**
 - 5: $\tilde{\mathbf{y}}_t = \mathbf{z}_t - h(\bar{\mathbf{x}}_t)$
 - 6: $\mathbf{S}_t = (\mathbf{H}_t \bar{\Sigma}_t \mathbf{H}_t^T + \frac{1}{\omega_t} \mathbf{Q}_t)^{-1}$
 - 7: $\mathbf{K}_t = \bar{\Sigma}_t \mathbf{H}_t^T \mathbf{S}_t$
 - 8: $\hat{\mathbf{x}}_t = \bar{\mathbf{x}}_t + \mathbf{K}_t \tilde{\mathbf{y}}_t$
 - 9: $\hat{\Sigma}_t = (I - \mathbf{K}_t \mathbf{H}_t) \bar{\Sigma}_t$
 - 10: **return** $\hat{\mathbf{x}}_t, \hat{\Sigma}_t$.
-

2.3.2.2 Weighted-based filtering methods

Inspired by weighted least squares, researchers have devised weighted-based algorithms for filtering. The underlying principle of such algorithms is to associate each observation with a weight that determines its contribution to the state estimates [26]. Different strategies have been proposed to model the underlying weight functions, such as the Huber function [47] or some heuristic functions.

A representative method under this paradigm is [67], which employs a Bayesian approach to learn the weighting function. The aim of this technique is to track the outliers in the observed data: for each observation \mathbf{z}_t , its variance is associated with a scalar weight w_t , which can be computed as

$$w_t = \frac{\alpha_{w_t} + \frac{1}{2}}{\beta_{w_t} + \tilde{\mathbf{y}}_t^T \mathbf{Q}^{-1} \tilde{\mathbf{y}}_t}, \quad (2.24)$$

whose distribution is defined as gamma-distributed such that $w_t \sim \Gamma(\alpha_{w_t}, \beta_{w_t})$. Algorithm 3 outlines the method.

Close examination shows that there are two subtle differences between this modified EKF and the standard EKF. First, the covariance Σ_t is intrinsically dependent on the previous states covariance Σ_{t-1} through \mathbf{K}_t and \mathbf{H}_t . Second, \mathbf{Q}_t is weighted; see Algorithm 3 Line 6. It should be noted that if the innovation term $\tilde{\mathbf{y}}_t$ in (2.24) is huge, the resulting term will tend to be very small. This evokes a cascading effect, which will result in a small innovation covariance \mathbf{S}_t , leading to a small Kalman gain \mathbf{K}_t : a low Kalman gain implies that the filter is more certain about the predicted

state, with less weight being given to the observation \mathbf{z}_t for computation of the posteriori state estimates.

2.3.3 Filtering on Manifolds

As mentioned in Section 1.1, state estimation problems often involve estimating states which naturally evolve on the manifold. Recently, growing attention has been paid to systematically designing KF that function intrinsically on Lie groups. The goal is to treat the underlying geometry of the manifold in a principled manner to enhance the convergence and stability of the filter [19].

Exploiting the Lie group theories, a special class of symmetry-preserving EKF has been proposed. [18, 17, 13, 14] formalise the state dynamics and measurement model built upon the concentrated Gaussian on Lie groups. Such a formulation not only properly addresses the natural symmetries of the considered model, but also provides a geometrically-meaningful covariance representation.

Although these filters have stronger stability properties and appealing theories, they still assume the state and measurement density are Gaussian distributed and therefore are highly susceptible to outliers.

2.4 Algorithms for Multiple Rotation Averaging

In this section, we describe optimization algorithms for multiple rotation averaging, considering different rotation representations and distance metrics; see Section 2.1 and 2.2.

2.4.1 Extrinsic-averaging-based algorithms

Most early works on rotation averaging are extrinsic-averaging-based algorithms [40, 54, 45]. As its name (*extrinsic*) suggests, these algorithms do not perform averaging directly on the rotation space. This section introduces two least squares algorithms for *quaternion relaxation* and *chordal relaxation*.

2.4.1.1 Quaternion Relaxation

Representing the rotations as quaternions q_i, q_j and $q_{i,j}$, we can rewrite the compatibility constraint (1.19) in the quaternion form as

$$\mathbf{q}_{i,j}\mathbf{q}_i - \mathbf{q}_j = 0. \quad (2.25)$$

Generally, quaternions are represented as $\mathbf{q} = [q_w, q_x, q_y, q_z]$, where q_w is the real part of the quaternion, and q_x, q_y, q_z are purely imaginary components; see Section 2.1.2. Applying quaternion multiplication, (2.25) gives rise to

$$\begin{bmatrix} q_w^{i,j} & -q_x^{i,j} & -q_y^{i,j} & -q_z^{i,j} \\ q_x^{i,j} & q_w^{i,j} & -q_z^{i,j} & q_y^{i,j} \\ q_y^{i,j} & q_z^{i,j} & q_w^{i,j} & -q_x^{i,j} \\ q_z^{i,j} & -q_y^{i,j} & q_x^{i,j} & q_w^{i,j} \end{bmatrix} \begin{bmatrix} q_w^i \\ q_x^i \\ q_y^i \\ q_z^i \end{bmatrix} - \begin{bmatrix} q_w^j \\ q_x^j \\ q_y^j \\ q_z^j \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (2.26)$$

Such a quaternion parameterisation can allow a linear least squares formulation and was claimed to be optimal under the assumption of Gaussian noise [40]. However, a drawback of formulation (2.25) is that it ignores the orthogonality constraint of a rotation matrix, i.e., the solution does not have a unit norm, thus they are not valid rotations.

2.4.1.2 Chordal Relaxation

We first define the rotation averaging problem in the chordal distance as

$$\min_{\{\mathbf{R}_i\}_{i=1}^N \in \text{SO}(3)} \sum_{(i,j) \in \mathcal{E}} \|\mathbf{R}_{i,j} - \mathbf{R}_j \mathbf{R}_i^T\|_F^2. \quad (2.27)$$

An alternative way is to first solve an unconstrained version of (2.27) (ignoring the $\text{SO}(3)$ constraint), and then approximate the solution to the nearest orthonormal matrix using Singular Value Decomposition (SVD) [54]. The idea behind Chordal Relaxation is similar to [40] in Section 2.4.1.1, except that the rotations are parameterised as rotation matrices

$$\min_{\{\mathbf{r}_i^k\}_{i=1}^N} \sum_{(i,j) \in \mathcal{E}} \sum_{k=1}^3 \|\mathbf{R}_{i,j} \mathbf{r}_i^k - \mathbf{r}_j^k\|^2, \quad (2.28)$$

where $\mathbf{r}_i^k \in \mathbb{R}^3$ is the k -th column of \mathbf{R}_i . While it is demonstrated in [54] that searching in the (approximate) rotation space is easier than in quaternion space, the estimated solution from (2.28) is not necessarily a valid rotation before manifold projection.

2.4.2 Intrinsic-averaging-based algorithm

Although the algorithms in Section 2.4.1 demonstrate reasonably good results in practice, they do not properly address the manifold structure, which can lead to non-optimal solutions. To establish a well-defined cost function, several descent type algorithms [28, 27, 44] that exploit the Lie Group theory to optimise directly on the rotation manifold are proposed. Their underlying principles are essentially: transverse from the nonlinear rotation manifold to the tangent space centred at the current estimate – find the next estimates and update at the tangent space – transition back to the rotation manifold.

Formally, let us define the rotation averaging problem in the geodesic metric as

$$\mathbf{R}_\mathcal{V} = \min_{\{\mathbf{R}_i\}_{i=1}^N \in SO(3)} \sum_{(i,j) \in \mathcal{E}} \rho(d_\angle(\mathbf{R}_{i,j}, \mathbf{R}_j \mathbf{R}_i^T)) \quad (2.29)$$

$$= \min_{\{\mathbf{R}_i\}_{i=1}^N \in SO(3)} \sum_{(i,j) \in \mathcal{E}} \rho(\|\text{Log}(\mathbf{R}_j^T \mathbf{R}_{i,j} \mathbf{R}_i)\|) \quad (2.30)$$

$$= \min_{\{\mathbf{R}_i\}_{i=1}^N \in SO(3)} \sum_{(i,j) \in \mathcal{E}} \rho(\|\text{Log}(\Delta \mathbf{R}_{i,j})\|), \quad (2.31)$$

where $\rho(\cdot)$ is a robust loss function.

The aim of a descent type algorithm is to compute a descent direction $\Delta \mathbf{R}_\mathcal{V}$ to update the absolute rotations $\mathbf{R}_\mathcal{V}$, which will decrease the objective value of (2.29) in each iteration t . Without loss of generality, let this update be $\{\Delta \mathbf{R}_i^{(t)}\}_{i=1}^N$, i.e., the updated estimation $\mathbf{R}_\mathcal{V}^{(t+1)} = \{\mathbf{R}_1^{(t)} \Delta \mathbf{R}_1^{(t)}, \dots, \mathbf{R}_N^{(t)} \Delta \mathbf{R}_N^{(t)}\}$, hence each iteration aims to minimize

$$\sum_{(i,j) \in \mathcal{E}} \rho(\|\text{Log}(\Delta \mathbf{R}_j^T \Delta \mathbf{R}_{i,j} \Delta \mathbf{R}_i)\|), \quad (2.32)$$

Algorithm 4 Intrinsic-averaging-based algorithm.

Require: $\{\tilde{\mathbf{R}}_{i,j}\} \in \mathcal{E}$, ϵ , maximum iterations k_{max}

- 1: **while** $\|\Delta\Omega_{\mathcal{V}}\| > \epsilon$ OR $k < k_{max}$ **do**
- 2: $\Delta\mathbf{R}_{i,j} \leftarrow \mathbf{R}_j^T \tilde{\mathbf{R}}_{i,j} \mathbf{R}_i$
- 3: $\Delta\omega_{i,j} \leftarrow \text{Log}(\Delta\mathbf{R}_{i,j})$
- 4: Concatenate $\Delta\omega_{i,j}$ into $\Omega_{\mathcal{E}}$ for all $(i, j) \in \mathcal{E}$
- 5: Solve $\mathbf{A}\Delta\Omega_{\mathcal{V}} = \Omega_{\mathcal{E}}$
- 6: $\mathbf{R}_i = \mathbf{R}_i \text{Exp}(\omega_i)$, $\forall i = 1, \dots, N$
- 7: $k \leftarrow k + 1$

return $\mathbf{R}_{\mathcal{V}}$

Using the axis-angle representation as described in Section 2.1.1, we define

$$\text{Exp}(\Delta\omega_{i,j}) = \Delta\mathbf{R}_{i,j}, \quad \text{Exp}(\Delta\omega_j) = \Delta\mathbf{R}_j, \quad \text{Exp}(\Delta\omega_i) = \Delta\mathbf{R}_i, \quad (2.33)$$

which yields

$$\text{Exp}(\Delta\omega_{i,j}) = \text{Exp}(\Delta\omega_j)\text{Exp}(-\Delta\omega_i) \quad (2.34)$$

Assuming $\text{Exp}(\Delta\omega_i)$ and $\text{Exp}(\Delta\omega_j)$ are close to the identity, we apply the first order approximation of Baker-Campbell-Hausdorff (BCH) to obtain

$$\Delta\omega_{i,j} = \Delta\omega_j - \Delta\omega_i. \quad (2.35)$$

Consequently, we can aggregate the relative rotation observations into

$$\mathbf{A}\Delta\Omega_{\mathcal{V}} = \Delta\Omega_{\mathcal{E}}, \quad (2.36)$$

where $\Delta\Omega_{\mathcal{V}} = [\Delta\omega_1^T, \dots, \Delta\omega_N^T]^T \in \mathbb{R}^{3N \times 1}$, $\Delta\Omega_{\mathcal{E}} \in \mathbb{R}^{3M \times 1}$ for all $(i, j) \in \mathcal{E}$ and \mathbf{A} is formed by placing \mathbf{I} and $-\mathbf{I}$ at each row for each camera edge.

A general intrinsic-based rotation averaging algorithm is outlined in Algorithm 4. In each iteration of Algorithm 4, the discrepancies between the measurements $\tilde{\mathbf{R}}_{i,j}$ and the estimated rotations \mathbf{R}_i (obtained in $\text{SO}(3)$) are computed and mapped to \mathbb{R}^3 (Line 2 - Line 3). After solving the linear equations in vector space (Line 5), each rotation is then updated by projecting the estimates back to the rotation group through the exponential mapping (Line 6).

The algorithms explained in the rest of this section share a similar spirit to Algorithm 4. What makes them essentially distinct from one another is the technique employed to solve the linear equation in Step 5 in Algorithm 4.

2.4.2.1 Least Squares

The strategy used in [41] to solve Step 5 in Algorithm 4 is Least Squares, which estimates the rotations by minimizing the sum of the squared residuals, i.e.

$$\min_{\Delta\Omega_{\mathcal{V}}} F = \min_{\Delta\Omega_{\mathcal{V}}} \|\mathbf{A}\Delta\Omega_{\mathcal{V}} - \Delta\Omega_{\mathcal{E}}\|^2, \quad (2.37)$$

whose gradient is

$$\Delta F(\Delta\Omega_{\mathcal{V}}) = \mathbf{A}^T(\mathbf{A}\Delta\Omega_{\mathcal{V}} - \Delta\Omega_{\mathcal{E}}) \quad (2.38)$$

As (2.37) is convex [59], any point $\Delta\Omega_{\mathcal{V}}$ for which $\Delta F(\Delta\Omega_{\mathcal{V}}) = 0$ is the global minimizer, which is equivalent to

$$\mathbf{A}^T(\mathbf{A}\Delta\Omega_{\mathcal{V}} - \Delta\Omega_{\mathcal{E}}) = 0 \quad (2.39)$$

Due to (2.39), $\Delta\Omega_{\mathcal{V}}$ can be computed in closed form as

$$\Delta\Omega_{\mathcal{V}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \Delta\Omega_{\mathcal{E}} \quad (2.40)$$

Minimizing (2.37) is essentially performing the maximum likelihood principle [7], which assumes Gaussian noise. However, the least squares solutions to (2.37) are not robust, meaning that a single outlier can arbitrarily bias the estimated rotations [7].

2.4.2.2 M-estimators

To estimate the rotations in a manner that is tolerant towards outliers, [28, 27] devised algorithms which employ a more robust loss function compared to the sum of the squared errors. We rewrite the optimization problem as

$$\min_{\{\mathbf{R}_i\}_{i=1}^N \in SO(3)} \sum_{(i,j) \in \mathcal{E}} \rho(d_{\mathcal{L}}(\mathbf{R}_{i,j}, \mathbf{R}_j \mathbf{R}_i^T)) \quad (2.41)$$

It is easy to see that the least squares method in Section 2.4.2.1 operates on a loss function of $\rho(x) = x^2$. Statistically, the quadratic loss function $\rho(x) = x^2$ is not robust as ρ increases quadratically and is unbounded. Intuitively, $\rho(\cdot)$ determines the influence of each measurement to the rotation estimation.

To be outlier-robust, the ρ should possess certain properties to discount the effect of outlying data; see [27] for a list of robust loss functions. An example is Huber's Loss [47]

$$\rho(x) = \begin{cases} \frac{x^2}{2} & \text{if } |x| \leq \epsilon \\ \epsilon(|x| - \frac{\epsilon}{2}) & \text{if } |x| > \epsilon. \end{cases} \quad (2.42)$$

Observe that the function exhibits quadratic growth until $|x| > \epsilon$, after which it increases linearly. Although more robust than l_2 , the influence of outlying data does not diminish completely.

Redescending M-estimators [8], which are a sub-class of M-estimators, possess a high degree of robustness; they can handle a large number (up to 50%) of outliers. For instance, Tukey's Biweight Loss

$$\rho(x) = \begin{cases} \frac{\epsilon^2}{6} \left(1 - \left(1 - \left(\frac{x}{\epsilon} \right)^2 \right)^3 \right) & \text{if } |x| \leq \epsilon \\ 0 & \text{if } |x| > \epsilon. \end{cases} \quad (2.43)$$

In contrast to Huber's Loss (2.42), Tukey's Biweight function (2.43) remains constant for large residuals.

Inspired by this line of work, Chapter 3 demonstrates an M-estimator optimisation framework to mitigate the practically important outliers in the INS/GPS sensor fusion problem.

2.4.2.3 Weighted Least Squares

To minimise the robust cost function (2.41), [28, 27] devised an *Iteratively Reweighted Least Squares* (IRLS) rotation averaging algorithm. We rewrite (2.32) in Section 2.4.2 as

$$\min_{\Delta\Omega_{\mathcal{V}} \in \mathbb{R}^{3N \times 1}} \sum_{(i,j) \in \mathcal{E}} \rho(\|\text{Log}(\Delta\mathbf{R}_j^T \Delta\mathbf{R}_{i,j} \Delta\mathbf{R}_i)\|). \quad (2.44)$$

[28, 27] reformulated the optimization problem in (2.44) to an IRLS problem as

$$\min_{\Delta\Omega_{\mathcal{V}} \in \mathbb{R}^{3N \times 1}} \sum_{(i,j) \in \mathcal{E}} \phi_{i,j} \|\mathbf{x}_{i,j}(\Delta\Omega_{\mathcal{V}})\|^2, \quad (2.45)$$

where $\mathbf{x}_{i,j}(\Delta\Omega_{\mathcal{V}}) = \text{Log}(\text{Exp}(-\Delta\boldsymbol{\omega}_j)\text{Exp}(\Delta\boldsymbol{\omega}_{i,j})\text{Exp}(\Delta\boldsymbol{\omega}_i))$ and $\phi_{i,j}(\cdot)$ denotes the weight function.

Given an initial estimate $\Delta\Omega_{\mathcal{V}}^{(0)}$ at $t = 0$, the IRLS alternates between assigning weights $\phi_{i,j}$ to each edge $(i, j) \in \mathcal{E}$ based on the current estimated rotations $\Delta\Omega_{\mathcal{V}}^{(t)}$, and updating the estimates for the next iteration $t + 1$ by solving a weighted least squares problem

$$\Delta\Omega_{\mathcal{V}} = -(\mathbf{A}^T\Phi\mathbf{A})^{-1}\mathbf{A}^T\Phi\Delta\Omega_{\mathcal{E}}, \quad (2.46)$$

where \mathbf{A} is the incidence-matrix and Φ is a diagonal matrix with the elements of $\phi_{i,j}$.

2.4.2.4 L1 Weiszfeld algorithm

The *L1 Weiszfeld* algorithm proposed by Hartley et al. [44] is a special case of (2.41) where $\rho(\mathbf{x}) = |\mathbf{x}|$.

To develop the intuition, let us consider a simple L_1 averaging problem, where $\mathcal{D} = \{a_i\}_{i=1}^n$ are n points on \mathbb{R}^n , and our interest is to find another point $b \in \mathbb{R}$, where the sum of all Euclidean distances to the a_i 's are the minimum

$$\min_{b \in \mathbb{R}} \sum_{i=1}^n \|b - a_i\|. \quad (2.47)$$

Given a current estimate $b^{(t)}$, the Weiszfeld algorithm computes the next estimate $b^{(t+1)}$ by solving

$$b^{(t+1)} = b^{(t)} + \lambda \sum_{i=1}^n \frac{a_i - b^{(t)}}{\|a_i - b^{(t)}\|}, \quad (2.48)$$

where the closed-form step size $\lambda = \sum_{i=1}^n \|a_i - b^{(t)}\|^{-1}$. Weiszfeld guarantees that a median will be achieved for $b^{(t)} \neq a_i$.

We now attempt to establish the L1 rotation averaging using the Weiszfeld algorithm for the manifold [44] for multiple rotation averaging; see Algorithm 5. Basically, Algorithm 5 estimates each geodesic median of the rotations \mathbf{R}_j in turn using a successive Weiszfeld method, which involves only the neighbouring vertices (Line 3 - 4), while the rest of the rotations remain constant. Specifically, the Weiszfeld

Algorithm 5 L1 Weiszfeld Multiple Rotation Averaging in SO(3).

Require: Initial $\{\mathbf{M}_j^{(0)}\}_{j=1}^N$, $t = 0$

- 1: **repeat**
 - 2: **for** $j = 1, \dots, N$ **do**
 - 3: $\omega_j^{(i)} \leftarrow \log(\mathbf{R}_j^{(i)}(\mathbf{M}_j^{(t)})^{-1})$, $\forall i \in \mathcal{N}(j)$, where $\mathcal{N}(j)$ is the set of vertices connected to j .
 - 4: $\delta_j^{(i)} \leftarrow \left(\sum_i \frac{\omega_j^{(i)}}{\|\omega_j^{(i)}\|} \right) / \left(\sum_i \frac{1}{\|\omega_j^{(i)}\|} \right)$, $\forall i \in \mathcal{N}(j)$
 - 5: $\mathbf{M}_j^{(t+1)} \leftarrow \mathbf{M}_j^{(t)} \text{Exp}(\delta_j^{(i)})$
 - 6: $t \leftarrow t + 1$
 - 7: **until** Convergence
-

median M_j is updated by averaging $\mathbf{R}_j^{(i)}$ derived from its neighbouring vertices i , i.e., $\{\mathbf{R}_j^{(i)} = \mathbf{R}_{i,j} \mathbf{R}_i \mid \forall i \in \mathcal{N}(j)\}$.

In contrast to L_2 [41], Weiszfeld [44] has been demonstrated to be more robust in averaging the rotations. However, Algorithm 5 does not scale well to large problems as the rotations are updated individually.

2.4.3 Preprocessing

In contrast to previous robust algorithms, which implicitly address the issue of robust estimation in the presence of outliers, this class of preprocessing algorithms identifies/removes outliers before performing L_2 averaging.

2.4.3.1 Random Sampling Method

A random sampling scheme to prune the outliers for a rotation averaging algorithm is outlined in Algorithm 6. At each iteration t , a $(N - 1)$ -minimum spanning tree \mathcal{M} is sampled from the viewgraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $N = |\mathcal{V}|$. Each vertex in \mathcal{V} represents an absolute rotation and each edge $(i, j) \in \mathcal{E}$ is the relative rotation between i and j . Then, the rotations $\mathbf{R}_{MST} = \{\mathbf{R}_1, \dots, \mathbf{R}_N\}$ are estimated from the selected minimum spanning tree \mathcal{M} . For each *model hypothesis* \mathcal{M} , $d(\mathbf{R}_{i,j}, \mathbf{R}_j \mathbf{R}_i^T)$ is evaluated on the viewgraph \mathcal{G} with \mathbf{R}_{MST} and; the hypothesis with the highest number of edges that lies within a predefined distance threshold ϵ is then returned for L_2 averaging after a given number of trials t_{max} .

Algorithm 6 A random sampling scheme for rotation averaging.

Require: Viewgraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, distance threshold ϵ , and maximum iteration t_{max}

```

1:  $\mathcal{I}^* \leftarrow 0, \mathcal{M}^* \leftarrow NULL$ 
2: for  $t = 1, \dots, t_{max}$  do
3:    $\mathcal{M} \leftarrow$  Sample a minimum spanning tree from  $\mathcal{G}$ 
4:    $\mathbf{R}_{MST} \leftarrow$  Minimal estimate from  $\mathcal{M}$ 
5:    $\tilde{\mathcal{I}} \leftarrow$  Count the number of edges that are within distance  $\epsilon$  with  $\mathbf{R}_{MST}$ 
6:   if  $\tilde{\mathcal{I}} > |\mathcal{I}^*|$  then
7:      $\mathcal{I}^* \leftarrow \tilde{\mathcal{I}}, \mathcal{M}^* \leftarrow \mathcal{M}$ 
return  $\mathcal{M}^*$ 

```

Since the optimization machinery in Algorithm 6 is random sampling, this method naturally inherits the disadvantages of RANSAC; that is, it may provide different results in different runs.

2.4.3.2 Bayesian Method

The core idea of [70] in identifying/pruning incorrect relative rotations hinges on concatenating the edges, which should yield a result close to the identity loop. [70] proposed an involved Bayesian framework to classify inliers/outliers, given the statistics collected on many loops of the viewgraph \mathcal{G} . However, this approach is computationally expensive for a large viewgraph.

2.4.4 Duality-based algorithms

While the algorithms discussed in the previous sections are efficient and/or robust, those restrictions to *local search* come at the expense of reliability, such that they do not guarantee local correctness. Owing to its non-convexity, the multiple rotation averaging cost function can have multiple local minima with costs close to the global minimum [45]; hence the descent type algorithms are susceptible to bad minima [23, 24].

Exploiting Lagrangian duality, it has been established in [38, 24] that the associated dual problem of multiple rotation averaging parameterised as either quaternions or rotation matrices, is essentially a *semidefinite programming problem* (SDP).

Formally, we define the rotation averaging problem (2.29) with the chordal distance as

$$\min_{\{\mathbf{R}_i\}_{i=1}^N \in \text{SO}(3)} \sum_{(i,j) \in \mathcal{E}} \|\mathbf{R}_{i,j} \mathbf{R}_i - \mathbf{R}_j\|_F^2. \quad (2.49)$$

Using trace notation, (2.49) can be rewritten as

$$\min_{\{\mathbf{R}_i\}_{i=1}^N \in \text{SO}(3)} - \sum_{(i,j) \in \mathcal{E}} \text{tr}(\mathbf{R}_j^T \mathbf{R}_{i,j} \mathbf{R}_i), \quad (2.50)$$

which can be further transformed into

$$\min_{\mathbf{R}} - \text{tr}(\mathbf{R}^T \tilde{\mathbf{R}} \mathbf{R}) \quad (2.51)$$

$$\text{s.t. } \mathbf{R} \in \text{SO}(3)^N. \quad (2.52)$$

where $\mathbf{R} = [\mathbf{R}_1^T, \dots, \mathbf{R}_N^T]^T$ and $\tilde{\mathbf{R}} \in \mathbb{R}^{3N \times 3N}$ symmetric matrix with upper-triangle elements (i, j) equal to $\mathbf{R}_{i,j}$ whenever $(i, j) \in \mathcal{E}$ and 0_3 otherwise. Naturally, the diagonal elements are 0_3 's. (2.51) constitutes the *primal problem*.

Observe that the rotation group $\text{SO}(3)^N$ is comprised of two types of constraints

$$\text{Orthogonality constraint : } \mathbf{R}_i^T \mathbf{R}_i = \mathbf{I}_3 \quad (2.53)$$

$$\text{Determinant constraint : } \det(\mathbf{R}_i) = 1 \quad (2.54)$$

To derive the dual problem, [25, 23, 33, 34] *relaxes* the determinant constraint (2.54) which yields

$$\min_{\mathbf{R}} - \text{tr}(\mathbf{R}^T \tilde{\mathbf{R}} \mathbf{R}) \quad (2.55)$$

$$\text{s.t. } \mathbf{R} \in O(3)^N.$$

As derived in [25, 23, 33, 34], the associated Lagrangian dual of (2.55) is the SDP relaxation

$$\min_{Z \in \mathbb{R}^{3N \times 3N}} - \text{tr}(\tilde{\mathbf{R}} Z) \quad (2.56)$$

$$\text{s.t. } Z_{i,i} = \mathbf{I}_3, \quad \forall i = 1, \dots, n \quad (2.57)$$

$$Z \succeq 0, \quad (2.58)$$

where Z is a positive-semidefinite matrix (PSD). Observe that (2.57) merely enforces the orthogonality constraint (2.53) in every diagonal block of Z .

It has been proven in [23, 33, 34] that rotation averaging satisfies strong duality under mild noise conditions, i.e., solving the semidefinite relaxed problem (2.56) is equivalent to solving the original problem (2.55). Hence, the optimiser Z^* of (2.56) is generally rank- d , which admits the factorisation

$$Z^* = \mathbf{R}^{*T} \mathbf{R}^*, \quad (2.59)$$

where $\mathbf{R}^* \in \text{SO}(d)^N$. However, the generic solvers for SDP generally do not scale well with the problem size ($N \leq 300$). A typical rotation averaging instance arising in SfM and SLAM applications, which usually deal with $N \geq 400$, are beyond the reach of these off-the-shelf solvers. Therefore, several algorithms that design specialised optimization procedures for solving relaxed SDP efficiently are proposed. Here, the Riemannian Staircase Method, Shonan and Block Coordinate Descent (BCD) are surveyed.

2.4.4.1 Riemannian Staircase Method

The major computational cost incurred in solving (2.56) using SDP solvers is due to the need to store and manipulate the *large* and *dense* PSD variable Z . To circumvent the scalability issue, [61] proposes searching through the low-rank solutions.

As established in [61], in general (even when strong duality does not hold), Z^* of (2.56) has a rank r not much greater than d , hence enabling a symmetrical rank decomposition

$$Z^* = X^{*T} X^* \quad (2.60)$$

for $X^* \in \mathbb{R}^{r \times dN}$ with $r \ll dN$.

Replacing PSD variable Z in (2.56) with its low-rank factorization $X^T X$, they first formulate a rank-restricted version of (2.56) as

$$\min_{X \in \mathbb{R}^{r \times dN}} -\text{tr}(\tilde{\mathbf{R}} X^T X) \quad (2.61)$$

$$\text{s.t.} \quad X_i^T X_i = \mathbf{I}_d, \quad \forall i = 1, \dots, n. \quad (2.62)$$

Observe that (2.61) has two outcomes.

- Since now we are solving for X which has a much lower dimensional space than Z for $r \ll dn$, the search space is reduced dramatically.
- By construction, $X^T X \succeq 0$ for all X ; thus the PSD constraint is *redundant*.

Exploiting the fact that $X_i^T X_i = I_d$, where $X_i \in \mathbb{R}^{r \times d}$ is essentially the *Stiefel manifold* [15]; see (2.64) for its definition, [61] reformulated (2.61) as a Riemannian rank-restricted problem

$$\min_{X \in \text{St}(d,r)^n} -\text{tr}(\tilde{\mathbf{R}}X^T X), \quad (2.63)$$

where

$$\text{St}(d,r) \triangleq \{X \in \mathbb{R}^{r \times d} \mid X^T X = I_d\}. \quad (2.64)$$

However, unlike the convex function in (2.56), (2.61) is a standard *nonlinear programming problem* due to the reintroduction of the *non-convex* orthogonality constraint. Nevertheless, owing to Corollary 2.1 [16], (2.63) can be solved via any (fast) local algorithm. Therefore, [61] proposes a Riemannian truncated-Newton trust-region method to solve the reduction efficiently.

Corollary 2.1 (A sufficient condition for global optimality in (2.63)). *If $X^* \in \text{St}(d,r)$ is a (row) rank-deficient 2^{nd} order critical point of (2.63), then X^* is a global minimizer of (2.63), and $Z^* = X^{*T} X^*$ is a global minimizer of (2.56).*

2.4.4.2 Shonan Method

Following a similar spirit to Riemannian's method, Shonan method [32] adapted a low-rank optimization scheme over the rotation manifold $SO(r)$ rather than dealing with the unusual Stiefel manifold. The core idea of Shonan method is to leverage existing high-performance iterative algorithms [9] tailored for rotation manifolds to solve the increasingly higher-dimensional problem (2.65)

$$\min_{Q \in \text{SO}(r)^n} \sum_{(i,j) \in \mathcal{E}} -\text{tr}(Q_j^T P \tilde{\mathbf{R}}_{i,j} P^T Q_i), \quad (2.65)$$

for $r \geq 3$. Specifically, the optimization procedure begins by solving (2.65) for $r = 3$ via local optimisation; if the solution fails the global certification mechanism, the optimisation will be lifted to the successively higher dimension r and resolve; the process will be terminated when the solution passes the verification test. Observe that the existence of the $r \times d$ projection matrix P , $P \triangleq [\mathbf{I}_d; 0]$ in (2.65), whose role is to project the problem to increasingly higher-dimensional domains $SO(r)$.

2.4.4.3 Block Coordinate Descent Method (BCD)

[33, 34] tailor a *block coordinate descent* (BCD) method to solve (2.56). For ease of exposition, we rewrite (2.56) as

$$\begin{aligned} \min_{Z \in \mathbb{R}^{3N \times 3N}} & -\text{tr}(\tilde{\mathbf{R}}Z) & (2.66) \\ \text{s.t.} & \begin{bmatrix} \mathbf{I}_3 & Z_{1,2} & Z_{1,3} & \dots & Z_{1,N} \\ Z_{2,1} & \mathbf{I}_3 & Z_{2,3} & \dots & Z_{2,N} \\ Z_{3,1} & Z_{3,2} & \mathbf{I}_3 & \dots & Z_{3,N} \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ Z_{N,1} & Z_{N,2} & Z_{N,3} & \dots & \mathbf{I}_3 \end{bmatrix} \succeq 0 & (2.67) \end{aligned}$$

At each iteration t , the BCD approach determines the k^{th} rows and columns of blocks in (2.67), then minimizes the corresponding block while fixing all other coordinates. It turns out that the resulting subproblem admits a simple closed form solution, which leads to a more efficient algorithm for (2.56) compared to the general-purpose SDP solver (SeDumi) [66] on small to moderately sized instances ($N \leq 300$).

Unfortunately, the efficiency of this approach deteriorates dramatically when the input size increases as the BCD approach needs to store and manipulate a $3N \times 3N$ dense PSD matrix. Chapter 4 proposes a novel technique that can significantly accelerate the coordinate descent method.

Chapter 3

Outlier-Robust Manifold Pre-Integration for INS/GPS Fusion

The work contained in this chapter has been published as the following paper

Shin-Fang Chng, Alireza Khosravian, Anh-Dzung Doan and Tat-Jun Chin: Outlier-Robust Manifold Pre-Integration for INS/GPS Fusion. IEEE/RSJ International-Conference on Intelligent Robots and Systems (IROS) 2019.

Statement of Authorship

Title of Paper	Outlier-Robust Manifold Pre-Integration for INS/GPS Fusion
Publication Status	<input checked="" type="checkbox"/> Published <input type="checkbox"/> Accepted for Publication <input type="checkbox"/> Submitted for Publication <input type="checkbox"/> Unpublished and Unsubmitted work written in manuscript style
Publication Details	Shin-Fang Chng, Alireza Khosravian, Anh-Dzung Doan and Tat-Jun Chin. "Outlier-Robust Manifold Pre-Integration for INS/GPS Fusion." IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2019.

Principal Author

Name of Principal Author (Candidate)	Shin-Fang Chng		
Contribution to the Paper	Proposed the main idea, conducted experiments and wrote the paper.		
Overall percentage (%)	60		
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.		
Signature		Date	6 January 2021

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

- i. the candidate's stated contribution to the publication is accurate (as detailed above);
- ii. permission is granted for the candidate to include the publication in the thesis; and
- iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Alireza Khosravian		
Contribution to the Paper	Provided major discussions and suggestions about the method and experiments. Modified the draft.		
Signature		Date	6 January 2021

Name of Co-Author	Anh-Dzung Doan		
Contribution to the Paper	Provided suggestions and proofread the draft.		
Signature		Date	6 January 2021

Name of Co-Author	Tat-Jun Chin		
Contribution to the Paper	Proposed the general research direction. Supervised the development of the work. Modified the draft.		
Signature		Date	21 Feb 2021

Outlier-Robust Manifold Pre-Integration for INS/GPS Fusion

Shin-Fang Ch'ng, Alireza Khosravian, Anh-Dzung Doan and Tat-Jun Chin

Abstract— We tackle the INS/GPS sensor fusion problem for pose estimation, particularly in the common setting where the INS components (IMU and magnetometer) function at much higher frequencies than GPS, and where the magnetometer and GPS are prone to giving erroneous measurements (outliers) due to magnetic disturbances and glitches. Our main contribution is a novel non-linear optimization framework that (1) fuses pre-integrated IMU and magnetometer measurements with GPS, in a manner that respects the manifold structure of the state space; and (2) supports the usage of robust norms and efficient large scale optimization to effectively mitigate the effects of outliers. Through extensive experiments, we demonstrate the superior accuracy and robustness of our approach over filtering methods (which are customarily applied in the target setting) with minimal impact to computational efficiency. Our work further illustrates the strength of optimization approaches in state estimation problems and paves the way for their adoption in the control and navigation communities.

I. INTRODUCTION

Pose estimation is integral to robotic navigation and control systems. Recent works and surveys suggest that this problem is a subject of active research [1]–[4]. Generally, micro-electromechanical Inertial Measurement Units (IMU) are favourable for pose estimation on robotics systems due to the IMU's low weight, power consumption, and cost. IMUs (that give angular velocity and acceleration measurements) are typically combined with 3-axis magnetometers (that give partial pose information) to realise Inertial Navigation Systems (INS) that are able to give a richer set of measurements for pose estimation. However, low cost INS suffer from high noise levels and time-varying biases. Estimating robot pose based on INS dead reckoning is thus subject to drift [5].

To mitigate INS drift, a common solution is to fuse it with a GPS navigation unit that provides velocity and position measurements [6]. However, low cost GPS units are vulnerable to glitches and measurement errors, especially in areas with poor line-of-sight to the GPS satellites [7]. In fact, magnetometer measurements can also be affected by magnetic interference arising from the robot motors or the environment, leading to erroneous measurements [8]. Hence, a significant challenge in INS/GPS fusion is to exploit the relative strengths of the sensors to mitigate drift, without being biased by measurement errors or outliers.

Many sensor fusion methods have been developed and successfully deployed in navigation systems [6], [9]–[16].

Shin-Fang Ch'ng, Alireza Khosravian, Anh-Dzung Doan and Tat-Jun Chin are with the School of Computer Science, The University of Adelaide (shinfang.chng@adelaide.edu.au, alirezakhosravian@gmail.com, dung.doan@adelaide.edu.au, tat-jun.chin@adelaide.edu.au).

This work was supported by ARC Centre of Excellence on Robotic Vision (CE140100016).

Stochastic filtering techniques, especially Extended Kalman Filtering (EKF), are arguably the most common approaches for INS/GPS fusion due to their well-understood principles [17]. However, outliers will invariably lead to poor outcomes in standard EKF, which assumes that all measurements are trustworthy [7], [18], [19]. Generally speaking, designing an EKF variant that is outlier-robust and asymptotically stable for a problem with nonlinear dynamics and a state space with a Lie group structure—characteristics of our INS/GPS fusion problem—has proven to be challenging [5].

A. Handling outliers in stochastic filtering

There have been efforts to improve the robustness of classical Kalman Filtering (KF) towards outliers. The simple and common technique of discarding any observation that differs from the predicted value by a predefined threshold [19] is prone to false negatives, which can lead to the (false) build up of estimation variance and eventually poor estimates. More principled approaches developed for outlier handling in KF, such as the usage of alternative noise models [20] and Huber technique to KF residuals [21], may negatively affect the stability of the system if directly applied to INS/GPS fusion, due to the Lie group structure of the state space [2], [11].

B. Stochastic filtering on Lie groups

Recently, there has been an attention on systematically designing KFs that function intrinsically on Lie groups [3], [11]. The aim is to properly observe the underlying symmetry of the problem to enhance the convergence and stability of the filter. These efforts have led to the development of invariant KFs [11] that exhibit stronger stability properties than ad-hoc adaptations of classical KFs, especially when applied to INS/GPS fusion. However, these invariant filters do not consider measurement outliers in their design. Also, robustifying the invariant filters via the ad-hoc or heuristic approaches alluded to above seems challenging, due to the complex design and structure of these filters.

C. Nonlinear optimization in state estimation

In a parallel development, impressive results from Visual SLAM have shown that state estimation approaches based on nonlinear optimization (specifically nonlinear least squares) consistently outperform stochastic filtering methods, given the equivalent amount of computing resources [22]–[25]. In fact, nonlinear optimization can readily be brought to bear on Lie groups, and can more conveniently attain robustness against outliers by using robust norms. Yet another advantage is the availability of “generic” open source optimization packages [26]–[28] that simplify implementation.

Unsurprisingly, enthusiasm for optimization-based approaches have begun to grow in the control community, who have traditionally used stochastic filtering approaches. The recent works [29]–[32] have in fact targeted inertial navigation applications. However, these works have not considered scenarios with outliers or have systematically handled asynchronous sensor modalities (the latter is a fundamental weakness of optimization-based state estimation approaches [25]). Techniques including downsampling/interpolation [30] and averaging [31], [32] have been adopted by previous works to tackle the latter problem. However, these relatively simple strategies to handle sensor asynchrony are problematic, e.g. downsampling discards useful information, whereas the interpolation approach is dependent on the choice of the interpolation function (e.g. piecewise constant, polynomial, linear) and characteristics of the data points. If there are outliers in the data (which often occur in practice), the interpolated data creates even more problematic data. Moreover, generating interpolated data for the slower sensor will lead to a more expensive optimization problem as more variables are required to be optimized. Also, crude averaging method ignores the manifold structure of the state space.

D. Our contributions

We develop a novel non-linear optimization technique to address the state estimation problem in the INS/GPS fusion. The primary contribution of our work is the proposal of a sliding-window optimization technique which; 1) computes an accurate 6DoF robot trajectory, 2) concurrently compensate for the inherent IMU bias, 3) correctly fuse measurements from the three complementary but asynchronous sensors (IMU, magnetometer and GPS), by adapting the pre-integration approach [25] to derive the error terms associated with IMU and magnetometer that enable them to be pre-integrated across time and in a manner than respects the Lie group structure. Also, leveraging the ability of pre-integration theory to perform recursive optimization can significantly reduce the computational complexity. Our work can be seen as an extension of the pre-integration theory for visual-inertial (camera and IMU) SLAM [25] to INS/GPS fusion.

Moreover, we also explore the usage of robust norms in nonlinear least squares to effectively handle outliers from the measurements (particularly the GPS outliers), which can be easily affected by environmental factors. Our experimental results demonstrate the superior accuracy and robustness of our method over existing filtering techniques that solve the equivalent problem, i.e., INS/GPS fusion in the absence and presence of the outliers.

Note that the works closest in spirit to ours [29]–[32] have not considered outliers or have systematically handled asynchrony in the measurements, as described in Sec. I-C.

II. PROBLEM FORMULATION

Consider a rigid body is equipped with an IMU, a GPS, and a magnetometer. The body-fixed frame coincides with the IMU frame, which is denoted by b . We denote the North-East-Down (NED) reference frame as w (the world

frame). Neglecting the effects due to the rotation of the Earth, we assume that w is an inertial frame. The following measurements are available:

- The IMU consists of a 3-axis gyro which measures the angular velocity ${}_b\omega$, and a 3-axis accelerometer that measures the specific acceleration ${}_b\mathbf{a}$. The sampling rate of IMU is denoted by f_{IMU} .
- The GPS unit measures the linear velocity ${}_w\mathbf{v}$ and position ${}_w\mathbf{p}$, sampled at rate f_{GPS} .
- The 3-axis magnetometer measures the magnetic field of the earth in the body-fixed frame. The magnetometer output, ${}_b\mathbf{m}$ provides partial information of the attitude matrix, \mathbf{R}_b^w as:

$${}_b\mathbf{m} = (\mathbf{R}_b^w)^T {}_w\mathring{\mathbf{m}}, \quad (1)$$

where ${}_w\mathring{\mathbf{m}}$ is the (approximately constant) magnetic field of the earth at the position of the rigid body expressed in the NED frame. We represent f_{Mag} as the sampling rate of magnetometer measurements.

Here we allow the sensor measurements to be asynchronous, i.e., $f_{\text{IMU}}, f_{\text{GPS}}, f_{\text{Mag}}$ can be different. By default, we assume that $f_{\text{IMU}} > f_{\text{Mag}} > f_{\text{GPS}}$, which is sensible since in most practical settings the sampling rate of IMU exceeds those of the magnetometer and GPS [30].

A. The State

Our goal is to estimate the state at time t when each GPS measurement is received up to time T . We define the state of our system as:

$$\hat{\chi}_t = (\mathbf{R}_t, \mathbf{v}_t, \mathbf{p}_t, \mathbf{b}_t), \quad (2)$$

where $(\mathbf{R}_t, \mathbf{p}_t) \in \text{SE}(3)$ is the pose of the rigid body, $\mathbf{v}_t \in \mathbb{R}^3$ is its linear velocity, and $\mathbf{b}_t \in \mathbb{R}^3$ is the (unknown) gyroscope bias. Here, we propose a non-linear least square formulation to minimize the sum of squared of all measurement residuals, as:

$$\min_{\hat{\chi}_t} \frac{1}{2} \sum_{t=1}^T \left(\|\mathbf{r}_{\text{IMU}}(\mathbf{z}_{\text{IMU}t \rightarrow t+1}, \hat{\chi}_t)\|_{\Sigma_i}^2 + \|\mathbf{r}_{\text{GPS}}(\mathbf{z}_{\text{GPS}}, \hat{\chi}_t)\|_{\Sigma_b}^2 + \|\mathbf{r}_{\text{B}}(\mathbf{z}_{\text{B}}, \hat{\chi}_t)\|_{\Sigma_c}^2 + \|\mathbf{r}_{\text{Mag}}(\mathbf{z}_{\text{Mag}}, \hat{\chi}_t)\|_{\Sigma_d}^2 \right), \quad (3)$$

where $\mathbf{r}_{\text{IMU}}(\mathbf{z}_{\text{IMU}t \rightarrow t+1}, \hat{\chi}_t)$, $\mathbf{r}_{\text{GPS}}(\mathbf{z}_{\text{GPS}}, \hat{\chi}_t)$, $\mathbf{r}_{\text{B}}(\mathbf{z}_{\text{B}}, \hat{\chi}_t)$ and $\mathbf{r}_{\text{Mag}}(\mathbf{z}_{\text{Mag}}, \hat{\chi}_t)$ correspond to residuals for IMU, GPS, IMU bias and magnetometer measurements, respectively. Detailed definition of each residual term will be presented in Sec. III-A, III-B, III-C, III-E.

B. IMU model

The IMU measures angular velocity and linear acceleration of b frame relative to w frame. We assume that raw gyroscope measurements, ${}_b\tilde{\omega}$ is affected by a slowly varying

sensor bias \mathbf{b}^g [25]:¹

$${}_b\tilde{\boldsymbol{\omega}}_n = {}_b\boldsymbol{\omega}_n + \mathbf{b}_n^g \quad (4)$$

$${}_b\tilde{\mathbf{a}}_n = \mathbf{R}_{b_n}^{wT} ({}_w\mathbf{a}_n - {}_w\mathbf{g}), \quad (5)$$

where ${}_b\boldsymbol{\omega} \in \mathbb{R}^3$ is the instantaneous angular velocity of b relative to w expressed in coordinate frame b , ${}_w\mathbf{a} \in \mathbb{R}^3$ is the instantaneous linear acceleration of b relative to w expressed in w , and ${}_w\mathbf{g}$ is the constant gravitational acceleration vector in w frame.

We employ the following continuous-time model [25]:

$${}_w\dot{\mathbf{p}} = {}_w\mathbf{v}, \quad {}_w\dot{\mathbf{v}} = {}_w\mathbf{a}, \quad \dot{\mathbf{R}}_b^w = \mathbf{R}_b^w {}_b\boldsymbol{\omega} \times, \quad (6)$$

where the operator $(\cdot)_\times$ maps a vector in \mathbb{R}^3 to its associated skew symmetric matrix in $\mathfrak{so}(3)$.

Assuming that ${}_b\boldsymbol{\omega}$ and ${}_w\mathbf{a}$ are constant between two time instants $n = i$ and $n = i + 1$, Euler integration is applied to (6) to propagate the rigid body's pose and velocity using IMU measurements, yielding:

$${}_w\mathbf{p}_{i+1} = {}_w\mathbf{p}_i + {}_w\mathbf{v}_i\Delta t + \frac{1}{2}(\mathbf{R}_{b_i}^w {}_b\tilde{\mathbf{a}}_i + {}_w\mathbf{g})\Delta t^2 \quad (7a)$$

$${}_w\mathbf{v}_{i+1} = {}_w\mathbf{v}_i + (\mathbf{R}_{b_i}^w {}_b\tilde{\mathbf{a}}_i + {}_w\mathbf{g})\Delta t \quad (7b)$$

$$\mathbf{R}_{b_{i+1}}^w = \mathbf{R}_{b_i}^w \exp\left(\left({}_b\tilde{\boldsymbol{\omega}}_i - \mathbf{b}_i^g\right)_\times \Delta t\right), \quad (7c)$$

where $\exp : \mathfrak{so}(3) \rightarrow \text{SO}(3)$. Although more sophisticated numerical integrated methods can be employed [35]–[38], our experiments suggest that the above Euler approximation performs very well for our specific application where IMU sampling rate is high [39].

C. Pre-integration of IMU on manifold

In this section, to simplify the presentation and without loss of generality, we assume $f_{\text{GPS}} = f_{\text{Mag}}$ and $f_{\text{GPS}}, f_{\text{Mag}} < f_{\text{IMU}}$. We further generalize this in Sec. III-E.

We initialize a state variable (i.e. a node in the optimization) of the form (2) each time we receive a GPS measurement. Our goal in this section is to combine all of the IMU measurements received between successive GPS measurements and generate a single pre-integrated IMU measurement. This pre-integration significantly reduces the computational complexity of the least squares problem (3) since it prevents re-incorporating all of the IMU measurements at each iteration of the least-squares problem.

Assume that two consecutive GPS measurements are received at times $t = i$ and $t = j$. We, hence, initialize two state variables (i.e. two nodes of the optimization) according to (2) at times $t = i$ and j . Inspired by [25], we summarize all the IMU measurements between the two required states $\hat{\boldsymbol{\chi}}_i$ and $\hat{\boldsymbol{\chi}}_j$ (to be estimated).

We denote the pre-integrate position, velocity, and orientation from $t = i$ to $t = j$ by $\Delta\mathbf{p}_{i \rightarrow j}^{b_i}$, $\Delta\mathbf{v}_{i \rightarrow j}^{b_i}$, $\Delta\mathbf{R}_{b_i \rightarrow j}^{b_i}$, respectively, to represent the relative motion increments between

two consecutive poses and velocities. The pre-integrated delta components are initialized as $\Delta\mathbf{p}_{i \rightarrow i}^{b_i} = 0$, $\Delta\mathbf{v}_{i \rightarrow i}^{b_i} = 0$, $\Delta\mathbf{R}_{b_i \rightarrow i}^{b_i} = \mathbf{I}$. By taking b_i as the reference frame, successive application of (7) between $t = i$ and $t = j$ yields

$$\Delta\mathbf{p}_{i \rightarrow j}^{b_i} = \sum_{t=i}^{j-1} \left[\Delta\mathbf{v}_t^{b_i} \Delta t + \frac{1}{2} \Delta\mathbf{R}_t^{b_i} (\tilde{\mathbf{a}}_t) \Delta t^2 \right] \quad (8a)$$

$$\Delta\mathbf{v}_{i \rightarrow j}^{b_i} = \sum_{t=i}^{j-1} \Delta\mathbf{R}_t^{b_i} (\tilde{\mathbf{a}}_t) \Delta t \quad (8b)$$

$$\Delta\mathbf{R}_{b_i \rightarrow j}^{b_i} = \prod_{t=i}^{j-1} \left(\exp(\tilde{\boldsymbol{\omega}}_t - \mathbf{b}_t^g)_\times \Delta t \right), \quad (8c)$$

where i is the discrete sample of one IMU measurement within $t = [i, j]$, and Δt is the time interval between two IMU measurements i and $i + 1$.

Note that (8) is now independent of the estimated states which prevents re-calculation whenever pose and velocity estimates change, except for the bias. To avoid repeating the same equations in our paper, please find the 1st order Taylor expansion presented in [25] for the recursive implementations when the bias estimate changes. We remark that adapting the pre-integration strategy [25] in tackling asynchrony sensor modalities is conceptually superior over [30]–[32].

III. MEASUREMENT RESIDUAL TERMS

In this section, we introduce our residual error terms of IMU, GPS, bias and magnetometer measurements.

A. Preintegrated IMU Factor

Given the pre-integrated measurement model in (8), we can further rewrite (7), which yields:

$$\Delta\mathbf{p}_{i \rightarrow j}^{b_i} \doteq (\mathbf{R}_{b_i}^w)^T ({}_w\mathbf{p}_j - {}_w\mathbf{p}_i - {}_w\mathbf{v}_i \Delta t_{ij} - \frac{1}{2} \mathbf{g} \Delta t_{ij}^2) \quad (9a)$$

$$\Delta\mathbf{v}_{i \rightarrow j}^{b_i} \doteq (\mathbf{R}_{b_i}^w)^T ({}_w\mathbf{v}_j - {}_w\mathbf{v}_i - \mathbf{g} \Delta t_{ij}) \quad (9b)$$

$$\Delta\mathbf{R}_{b_i \rightarrow j}^{b_i} \doteq (\mathbf{R}_{b_i}^w)^T \mathbf{R}_{b_j}^w, \quad (9c)$$

where $\Delta t_{ij} = \sum_{t=i}^j \Delta t$.

We express the residual error $\mathbf{r}_{\text{IMU}}(\mathbf{z}_{\text{IMU}_{i \rightarrow j}}, \hat{\boldsymbol{\chi}}_i) \doteq [\mathbf{e}_{\Delta\mathbf{p}_{i \rightarrow j}}, \mathbf{e}_{\Delta\mathbf{v}_{i \rightarrow j}}, \mathbf{e}_{\Delta\mathbf{R}_{i \rightarrow j}}]^T \in \mathbb{R}^9$ as:

$$\mathbf{e}_{\Delta\mathbf{p}_{i \rightarrow j}} = (\mathbf{R}_{b_i}^w)^T ({}_w\mathbf{p}_j - {}_w\mathbf{p}_i - {}_w\mathbf{v}_i \Delta t_{ij} - \frac{1}{2} \mathbf{g} \Delta t_{ij}^2) \quad (10a)$$

$$- \Delta\mathbf{p}_{i \rightarrow j}^{b_i}$$

$$\mathbf{e}_{\Delta\mathbf{v}_{i \rightarrow j}} = (\mathbf{R}_{b_i}^w)^T ({}_w\mathbf{v}_j - {}_w\mathbf{v}_i - \mathbf{g} \Delta t_{ij}) - \Delta\mathbf{v}_{i \rightarrow j}^{b_i} \quad (10b)$$

$$\mathbf{e}_{\Delta\mathbf{R}_{i \rightarrow j}} = \mathbf{q}_v \left(\mathbf{R}_{b_i}^w (\mathbf{R}_{b_j}^w)^T \Delta\mathbf{R}_{b_i \rightarrow j}^{b_i} \right), \quad (10c)$$

where the notation $\mathbf{q}_v(\mathbf{R}) \in \mathbb{R}^3$ denotes the vector part of the quaternion representation of $\mathbf{R} \in \text{SO}(3)$ [5], [40].

B. GPS measurement residual

GPS measurements, namely ${}_w\tilde{\mathbf{v}}_t$ and ${}_w\tilde{\mathbf{p}}_t$ received at time $t = i$ have direct relationship with the estimated states. Hence, we can construct the algebraic equation for the residual error $\mathbf{r}_{\text{GPS}}(\mathbf{z}_{\text{GPS}}, \hat{\boldsymbol{\chi}}_i) \doteq [\mathbf{e}_{\mathbf{v}_i}, \mathbf{e}_{\mathbf{p}_i}]^T \in \mathbb{R}^6$ at $t = i$ as:

$$\mathbf{e}_{\mathbf{v}_i} = {}_w\mathbf{v}_i - {}_w\tilde{\mathbf{v}}_i, \quad \mathbf{e}_{\mathbf{p}_i} = {}_w\mathbf{p}_i - {}_w\tilde{\mathbf{p}}_i. \quad (11)$$

¹We opt not to incorporate the accelerometer bias compensation as adding an unknown accelerometer bias to (5) (on top of the unknown gyro bias) our problem setup would introduce unobservable modes, that in turn might lead to instability/divergence of the optimization solution [33], [34]. This is of particular importance in our scenario where we consider measurement outliers in addition to the bias.

C. Bias model

Since we assume the gyro measurement in the IMU is corrupted with a slow time-varying bias, this unknown bias must be estimated and compensated to achieve asymptotically accurate estimation [2], [5]. Here, we model the bias as a "random walk", resulting from the integration of the white noise.

$$\dot{\mathbf{b}}_t^g = \eta^{bg}. \quad (12)$$

By integrating (12) over successive discrete time samples $t = [i, j]$, we can form the bias residual error term, $\mathbf{r}_B(\mathbf{z}_B, \hat{\chi}_t) \doteq \mathbf{e}_{B_i} \in \mathbb{R}^3$ as:

$$\mathbf{e}_{B_i} = \mathbf{b}_j^g - \mathbf{b}_i^g. \quad (13)$$

D. Magnetometer measurement residual

Given the magnetometer model presented in (1), we can naturally form the residual error of magnetometer measurement at time $t = i$ as:

$$\mathbf{r}_{\text{Mag}}(\mathbf{z}_{\text{Mag}}, \hat{\chi}_t) \doteq \mathbf{e}_{M_i} = {}_b\tilde{\mathbf{m}}_i - (\mathbf{R}_{b_i}^w)^T {}_w\mathring{\mathbf{m}}_i, \quad (14)$$

where $\mathbf{e}_{M_i} \in \mathbb{R}^3$.

E. Incorporating intermediate magnetometer measurements

In Sec. II-C, we assume that the sampling rate of magnetometer is the same as the sampling rate of GPS, such that $f_{\text{Mag}} = f_{\text{GPS}}$. Nevertheless, in most practical scenarios, we have $f_{\text{Mag}} > f_{\text{GPS}}$. In this section, we generalize our proposed optimization framework to allow $f_{\text{Mag}} > f_{\text{GPS}}$. Inspired by the recursive predictor theory proposed by [2, Chapter 4] that compensates delays and sampling effects in pose estimation, we propose an approach that allows the incorporation of sensory data with various sampling rates into the least-squares optimization.

Assume that two consecutive GPS measurements are received at time $t = i$ and $t = j$, and a magnetometer measurement ${}_b\tilde{\mathbf{m}}_k$ is received at the time $t = k$ where $i \leq k \leq j$. The nodes $\hat{\chi}_i$ and $\hat{\chi}_j$ exist in the optimization, but the node $\hat{\chi}_k$ does not exist because no GPS measurement is received at time k . Hence, it is not possible to use the magnetometer residual as proposed by (14). Instead, we, use (9c) to obtain $\mathbf{R}_{b_k}^w = \mathbf{R}_{b_i}^w \Delta \mathbf{R}_{b_i \rightarrow k}^{b_i}$ where $\Delta \mathbf{R}_{b_i \rightarrow k}^{b_i}$ is the pre-integrated orientation which can be computed using gyro measurements from $t = i$ to $t = k$ according to (8c). Replacing for ${}_b\mathbf{m}_k = \mathbf{R}_{b_k}^w {}_w\mathring{\mathbf{m}}_k$ and using (14), we obtain

$$\begin{aligned} \mathbf{e}_{M_k} &= {}_b\tilde{\mathbf{m}}_k - (\mathbf{R}_{b_k}^w)^T {}_w\mathring{\mathbf{m}}_k \\ &= {}_b\tilde{\mathbf{m}}_k - \left(\Delta \mathbf{R}_{b_i \rightarrow k}^{b_i} \right)^T (\mathbf{R}_{b_i}^w)^T {}_w\mathring{\mathbf{m}}_k. \end{aligned} \quad (15)$$

It is now possible to implement the residual term (15) in the least-squares to incorporate the intermittent magnetometer measurements ${}_b\tilde{\mathbf{m}}_k$. Note that the residual error (15) relies on the available state $\hat{\chi}_i$ rather than the unavailable state $\hat{\chi}_k$. A similar methodology to the approach presented above has been proposed in [41] to mitigate asynchrony between IMU and LIDAR measurements, albeit in a different problem setup to the present paper. We remark that this concept can be employed to tackle GPS measurement delay problem. For

slower GPS measurement rate, one can also consider to apply associated concept to perform the state estimation at a higher sampling rate to achieve real-time compliant applications.

IV. HANDLING OUTLIERS

In practice, sensor measurements are often corrupted by outliers. From statistical point of view, an outlier is a measurement which significantly deviates from other candidates of the distribution in which it is sampled. Realistically, outliers are often derived from unmodeled factors or bizarre causes, such as temporary sensor failure, erroneous measurements or transient environment disturbance.

Generally, least square function is highly vulnerable to these outliers as a single outlier can drastically pull the estimation arbitrarily far away from the true solution [42]. This is of particular crucial importance for the INS/GPS fusion since high amplitude GPS glitches can often occur in practice, e.g. due to blockage of signals or multi-path. Also, sudden magnetic disturbance may occur in aerial vehicles, e.g. while passing from the proximity of power lines, causing temporary outliers in the magnetometer readings.

Since we are targeting a setting where we have a sequence of time-dependent variables (pose, velocity, bias) to estimate, the interaction and evolution of the variables across time are vital aspects of the problem. Therefore, our approach determine the outliers by exploring the M-estimator to implicitly alleviate the influence of a sequence of potentially erroneous GPS and magnetometer measurements. Instead of minimizing the sum of squared residual, we, hence, propose the use of robust norm function $\rho(\cdot)$ in our non-linear optimization problem. Examples of such robust $\rho(\cdot)$ are l_1 , Huber and Cauchy norm [42]. Note that we robustify our non-linear optimization problem using Cauchy norm (16) which leads to (17).

$$\rho(x) = \log(1 + x). \quad (16)$$

We propose the following robust non-linear least squares function that fuse IMU, GPS and magnetometer which arrive at different rates:

$$\begin{aligned} \min_{\hat{\chi}_t} \frac{1}{2} \sum_{t=T-N}^T & \left(\|\mathbf{r}_{\text{IMU}}(\mathbf{z}_{\text{IMU}t \rightarrow t+1}, \hat{\chi}_t)\|_{\Sigma_i}^2 \right. \\ & + \rho \left(\|\mathbf{r}_{\text{GPS}}(\mathbf{z}_{\text{GPS}}, \hat{\chi}_t)\|_{\Sigma_b}^2 \right) + \|\mathbf{r}_B(\mathbf{z}_B, \hat{\chi}_t)\|_{\Sigma_c}^2 \\ & \left. + \sum_{{}_b\tilde{\mathbf{m}}_k \in \Lambda} \rho \left(\|\mathbf{r}_{\text{Mag}}(\mathbf{z}_{\text{Mag}}, \hat{\chi}_t)\|_{\Sigma_d}^2 \right) \right), \end{aligned} \quad (17)$$

where N indexes all nodes in the window and Λ denotes the set of magnetometer measurement received.

To achieve real time processing time, the proposed method optimizes over a bounded N size sliding window of recent states. Note that each term of (17) is weighed by the sensor's noise covariances matrices Σ .²

Also, note that the optimization problem (17) can be solved via generic least square solvers [26]–[28]. In Section

²In the case of IMU, readers can find the derivation of the pre-integrated covariance in [25].

V, we demonstrate that the above mentioned robustification successfully removes GPS outliers in real scenarios.

V. EXPERIMENTAL RESULTS

This section presents a number of experimental results to compare our proposed robust state estimation approach against a popular EKF implementation for Unmanned Aerial Vehicles (UAV), namely Autopilot [43], as the baseline. Owing to the fact that Autopilot has a large community of users including researchers, ordinary and commercial consumers, we regard this baseline as the current industrial state-of-the-art. The Autopilot EKF is designed with a threshold based outlier rejection. The strategy in the EKF is to use the ratio of the norm of the EKF innovation term to the observation variance to determine if the candidate observation is within a predefined confidence interval.

Besides that, we also regard [30]–[32] as the baseline methods. Since none of these works have considered scenarios involving measurement anomalies, we examine the sensitivity of their approaches (i.e., standard non-linear least squares) towards outliers. Also, as explained in Section I, their strategies in handling sensor asynchrony have fundamental weaknesses (vulnerable to wrong interpolation, a much larger set of variables to optimize), therefore, this aspect of their work is not tested in our experiments.

Since there is no openly available dataset that contains both an accurate (independently measured) ground truth information and all the sensory data that we require, i.e. IMU, magnetometer, and GPS, we provide two sets of experiments each aiming at illustrating different aspects of the comparison.

The first set of experiment is performed on the EuRoC Dataset [44]. The dataset is recorded indoor with a Micro Aerial Vehicle equipped with a low cost MEMS IMU. Corresponding 6D ground truth poses are provided by a Vicon system. Large IMU biases are observed in these datasets. The purpose of this experiment is to compare the performance of our proposed approach with the existing filtering method in a controlled environment where ground truth is available. The disadvantage of this dataset is that it does not contain real magnetometer (presumably, due to high magnetic disturbances indoor) and GPS measurements. To address this problem, we synthetically generate magnetometer and GPS measurements corresponding to the datasets, using the available data.

The second set of experiment is performed on real flight data using onboard sensory data log of actual autonomous flights performed outdoor. This dataset contains all of the required sensory data, including the magnetometer, but does not include an independently measured ground truth information (as it is outdoor). Despite the lack of ground truth to evaluate absolute accuracy, this dataset permits a qualitative comparison. Levenberg-Marquadt algorithm is applied to solve the nonlinear optimization problem (17). In all of our experiments, we use Ceres Solver [28].

A. Initialisation

For our proposed method, we assume no prior information is available about the states and we initialise every new state to the origin, i.e. $\mathbf{R}_{b_0}^w = \mathbf{I}$, ${}^w\mathbf{v}_0 = [0, 0, 0]^T$, ${}^w\mathbf{p}_0 = [0, 0, 0]^T$ and $\mathbf{b}_0^g = [0, 0, 0]^T$. A more sophisticated initialisation could be employed, but, we try to consider the worst case scenario for our method. For the EKF, however, we initialise the pose and velocity to the ground truth, but we initialise the unknown bias to zero. Even though such setting gives an advantage to the EKF, this has been chosen intentionally to prevent EKF from divergence. Also, this highlights that our least squares approach is far more robust and does not necessarily require accurate initialization.

B. Size of window

We employ $N = 40$ in all of our experiments. It has been tuned carefully to achieve an optimum trade-off between the accuracy and the test time.

C. EuRoC Dataset Simulation

IMU measurements, ${}^b\tilde{\omega}$ and ${}^b\tilde{\mathbf{a}}$, are sampled at 200Hz and perturbed by an additive noise of 0.0024rad/s and 0.0283m/s² respectively in each axis. Raw GPS/barometer and magnetometer measurements log are not available in this dataset. To generate barometer and GPS data, we corrupt the ground truth velocity and position measurements with Gaussian noise. We consider zero mean Gaussian noise with a standard deviation of 0.01m, and a sampling rate of 5Hz for barometer altitude. The noise signal with a standard deviation of 0.1m/s is selected for GPS velocity and 0.1m for position NE, and they are sampled at 5Hz. To simulate magnetometer measurements, we consider the normalized reference direction $\hat{\mathbf{y}}(t) = [1, 0, 0]^T$. We use (1) to generate ideal vector measurements, which are sampled at 100Hz. Zero mean Gaussian noise with a standard deviation of 0.01 is added to each axis of the resulting vector measurement. We evaluate the results on three sequences of the EuRoC dataset; V2_01_Easy, V2_02_Med, MH_03_Med. Two experiments are conducted, i.e., one without while another with outliers.

1) *Scenario without outliers*: Fig. 1 depicts the pose, velocity and bias estimates as well as their corresponding errors of our proposed algorithm compared with EKF in sequence MH_03_Med. The translation, velocity and bias estimation errors are simply the Euclidean norm of the error between the ground truth and the corresponding estimate. The attitude estimation error corresponds to the angle of rotation of the error $\hat{\mathbf{R}}(t)\mathbf{R}(t)^T$, where $\hat{\mathbf{R}}$ is the estimated orientation and

TABLE I: RMS Error of the proposed approach and the EKF [43] on three different EuRoC Sequences

Sequence	RMSE of	Attitude (deg)	Translation (m)	Velocity (m/s)	Bias (rad/s)
V2_01_Easy	EKF	1.0729	0.3438	0.1577	0.0435
	Proposed	0.5770	0.0859	0.1249	0.0024
V2_02_Med	EKF	0.9594	0.1753	0.1267	0.0476
	Proposed	0.6976	0.0891	0.1155	0.0028
MH_03_Med	EKF	1.3631	0.1639	0.1339	0.0406
	Proposed	0.4579	0.0567	0.0707	0.0017

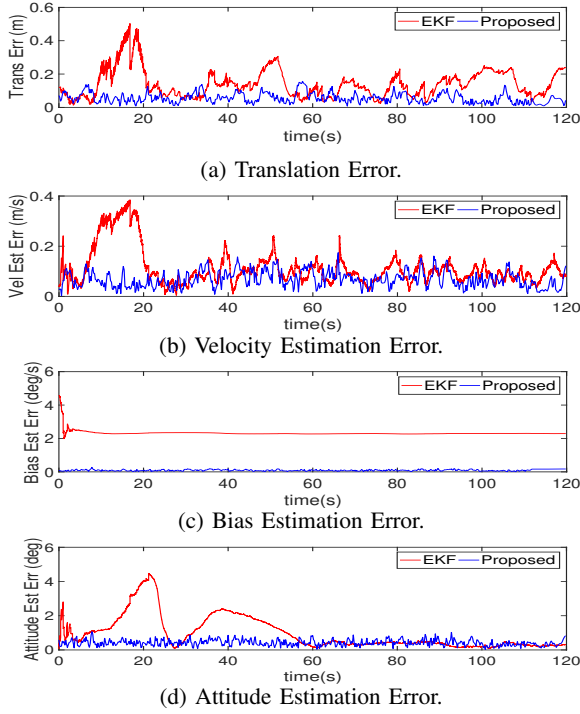
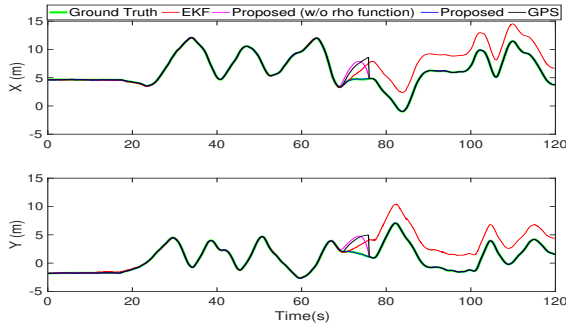
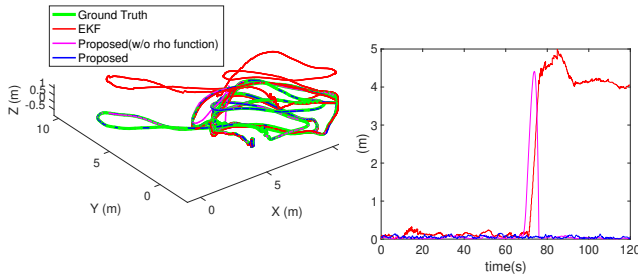


Fig. 1: MH_03_Med - Comparison between our proposed approach and the EKF.



(a) **Top:** Position X Estimates. **Down:** Position Y Estimates.



(b) **Left:** 3D view of the trajectory. **Right:** Translation Error.

Fig. 2: Seq MH_03_Med - Ground truth position XYZ vs their estimates via the EKF, the proposed method with ρ (rho) function disabled and the proposed method when measurement outliers occur from $t = 70$ s to 76 s.

\mathbf{R} is its corresponding ground truth orientation.³ The error plots show that our approach produces significantly lower errors than EKF. This is also confirmed by the rms error of the proposed approach versus EKF presented in Table I, which shows that the non-linear optimization function may offer better advantages in providing more accurate solution as computing the estimates at every iteration has the benefit of *gaining insight from a sequence of "raw" data quality* that is not possible in filtering approach.

2) *Scenario with outliers:* Fig. 2 compares the proposed approach (denoted by the blue line) with EKF when GPS position measurements are corrupted with outliers (denoted by the black line) from $t = 70$ s to $t = 76$ s. Throughout the flight of a total trajectory of 130.9m, it is observed that during the period when measurement outliers occur, our method tracks the true trajectory more accurately than EKF. In fact, the outlier identification method of EKF fails to isolate some of the outliers and the EKF position estimate (denoted by the red line) follows the black line as seen in Fig. 2a. This explains the slowly varying translation error of EKF. Also, this causes an adverse effect on EKF estimates even after $t=76$ s when there is no more outlier. The EKF wrongly rejects healthy GPS measurements after ($t=76$ s onwards) and relies mostly on dead reckoning, which leads to a significant deviation from the ground truth. In this experiment, we also assess the sensitivity of approach in [30]–[32] towards outliers by implementing (3) (denoted by the magenta line). Fig. 2 also presents a notable evidence that the resulting position estimates are distinctly biased to the measurement outliers without incorporating the robust norm function in the nonlinear least squares. Nevertheless, they still track the true pose closely once the GPS measurements become trustworthy again.

We also perform a Monte Carlo analysis with 50 simulation runs, each with randomised outlier insertion to the GPS position measurement. Fig. 3 presents a substantial evidence on the robustness of our approach as the r.m.s error averaged over 50 runs is lower compared to the EKF.

3) *Timing:* The experiment is implemented on a standard laptop (Macbook Pro, Intel i5, 2.3GHz) and is running on single core. As shown in Fig. 4, the average CPU time per window for the proposed approach over 50 Monte Carlo

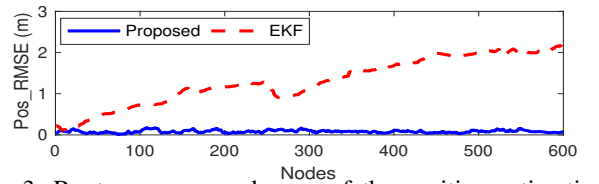


Fig. 3: Root-mean-squared error of the position estimation averaged over 50 Monte Carlo experiments with randomised outlier insertion.

³This angle is related to the Frobenius norm $\|I_3 - \hat{\mathbf{R}}\mathbf{R}^T\|_F^2 = \text{tr}((I - \hat{\mathbf{R}}\mathbf{R}^T)^T(I - \hat{\mathbf{R}}\mathbf{R}^T))$ and is given by $\hat{\theta}(t) = \frac{180}{\pi} * \text{acos}(1 - 0.25\|I - \hat{\mathbf{R}}(t)\mathbf{R}(t)^T\|_F^2)$.

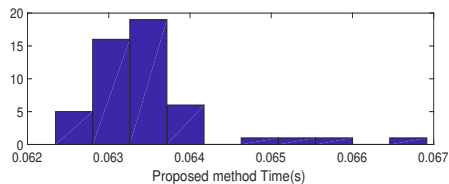


Fig. 4: Histogram plot of average CPU time per window for the proposed approach over 50 Monte Carlo runs.

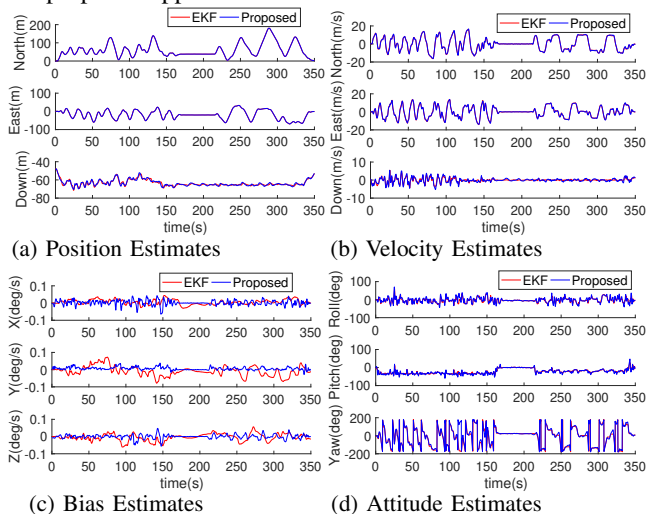


Fig. 5: Real flight dataset-The estimation results via the proposed method and EKF.

runs is approximately 60ms, which matches the real-time constraints in our problem setup.

D. PX4 flight data

The second experiment is performed on the real flight data. The dataset is recorded with a F450-Pixhawk4 that is equipped with an IMU, a magnetometer, a GPS unit and a barometer.⁴ The IMU is sampled at 250Hz while the magnetometer and GPS/barometer measurements are sampled at 50Hz and 5Hz, respectively. Again, we consider two scenarios for the real flight data as discussed in Sec. V-D.1 and V-D.2.

1) *Scenario without outliers*: As depicted in Fig. 5, the resulting state estimates of our proposed approach match very well with those of EKF. Note that there is no ground truth available in this dataset. As there is no outlier, we believe that the EKF estimates are reliable in Fig. 5.

2) *Scenario with outliers*: Fig. 6a illustrates the estimates of the proposed approach compared with EKF and (outlier-free) GPS measurements when GPS sensor fault occurs from $t = 213s$ to $220s$. Incorrect fusion of the measurement outliers (denoted by the black line) even for only a very short period of time has led to a very severe long term effect on EKF's performance. Contrarily, the resulting UAV's trajectory of our proposed approach still matches the trajectory path of (outlier-free) GPS measurements in the inset figure. Therefore, we highlight that our method demonstrate excellent

⁴The dataset is available at: https://logs.px4.io/plot_app?log=114d429c-d4f6-43e6-b3b4-740bab900d2a.

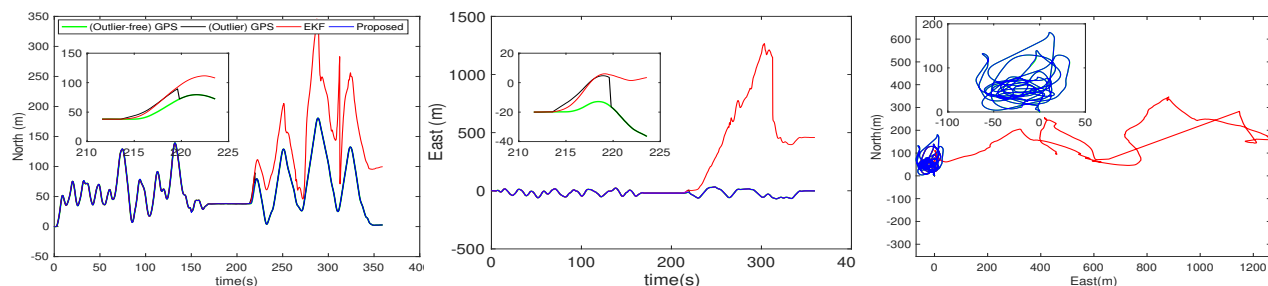
robustness and performance in mitigating outliers compared with EKF. Fig. 6b compares the resulting position estimates under scenario with and without robust $\rho(\cdot)$. We emphasise that incorporating robust norm in the nonlinear least squares for the state estimation is practically important to ensure long term autonomous navigation in a large scale environment.

VI. CONCLUSION

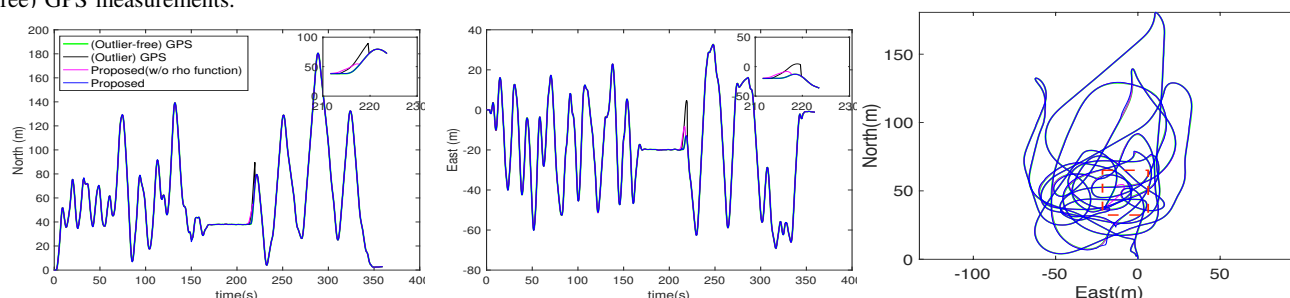
Aiming to offer a fresh insight to address the long standing pose estimation problem in INS/GPS fusion, we present a novel non-linear optimization framework to solve the equivalent problem. We extend the pre-integration technique to fuse different sensory inputs that arrive at different rates in a non-linear least squares optimisation framework. We also present a robust estimation framework to effectively mitigate the effects of practically important outlier measurements. Our experimental results demonstrate the superior accuracy and robustness of our approach over filtering methods. This further illustrate the huge potential of non-linear optimization approach in long term autonomous INS/GPS navigation.

REFERENCES

- [1] J. L. Crassidis, F. L. Markley, and Y. Cheng, "Survey of Nonlinear Attitude Estimation Methods," *Journal of guidance, control, and dynamics*, vol. 30, no. 1, pp. 12–28, 2007.
- [2] A. Khosravian, "State Estimation for Systems on Lie groups with Nonideal Measurements," Ph.D. dissertation, The Australian National University (Australia), 2016.
- [3] G. S. Chirikjian, *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*. Springer Science & Business Media, 2011, vol. 2.
- [4] S. Bonnabel, P. Martin, and P. Rouchon, "A non-linear symmetry-preserving observer for velocity-aided inertial navigation," in *American Control Conf.*, 2006, pp. 2910–2914.
- [5] R. Mahony, T. Hamel, and J.-M. Pfimlin, "Nonlinear Complementary Filters on the Special Orthogonal Group," *IEEE Trans. on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.
- [6] T. H. Bryne, J. M. Hansen, R. H. Rogne, N. Sokolova, T. I. Fossen, and T. A. Johansen, "Nonlinear Observers for Integrated INS/GNSS Navigation: Implementation Aspects," *IEEE Control Systems*, vol. 37, no. 3, pp. 59–86, 2017.
- [7] G. Agamennoni, J. I. Nieto, and E. M. Nebot, "An outlier-robust Kalman filter," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2011.
- [8] A. Ali, S. Siddharth, Z. Syed, and N. El-Sheimy, "Swarm Optimization-Based Magnetometer Calibration for Personal Handheld Devices," *Sensors*, vol. 12, no. 9, pp. 12455–12472, 2012.
- [9] H. F. Grip, T. I. Fossen, T. A. Johansen, and A. Saberi, "Globally exponentially stable attitude and gyro bias estimation with application to GNSS/INS Integration," *Automatica*, vol. 51, pp. 158–166, 2015.
- [10] A. Roberts and A. Tayebi, "On the attitude estimation of accelerating rigid-bodies using GPS and IMU measurements," in *IEEE Conf. on Decision and Control and European Control Conf. (CDC-ECC)*, 2011, pp. 8088–8093.
- [11] A. Barrau and S. Bonnabel, "The Invariant Extended Kalman Filter as a stable observer," *IEEE Trans. on Automatic Control*, vol. 62, no. 4, pp. 1797–1812, 2017.
- [12] J. F. Vasconcelos, C. Silvestre, and P. Oliveira, "A Nonlinear GPS/IMU based observer for rigid body attitude and position estimation," in *IEEE Conf. on Decision and Control (CDC)*, 2008, pp. 1255–1260.
- [13] M. Izadi, A. K. Sanyal, E. Barany, and S. P. Viswanathan, "Rigid Body Motion Estimation based on the Lagrange-d'Alembert principle," in *IEEE Conf. on Decision and Control (CDC)*, 2015, pp. 3699–3704.
- [14] H. Rehbinder and B. K. Ghosh, "Pose estimation using line-based dynamic vision and inertial sensors," *IEEE Trans. on Automatic control*, vol. 48, no. 2, pp. 186–199, 2003.
- [15] D. Senejohnny and M. Namvar, "A predictor-based attitude and position estimation for rigid bodies moving in planar space by using delayed landmark measurements," *Robotica*, vol. 35, no. 6, pp. 1415–1430, 2017.



(a) **Left, Center:** Position estimates; **Right:** Estimated trajectory; of the EKF, our proposed method with ρ (rho) function versus (outlier-free) GPS measurements.



(b) **Left, Center:** Position estimates; **Right:** Estimated trajectory; of our proposed method with and without ρ (rho) function versus (outlier-free) GPS measurements. The inset figure (**Left, Center**) shows that with the ρ (ρ)-disabled approach, the resulting position estimates are biased to the GPS outliers. Also, take note the incorrect trajectory (denoted by the square bracket) on the **Right**.

Fig. 6: Estimated Position and the corresponding trajectory when measurement outliers occur from $t = 213$ (s) to $t = 220$ (s).

- [16] R. Mahony, T. Hamel, and J.-M. Pfimlin, "Complementary filter design on the special orthogonal group $SO(3)$," in *Proc. of the IEEE Transactions on Decision and Control, CDC*, 2005.
- [17] L. A. McGee and S. F. Schmidt, "Discovery of the Kalman filter as a practical tool for aerospace and industry," NASA Ames Research Center, Tech. Rep. 86847, 1985.
- [18] M. A. Gandhi, "Robust Kalman filters using generalized maximum likelihood-type estimators," Ph.D. dissertation, Virginia Tech, 2009.
- [19] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.
- [20] J.-A. Ting, E. Theodorou, and S. Schaal, "A Kalman filter for robust outlier detection," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2007.
- [21] B. Kovačević, Ž. Đurović, and S. Glavaški, "On robust Kalman filtering," *Intl. Journal of Control*, vol. 56, no. 3, pp. 547–562, 1992.
- [22] H. Strasdat, J. Montiel, and A. J. Davison, "Real-time monocular SLAM: Why filter?" in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2010.
- [23] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *The Intl. Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.
- [24] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "An observability-constrained sliding window filter for SLAM," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2011.
- [25] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation," in *Robotics: Science and Systems*, 2015.
- [26] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A General Framework for Graph Optimization," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2011.
- [27] F. Dellaert, "Factor graphs and GTSAM: A Hands-on introduction," Georgia Institute of Technology, Tech. Rep., 2012.
- [28] S. Agarwal, K. Mierle, et al. (2012) Ceres Solver. [Online]. Available: <http://ceres-solver.org>
- [29] J. Vandersteen, M. Diehl, C. Aerts, and J. Swevers, "Spacecraft Attitude Estimation and Sensor Calibration Using Moving Horizon Estimation," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 3, pp. 734–742, 2013.
- [30] T. Polóni, B. Rohal-Ilkiv, and T. A. Johansen, "Moving Horizon Estimation for Integrated Navigation Filtering," *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 519–526, 2015.
- [31] F. Gırrbach, J. D. Hol, G. Belluscı, and M. Diehl, "Optimization-based Sensor Fusion of GNSS and IMU Using a Moving Horizon Approach," *Sensors*, vol. 17, no. 5, p. 1159, 2017.
- [32] F. Gırrbach, J. D. Hol, G. Belluscı, and M. Diehl, "Towards robust sensor fusion for state estimation in airborne applications using GNSS and IMU," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 13 264–13 269, 2017.
- [33] P. Martin and E. Salaün, "An invariant observer for earth-velocity-aided attitude heading reference systems," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 9857–9864, 2008.
- [34] P. Martin and E. Salaün, "Design and implementation of a low-cost observer-based attitude and heading reference system," *Control Engineering Practice*, vol. 18, no. 7, pp. 712–722, 2010.
- [35] P. E. Crouch and R. Grossman, "Numerical integration of ordinary differential equations on manifolds," *Journal of Nonlinear Science*, vol. 3, no. 1, pp. 1–33, 1993.
- [36] H. Munthe-Kaas, "High order runge-kutta methods on manifolds," *Applied Numerical Mathematics*, vol. 29, no. 1, pp. 115–127, 1999.
- [37] J. Park and W.-K. Chung, "Geometric integration on euclidean group with application to articulated multibody systems," *IEEE Trans. Robotics*, vol. 21, no. 5, pp. 850–863, 2005.
- [38] M. S. Andrieu and J. L. Crassidis, "Geometric integration of quaternions," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 6, pp. 1762–1767, 2013.
- [39] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual-inertial odometry," *IEEE Trans. Robotics*, vol. 33, no. 1, pp. 1–21, 2017.
- [40] A. Khosravian, J. Trumpf, R. Mahony, and C. Lageman, "Observers for invariant systems on Lie groups with biased input measurements and homogeneous outputs," *Automatica*, vol. 55, pp. 19–26, 2015.
- [41] C. Le Gentil, T. Vidal-Calleja, and H. Shoudong, "3D Lidar-IMU calibration based on upsampled preintegrated measurements for motion distortion correction," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2018.
- [42] K. Aftab and R. Hartley, "Convergence of Iteratively Re-weighted Least Squares to Robust M-Estimators," in *IEEE Conf. on Applications of Computer Vision (WACV)*, 2015, pp. 480–487.
- [43] (2018) PX4 Autopilot. [Online]. Available: <https://dev.px4.io>
- [44] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoc micro aerial vehicle datasets," *The Intl. Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

Chapter 4

Rotation Coordinate Descent for Fast Globally Optimal Rotation Averaging

The work contained in this chapter has been published as the following paper.


Alvaro Parra Bustos*, **Shin-Fang Chng***, Tat-Jun Chin, Anders Eriksson and Ian Reid: Rotation Coordinate Descent for Fast Globally Optimal Rotation Averaging.

* denotes equal contribution. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2021.

Statement of Authorship

Title of Paper	Rotation Coordinate Descent for Fast Globally Optimal Rotation Averaging
Publication Status	<input type="checkbox"/> Published <input type="checkbox"/> Accepted for Publication <input checked="" type="checkbox"/> Submitted for Publication <input type="checkbox"/> Unpublished and Unsubmitted work written in manuscript style
Publication Details	Alvaro Parra Bustos*, Shin-Fang Chng*, Tat-Jun Chin, Anders Eriksson and Ian Reid. "Rotation Coordinate Descent for Fast Globally Optimal Rotation Averaging". Submitted to Computer Pattern Recognition (CVPR) 2021. * denotes equal contribution

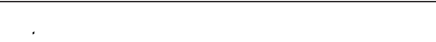
Principal Author


Name of Principal Author	Alvaro Parra Bustos		
Contribution to the Paper	Proposed the main idea. Provided major discussions about the method. Wrote the paper.		
Signature	 <table border="1" style="float: right;"> <tr> <td>Date</td> <td>6/1/2021</td> </tr> </table>	Date	6/1/2021
Date	6/1/2021		

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

- i. the candidate's stated contribution to the publication is accurate (as detailed above);
- ii. permission is granted for the candidate to include the publication in the thesis; and
- iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author (Candidate)	Shin-Fang Chng		
Contribution to the Paper	Implemented the algorithm. Conducted experiments and extensive analysis to refine the algorithm. Wrote the paper.		
Overall percentage (%)	40		
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis.		
Signature	 <table border="1" style="float: right;"> <tr> <td>Date</td> <td>6 January 2021</td> </tr> </table>	Date	6 January 2021
Date	6 January 2021		

Name of Co-Author	Tat-Jun Chin		
Contribution to the Paper	Proposed the general research direction. Supervised the development of the work. Modified the draft.		
Signature	 <table border="1" style="float: right;"> <tr> <td>Date</td> <td>21 Feb 2021</td> </tr> </table>	Date	21 Feb 2021
Date	21 Feb 2021		

Name of Co-Author	Anders Eriksson		
Contribution to the Paper	Provided discussions and suggestions about the method. Proofread the draft.		
Signature		Date	Jan 6 2020

Name of Co-Author	Ian Reid		
Contribution to the Paper	Provided suggestions and proofread the draft.		
Signature		Date	11/1/21

Rotation Coordinate Descent for Fast Globally Optimal Rotation Averaging

Alvaro Parra Bustos*
University of Adelaide

alvaro.parrabustos@adelaide.edu.au

Tat-Jun Chin
University of Adelaide

tat-jun.chin@adelaide.edu.au

Anders Eriksson
University of Queensland

a.eriksson@uq.edu.au

Shin-Fang Chng*
University of Adelaide

shinfang.chng@adelaide.edu.au

Ian Reid
University of Adelaide

ian.reid@adelaide.edu.au

Abstract

Under mild conditions on the noise level of the measurements, rotation averaging satisfies strong duality, which enables global solutions to be obtained via semidefinite programming (SDP) relaxation. However, generic solvers for SDP are rather slow in practice, even on rotation averaging instances of moderate size, thus developing specialised algorithms is vital. In this paper, we present a fast algorithm that achieves global optimality called rotation coordinate descent (RCD). Unlike block coordinate descent (BCD) which solves SDP by updating the semidefinite matrix in a row-by-row fashion, RCD directly maintains and updates all valid rotations throughout the iterations. This obviates the need to store a large dense semidefinite matrix. We mathematically prove the convergence of our algorithm and empirically show its superior efficiency over state-of-the-art global methods on a variety of problem configurations. Maintaining valid rotations also facilitates incorporating local optimisation routines for further speed-ups. Moreover, our algorithm is simple to implement; see supplementary material for a demonstration program.

1. Introduction

Rotation averaging, a.k.a. multiple rotation averaging [16] or $SO(3)$ synchronisation [3], is the problem of estimating absolute rotations (orientations w.r.t. a common coordinate system) from a set of relative rotation measurements. In vision and robotics, rotation averaging plays a crucial role in SfM [19, 22, 21, 7, 6, 18, 37, 32] and visual SLAM [4, 26, 29, 23, 17], in particular for initialising bundle adjustment. Fig. 1 illustrates the result of rotation averaging. With the increase in the size of SfM problems and continued emphasis on real-time visual SLAM, developing efficient rotation averaging algorithms is an active research area. In particular, real-world applications often give rise to

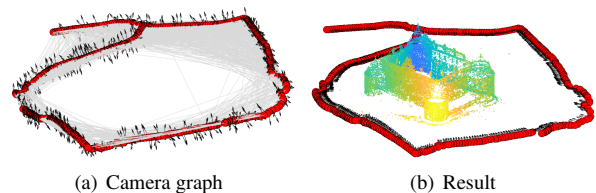


Figure 1. (a) Input camera graph from *Orebro Castle* [22] with $n = 761$ views and 116,589 connections (relative rotations: grey lines). The initial absolute rotations (represented as black arrows) were randomly chosen. For visualisation, the ground truth positions were used to locate the cameras (red points). (b) Globally optimal absolute rotations computed from our RCD algorithm in 1.96 s (Shonan averaging [8] required 54.62 s on the same input). Note the alignment of the arrows along the path of the camera (the reconstructed point cloud is also plotted for visualisation).

problem instances with thousands of cameras.

The input to rotation averaging is a set of noisy relative rotations $\{\tilde{R}_{ij}\}$, where each \tilde{R}_{ij} is a measurement of the orientation difference between cameras i and j which overlap in view. From the relative rotations, rotation averaging aims to recover the absolute rotations $\{R_i\}_{i=1}^n$ which represent the orientations of the cameras. In the ideal case where there is no noise in the relative rotations $\{R_{ij}\}$,

$$R_{ij} = R_j R_i^T. \quad (1)$$

The input relative rotations $\{\tilde{R}_{ij}\}$ define a *camera graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, n\}$ is the set of cameras, and $(i, j) \in \mathcal{E}$ is an edge in \mathcal{G} if the relative rotation \tilde{R}_{ij} between cameras i and j is measured. We assume a connected undirected graph \mathcal{G} , hence only \tilde{R}_{ij} with $i < j$ needs to be considered. See Fig. 1(a) for an example camera graph.

Rotation averaging is usually posed as a nonlinear optimisation problem with nonconvex domain

$$\min_{R_1, \dots, R_n \in SO(3)} \sum_{(i,j) \in \mathcal{E}} d(R_j R_i^T, \tilde{R}_{ij})^p, \quad (2)$$

where $d : SO(3) \times SO(3) \mapsto \mathbb{R}$ is a distance function that measures the deviation from the identity (1) based on measured and estimated quantities. For example,

$$d_{\text{chordal}}(R, S) = \|R - S\|_F, \quad (3)$$

which is known as the chordal distance, and

$$d_{\angle}(R, S) = \|\log(RS^T)\|_2 \quad (4)$$

which is called the angular distance ($\log : SO(3) \mapsto \mathbb{R}^3$ is the logarithmic map in $SO(3)$ [16]). Also, usually $p = 1, 2$.

The general form of (2) can be challenging to solve [16, 33]. Earlier efforts devised locally convergent methods [13, 20, 14, 15, 31, 5, 16], e.g., IRLS [5] and the Weiszfeld algorithm [15], though most are not able to guarantee local correctness [33]. Different from local methods, spectral decomposition methods [2] solve a relaxed problem optimally, though the deviation between the relaxed solution and the global solution is unknown. Tron *et al.* [32] surveyed and benchmarked approximate rotation averaging methods in the context of SfM. Recently, learning-based approaches [24] that can exploit the statistics of camera graphs from an environment have been developed.

1.1. Strong duality

Building upon empirical observations (e.g., [12]), Eriksson *et al.* [10] proved that the specific version

$$\min_{R_1, \dots, R_n \in SO(3)} \sum_{(i,j) \in \mathcal{E}} d_{\text{chordal}}(R_j R_i^T, \tilde{R}_{ij})^2, \quad (5)$$

which is a standard formulation in the literature [15, 16, 10, 5], satisfies *strong duality* [27] under mild conditions on the noise of the input relative rotations (see [10, Eq. (22)] or the supp. material for details). This means that the global solution to (5) can be obtained by solving its Lagrangian dual, which is a semidefinite program (SDP) (details in Sec. 2).

Our work focusses on solving the SDP relaxation of (5), especially for large-scale problems. Although SDPs are tractable, generic SDP solvers (e.g., conic optimisation [28]) can be slow on instances derived from rotation averaging. Thus, exploiting the problem structure to construct faster algorithms is an active research endeavour.

Eriksson *et al.* [10] presented a block coordinate descent (BCD) algorithm to solve the SDP relaxation, which consumed one order of magnitude less time than SeDuMi [28] on small to moderately sized instances ($n \leq 300$). The BCD algorithm maintains and iteratively improves a dense $3n \times 3n$ positive semidefinite (PSD) matrix by updating $3 \times 3n$ submatrices (called “block rows”) until convergence. At convergence, each block row contains rotation matrices (up to correcting for reflection) which are the solution to (5) (the solution of different block rows differ by a gauge freedom; see Sec. 2.3). However, recent results [30, 8] suggest that BCD is still not practical for large-scale problems encountered in SfM and SLAM, where $n \geq 1000$.

1.2. Riemannian staircase methods

The Riemannian staircase framework [1] has been applied successfully to pose graph optimisation (PGO) or $SE(3)$ synchronisation, which aim to recover absolute camera poses (6 DOF) from measurements of relative rigid motion. Under this framework, Rosen *et al.* [25] presented SE-Sync for PGO which guarantees global optimality for moderate noise levels. Tian *et al.* [30] builds upon SE-Sync to solve PGO in a distributed optimisation setting targeting collaborative SLAM for multi-robot missions.

Recently, Dellaert *et al.* [8] adapted SE-Sync for rotation averaging. Their algorithm, called Shonan rotation averaging (henceforth, “Shonan”) can globally solve the SDP relaxation of (5) through a chain of sub-problems on increasingly higher-dimensional domains $SO(d)$, with $d \geq 3$. A certification mechanism checks if the solution of each sub-problem has reached global optimality by computing the minimum eigenvalue of a large $3n \times 3n$ matrix. While optimality is ensured for $d \leq 3n + 1$, in practice the algorithm only needs to expand d once or twice to reach optimality. Results show that Shonan was an order of magnitude faster than BCD on moderate size instances ($n \leq 200$) and was able to solve large-scale instances ($n \geq 1000$) not achievable by BCD with impressive runtimes (instances with $n = 5750$ could be solved in 115 seconds).

1.3. Our contributions

We propose a novel algorithm called *rotation coordinate descent (RCD)* to solve rotation averaging (5) globally optimally. Unlike BCD, RCD neither maintains a $3n \times 3n$ dense PSD matrix nor updates the matrix block row-by-block row. Instead, the operation of RCD is equivalent to directly updating the n rotation matrices R_1, \dots, R_n , with provable convergence to global optimality. Moreover, since RCD maintains valid rotations at all times, local methods [5, 15] can be employed for further speed-ups.

See the supplementary material for an implementation of RCD and a demonstration program.

We will present results which show that RCD can be *up to two orders of magnitude* faster than Shonan, depending on the structure of the camera graph \mathcal{G} . More specifically, RCD is comparable to Shonan for sparse \mathcal{G} (e.g., SLAM camera graphs). However, RCD considerably outperforms Shonan on denser graphs (e.g., SfM camera graphs). This makes RCD a much more scalable algorithm.

On outliers An outlier in rotation averaging (2) is a measured relative rotation \tilde{R}_{ij} that significantly deviates from the true value. Note that formulation (5), i.e., least sum of squared chordal distances, is non-robust. Thus, if there are outliers in the input, BCD, Shonan and RCD will fail, in the sense that they do not return results that closely resemble

the “desired” solutions. In practice, such negative outcomes can be prevented by removing outliers with a preprocessing step [35, 9, 22]. We also emphasise that the theoretical validity of our work is not invalidated by the lack of robustness in the standard formulation (5) [15, 16, 5, 10].

Yang and Carlone [34] proposed a robust SDP relaxation for *single* rotation averaging, a special case where $n = 1$ (see [16]). The method has been demonstrated on relatively small scale problems (less than 100 measurements).

2. Preliminaries

2.1. Notation

We operate on block matrices composed of 3×3 blocks (submatrices in $\mathbb{R}^{3 \times 3}$). A block matrix is represented with a capital letter, e.g., $A \in \mathbb{R}^{3m \times 3n}$, and element (i, j) of a block matrix, denoted $A_{i,j}$, is the submatrix formed by rows $3(i-1)+1$ to $3(i-1)+3$ and columns $3(j-1)+1$ to $3(j-1)+3$ of A . Thus, $A_{i,i}$ are diagonal blocks.

We also define the k -th “row” of A as the submatrix

$$A_{k,:} = [A_{k,1} A_{k,2} \cdots A_{k,n}] \in \mathbb{R}^{3 \times 3n} \quad (6)$$

and similarly for the k -th “column” of A . If A has a single block column, we call it a “vector”. We use the notation

$$A_{(a:b);(c:d)} = \begin{bmatrix} A_{a,c} & \cdots & A_{a,d} \\ \vdots & & \vdots \\ A_{b,c} & \cdots & A_{b,d} \end{bmatrix} \in \mathbb{R}^{3m \times 3n} \quad (7)$$

for the submatrix of A from rows a to b and columns c to d . If A is a vector we use the notation $A_k = A_{k,1}$ and $A_{a:b} = A_{(a:1);(b:1)}$.

We denote the 3×3 identity and zero matrices as I_3 and O_3 , and the trace and Moore–Penrose pseudoinverse of a matrix M as $\text{tr}(M)$ and M^\dagger , respectively.

2.2. SDP relaxation

We first present the SDP relaxation of (5) following [10]. By rewriting the chordal distance using trace, (5) becomes

$$\min_{R_1, \dots, R_n \in SO(3)} - \sum_{(i,j) \in \mathcal{E}} \text{tr}(R_i^T \tilde{R}_{ij} R_j). \quad (8)$$

This can be further written more compactly as

$$\min_{R \in SO(3)^n} - \text{tr}(R^T \tilde{R} R) \quad (\text{P})$$

using matrix notations, where

$$R = [R_1^T R_2^T \cdots R_n^T]^T \in SO(3)^n \quad (9)$$

contains the target variables, and \tilde{R} is the $3n \times 3n$ block symmetric matrix with upper-triangle elements (i, j) equal

Algorithm 1 Block coordinate descent (BCD) for (DD).

Require: \tilde{R} and $Y^{(0)} \succeq 0$.

- 1: $t \leftarrow 0$.
 - 2: **repeat**
 - 3: Select an integer k in the interval $[1, n]$.
 - 4: $W \leftarrow$ the k -th column of \tilde{R} .
 - 5: $Z \leftarrow Y^{(t)} W$.
 - 6: $S \leftarrow Z \left[(W^T Z)^{\frac{1}{2}} \right]^\dagger$.
 - 7: $Y^{(t+1)} \leftarrow \begin{bmatrix} Y_{(1:k-1);(1:k-1)}^{(t)} & S_{1:(k-1)} & Y_{(1:k-1);(k+1:n)}^{(t)} \\ S_{1:(k-1)}^T & I_3 & S_{(k+1):n}^T \\ Y_{(k+1:n);(1:k-1)}^{(t)} & S_{(k+1):n} & Y_{(k+1:n);(k+1:n)}^{(t)} \end{bmatrix}$
 - 8: $t \leftarrow t + 1$.
 - 9: **until** convergence
 - 10: **return** $Y^* = Y^{(t)}$.
-

to \tilde{R}_{ij} if $(i, j) \in \mathcal{E}$ and O_3 otherwise (diagonal elements are O_3 's). Problem (P) is called the primal problem.

As derived in Eriksson *et al.* [10], the dual of the Lagrangian dual of (P) is the SDP relaxation

$$\min_{Y \in \mathbb{R}^{3n \times 3n}} - \text{tr}(\tilde{R} Y) \quad (\text{DD})$$

$$\text{s.t.} \quad Y_{i,i} = I_3, \quad i = 1, \dots, n. \quad (10a)$$

$$Y \succeq 0, \quad (10b)$$

where Y is a $3n \times 3n$ PSD matrix, and $Y_{i,i}$ is the i -th diagonal block of Y . The interested reader is referred to Eriksson *et al.* for the detailed derivations. It is proven that, under mild conditions (see supp. material), that

$$- \text{tr}(\tilde{R} Y^*) = - \text{tr}(R^{*T} \tilde{R} R^*), \quad (11)$$

where R^* and Y^* are respectively the optimisers of (P) and (DD), i.e., zero duality gap between (P) and (DD).

Output rotations Note that constraint (10a) in (DD) merely enforces orthogonality in each diagonal block. Hence, in general a feasible Y for (DD) is *not* factorisable as the product of two rotation matrices RR^T . It can be shown, however, that the optimiser Y^* of (DD) is rank-3 [10], which admits the factorisation

$$Y^* = Q^* Q^{*T}, \quad (12)$$

where $Q^* \in O(3)^n$ contains n 3×3 orthogonal matrices. To obtain R^* , first Q^* is obtained via SVD on Y^* , then for each Q_i^* whose determinant is negative, the sign of the Q_i^* is flipped to positive to yield a valid rotation.

2.3. Block coordinate descent

Algorithm 1 presents BCD [10] for (DD) using our notation, which also includes a minor improvement to the original. Specifically, instead of working on an auxiliary square

matrix obtained by removing the k -th row and column from $Y^{(t)}$ (see [10, Step 3 of Algorithm 1]), we directly operate over $Y^{(t)}$ and create a temporary block vector Z (Line 5). Since Z is smaller than the auxiliary square matrix, the efficiency of Line 6 which requires operating over Z twice is marginally improved. We emphasise that Algorithm 1 is intrinsically the same as the original (see supp. material for details and validity of the improvement).

The PSD matrix Y can be initialised as an arbitrary PSD matrix. A simple choice is setting $R_i = I_3$ for all i in R and initialising $Y^{(0)} = RR^T$. However, we remind again that the subsequent $Y^{(t)}$ are not factorisable as the product of rotations $R^{(t)}R^{(t)T}$ in general; see Sec 2.2.

Gauge freedom Note that the factorisation (12) is up to an arbitrary orthogonal transformation $G \in O(3)$, i.e.,

$$Y^* = Q^*Q^{*T} = (Q^*G)(Q^*G)^T. \quad (13)$$

We say that G represents a ‘‘gauge freedom’’ in the solution. This leads to another approach to retrieve R^* from Y^* , which recognises that the columns (and rows) of Y^* are related by orthogonal transformations as Y^* is rank-3 with diagonal elements equal to I_3 . Therefore, for any two columns k and k' in Y^* , there exists an orthogonal transformation $G_{k,k'} \in O(3)$ such that

$$Y_{:,k'}^* = Y_{:,k}^* G_{k,k'}. \quad (14)$$

Hence, $G_{k,k'}$ must transform the k' -th element of $Y_{:,k}^*$ to I_3 (i.e., $Y_{k',k}^* G_{k,k'} = I_3$). Therefore

$$G_{k,k'} = (Y_{k',k}^*)^T = Y_{k,k'}^* \quad (15)$$

as columns in Y^* are orthogonal and Y^* is symmetric.

The set of transformations relating columns (15)

$$\mathcal{G} = \{G_{k,k'}, \text{ for all } k, k' = 1, \dots, n\} \subset O(3) \quad (16)$$

corresponds to an special case of gauge freedom. Since all columns in Y^* are up to some transformation in \mathcal{G} to another column, we can take any as R^* ; the choice will depend on selecting one of the cameras as the reference frame, i.e., which camera takes $R_i^* = I_3$.

3. Rotation coordinate descent

In this section, we will describe our novel method called *rotation coordinate descent (RCD)*, summarised in Algorithm 2. While seemingly a minor modification to BCD, RCD is based on nontrivial insights (Sec. 3.1). More importantly, a major contribution is to mathematically prove the global convergence of RCD (Sec. 3.2). Another fundamental advantage is that since RCD maintains valid rotations throughout the iterations (in contrast to BCD; see Sec. 2.3),

Algorithm 2 Rotation coordinate descent (RCD) for (DD).

Require: \tilde{R} and $R^{(0)}$.

- 1: $t \leftarrow 0$.
- 2: **repeat**
- 3: Select an integer k in the interval $[1, n]$.
- 4: $W \leftarrow$ the k -th column of \tilde{R} .
- 5: $Z \leftarrow R^{(t)}(R^{(t)T}W)$.
- 6: $S \leftarrow Z \left[(W^T Z)^{\frac{1}{2}} \right]^\dagger$.
- 7: $Q^{(t+1)} \leftarrow [(S_{1:(k-1)})^T \ I_3 \ (S_{(k+1):n})^T]^T$.
- 8: $R^{(t+1)} \leftarrow$ Flip determinants over $Q^{(t+1)}$ (if needed) to ensure rotations.
- 9: $t \leftarrow t + 1$.
- 10: **until** convergence
- 11: **return** $Y^* = R^{(t)}R^{(t)T}$.

it can exploit local optimisation routines for (P) to speed-up convergence (Sec. 4). As the results will show (Sec. 5), our approach can be up to two orders of magnitude faster than Shonan [8], which is the state of the art for (DD).

3.1. Main ideas

As summarised in Algorithm 1, BCD requires to maintain and operate on a large dense PSD matrix $Y \in \mathbb{R}^{3n \times 3n}$. While the values of each update can be computed in constant time (specifically, SVD of a 3×3 matrix; Line 6), manipulating Y is unwieldy. Specifically, Line 5 performs

$$Z = Y^{(t)}W \quad (17)$$

to obtain temporary vector $Z \in \mathbb{R}^{3n \times 3}$ from a subset of the measurements $W \in \mathbb{R}^{3n \times 3}$, which costs

$$27n^2 \text{ multiplications} \equiv \mathcal{O}(n^2). \quad (18)$$

This quadratic dependence on n makes BCD slow on large-scale SfM or SLAM problems [30], e.g., where $n \geq 1000$, as we will also demonstrate in Sec. 5.

Although the PSD matrix Y of (DD) has size $3n \times 3n$, the ‘‘effective’’ variables are only $3n$ given that Y^* is rank-3. Our key insight comes from the gauge freedom of Y^* (Sec. 2.3) implying that any row of Y^* provides a valid solution for R^* . Choosing the k -th row implies choosing the k -th camera as the reference frame, i.e., $R_k = I_3$. Based on this insight, we devised RCD to maintain only the effective variables $R^{(t)}$. Each iteration executes what amounts to updating a single column of Y ; specifically, in Line 3, a camera k is picked to be the reference frame then set the k -th element of $Q^{(t+1)}$ as I_3 in Line 7 ($Q^{(t)}$ contains orthogonal matrices). Then, in Line 6 the other elements of $Q^{(t+1)}$ are updated via the same explicit form of BCD. To ensure keeping rotations elements during iterations, the sign

of the orthogonal elements in $Q^{(t+1)}$ is flipped if negative in Line 9 to produce $R^{(t+1)}$.

Maintaining and updating only $R^{(t)}$ provides immediate computational savings; in Line (5) obtaining the intermediate vector Z is now accomplished as

$$Z = R^{(t)} \underbrace{(R^{(t)T} W)}_{\text{Compute this first}}, \quad (19)$$

which costs

$$27n + 27n \text{ multiplications} \equiv \mathcal{O}(n) \quad (20)$$

and has only linear dependence on n . The next section proves the important result that this computational savings does not come at the expense of global optimality.

3.2. Global convergence of RCD

As proven in [10, 11], Algorithm 1 monotonically decreases the objective $-\text{tr}(\tilde{R}Y)$ at each iteration from any feasible initialisation. Our strategy for proving the global convergence of RCD is to show that updating the variables at each iteration t of Algorithm 2, i.e.,

$$R^{(t)} \rightarrow R^{(t+1)}, \quad (21)$$

has an effect on $-\text{tr}(\tilde{R}Y)$ that is equivalent to one iteration of Algorithm 1 initialised with

$$Y^{(0)} = R^{(t)} R^{(t)T}. \quad (22)$$

If this equivalence can be established, Algorithm 2 also provably monotonically decreases $-\text{tr}(\tilde{R}Y)$ and will converge to the optimiser Y^* of (DD).

To this end, we will first show that one iteration of Algorithm 1 initialised with (22) produces a PSD matrix $Y^{(1)}$ that is factorisable as

$$Y^{(1)} = R^{(1)} R^{(1)T}. \quad (23)$$

Note that in general this factorisation does not hold for $Y^{(t)}$ for $t > 1$ in Algorithm 1. Without loss of generality, we take $k = 1$ (the updated row and column in BCD during the iteration) and define $R_{\text{BCD}}^{(1)}$ as the first column of $Y^{(1)}$, i.e.,

$$R_{\text{BCD}}^{(1)} = Y_{:,1}^{(1)}. \quad (24)$$

Then, we will prove that $R^{(t+1)} = R_{\text{BCD}}^{(1)}$. From Line 7 in Algorithm 1, $Y^{(1)}$ can be written (for $k = 1$) as

$$Y^{(1)} = \begin{bmatrix} I_3 & X^{*T} \\ X^* & B \end{bmatrix}, \quad (25)$$

where $B = Y_{(2:n);(2:n)}^{(0)}$ is the unchanged sub-matrix during the iteration $Y^{(0)} \rightarrow Y^{(1)}$, and $X^* \in \mathbb{R}^{3(n-1) \times 3}$ contains

the updated values. From [10, 11], X^* is the optimiser of the following SDP problem:

$$\min_{X \in \mathbb{R}^{3(n-1) \times 3}} -\text{tr}(C^T X) \quad (26a)$$

$$\text{s.t.} \quad \begin{bmatrix} I_3 & X^T \\ X & B \end{bmatrix} \succeq 0, \quad (26b)$$

where $C \in \mathbb{R}^{3(n-1) \times 3}$ is equal to W as in Line 4 in Algorithm 1 but without the k -th element (which is zero).

Note that the optimal PSD matrix in Problem (26) is $Y^{(1)}$ (25). The goal of Problem (26) is to find the optimal update X^* to produce $Y^{(1)}$ that keeps feasibility (constraint (26b)).

We show in the next result that problem (26) is a special case of (DD); hence, the optimal PSD matrix of Problem (26) admits the factorisation

$$Y^{*(1)} = R^{*(1)} R^{*(1)T}, \quad (27)$$

which proves (23) as BCD optimally solves Problem (26) during updates (Lines 5–7 in Algorithm 1) [10, 11].

Result 1 (Problem (26) is a special case of (DD))

Consider the instance of Problem (DD) with

$$\tilde{R} = \begin{bmatrix} 0_3 & C^T \\ C & 0 \end{bmatrix}. \quad (28)$$

We first show that a feasible PSD matrix in Problem (26)

$$Y = \begin{bmatrix} I_3 & X^T \\ X & B \end{bmatrix} \quad (29)$$

is a feasible solution in (DD).

From the initialisation of $Y^{(0)}$ in (23),

$$B = R_{2:n}^{(t)} R_{2:n}^{(t)T}; \quad (30)$$

hence, all diagonal elements in Y (29) are identities which fulfills the first constraint (10a) in (DD). From (26b), $Y \succeq 0$, which is the second constraint (10b) in (DD).

We now show the objective of (DD) with \tilde{R} from (28) is equivalent to the objective in Problem (26). The objective of (DD) becomes

$$-\text{tr}(\tilde{R}Y) = -\text{tr} \left(\begin{bmatrix} C^T X & C^T B \\ C & C X^T \end{bmatrix} \right) \quad (31a)$$

$$= -\text{tr}(C^T X) - \text{tr}(C X^T) \quad (31b)$$

$$= -2 \text{tr}(C^T X) \quad (31c)$$

which is twice to the objective of (26). Therefore, Problem (26) is a special case of (DD) since any feasible solution of (26) is also feasible in (DD), and both objectives are equivalent.

Finally, we establish that $R^{(t+1)} = R_{\text{BCD}}^{(1)}$.

Result 2 ($R^{(t+1)} = R_{\text{BCD}}^{(1)}$) The equality is by construction of Algorithm 2. From the definition of $R_{\text{BCD}}^{(1)}$ in (22), $R_{\text{BCD}}^{(1)}$ is the first column in $Y^{(1)}$ (25), i.e.,

$$R_{\text{BCD}}^{(1)} = \begin{bmatrix} I_3 & X^{*T} \end{bmatrix}^T, \quad (32)$$

where X^* is S (in Line 6, Algorithm 1) without the k -th element (see sup. material for details). Lines 3–6 in Algorithm 1 are the same to Lines 3–6 in Algorithm 2 except on obtaining Z , which takes the same value since from the initialisation (22) of $Y^{(0)}$ in Algorithm 1,

$$Z = Y^{(0)}W = R^{(t)}(R^{(t)T}W) \quad (33)$$

is equal to Z as obtained in Algorithm 2. Thus also

$$R^{(t+1)} = \begin{bmatrix} I_3 & X^{*T} \end{bmatrix}^T. \quad (34)$$

4. Speeding up RCD with local optimisation

Since Algorithm 2 iterates over $SO(3)^n$, local methods for (P) can be directly used to speedup convergence of Algorithm 2. Contrast this to BCD that updates a PSD matrix from where, in general, valid rotations can be retrieved only at convergence. Algorithm 3 integrates a local method (Line 13) that we design from experimental observations:

1. Substantial reductions in the objective often occur after n iterations. We call it an *epoch* and we ensure we sample all k 's during each epoch (Line 4).
2. In practice, one iteration of Algorithm 2 takes $\approx 0.02\%$ of the runtime of solving a local optimisation instance. Thus, Algorithm 3 invokes local optimisation and check for convergence only after completing epochs.
3. Local optimisation produces more drastic “jumps” in the objective at earlier iterations. Thus, Algorithm 3 delays local optimisation when the local method fails on reducing the objective (Line 17).

To demonstrate the effect of local optimisation on the convergence of RCD, Fig. 2 plots the objective value for RCD and RCDL at increasing epochs on the input graph *torus* [4] with $n = 5000$ cameras (see Table 1 in Sec. 5 for more details). During the 1st epoch, the local algorithm drastically reduced the objective (from stage in *green* to stage in *magenta*). This “jump” of the objective value reveals the collaborative strength of global and local methods, which enabled RCDL to converge in much fewer epochs (*red* stage) compared to RCD (*blue* stage).

5. Experiments

We benchmarked the following algorithms over a variety of synthetic and real world camera graph inputs: Algorithm 1 (BCD), Algorithm 2 (RCD), Algorithm 3 (RCDL)

Algorithm 3 RCD with local optimisation (RCDL).

Require: \tilde{R} and $R^{(0)}$.

```

1:  $t \leftarrow 0, e \leftarrow 0, s \leftarrow 0$ 
2: repeat
3:   for  $i = 1, \dots, n$  do
4:     Select an integer  $k$  in the interval  $[1, n]$  w/o rep.
5:      $W \leftarrow$  the  $k$ -th column of  $\tilde{R}$ .
6:      $Z \leftarrow R^{(t)}(R^{(t)T}W)$ .
7:      $S \leftarrow Z \left[ (W^T Z)^{\frac{1}{2}} \right]^\dagger$ .
8:      $R^{(t+1)} \leftarrow [(S_{1:(k-1)})^T \ I_3 \ (S_{(k+1):n})^T]^T$ .
9:      $t \leftarrow t + 1$ .
10:  end for
11:  if ( $s = 0$  or  $\text{MOD}(e, s) = 0$ ) then
12:     $R^{(t)} \leftarrow$  Flip determinants over  $R^{(t)}$  (if needed) to
    ensure rotations.
13:     $\hat{R} \leftarrow$  local method with initial estimate  $R^{(t)}$ .
14:    if  $-\text{tr}(\hat{R}^T \tilde{R} \hat{R}) < -\text{tr}(R^{(t)T} \tilde{R} R^{(t)})$  then
15:       $R^{(t)} \leftarrow \hat{R}$ .
16:    else
17:       $s \leftarrow s + 2$ 
18:    end if
19:  end if
20:   $e \leftarrow e + 1$ .
21: until convergence
    
```

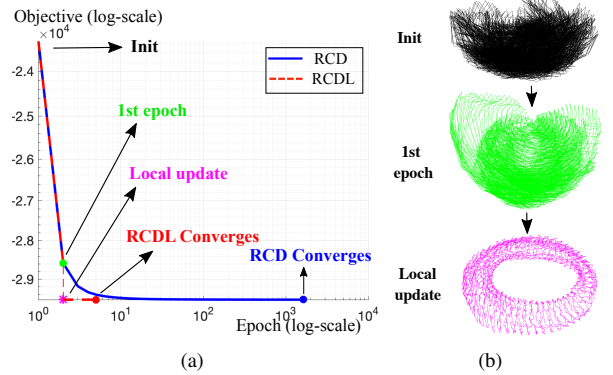


Figure 2. Evolution of RCD and RCDL on the large-scale SLAM instance *torus* [4] with $n = 5000$ cameras. (a) Evolution of the objectives. (b) Camera poses from RCDL. A single local update was able to produce a visually correct solution.

with local optimisation routine adapted from [23], and Shonan [8] (SA). We implemented all the optimisation routines in C++ except for SA for which we used the author’s implementation (which also has optimisation routines in C++¹). We ran our experiments on a standard machine with an Intel Core i5 2.3 GHz CPU and 8 GB RAM.

¹<https://github.com/dellaert/ShonanAveraging>

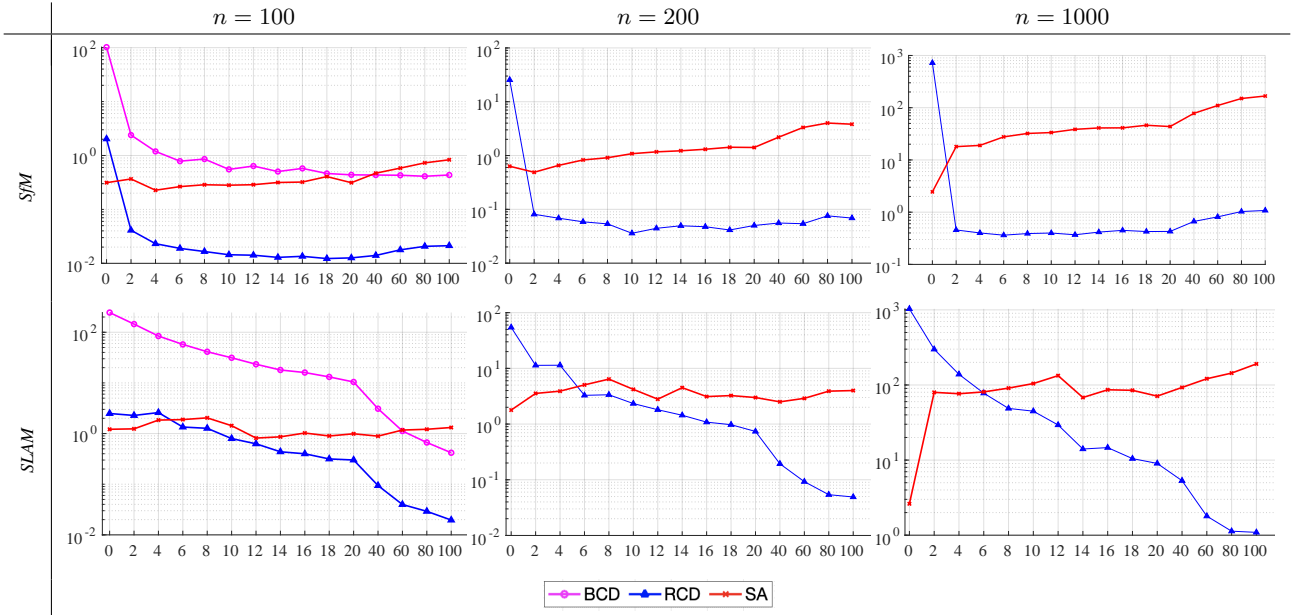


Figure 3. Runtime [s] (y -axis in log-scale) at varying graph densities d_G (x -axis in $\times 10^{-2}$) for SfM and SLAM graphs with $n = 100, 200, 1000$ cameras. We denser sampled the interval $[0, 0.2]$.

Graph density Consider a connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $n = |\mathcal{V}|$ vertices and $m = |\mathcal{E}|$ edges. Define

$$d_G := \frac{|\mathcal{E}| - |\mathcal{E}_{\min}|}{|\mathcal{E}_{\max}| - |\mathcal{E}_{\min}|}, \quad (35)$$

as the density of graph \mathcal{G} , where \mathcal{E}_{\max} and \mathcal{E}_{\min} denote the set of edges of the complete $(\mathcal{V}, \mathcal{E}_{\max})$ and the cycle graph $(\mathcal{V}, \mathcal{E}_{\min})$. Excluding graphs with $n - 1$ edges², d_G takes values in $[0, 1]$. Thus, by definition (35), $d_G = 0$ for a cycle graph and $d_G = 1$ for a complete graph.

5.1. Synthetic Data

To test RCD over a variety of graph configurations, we synthesised graphs with varying densities to simulate SfM and SLAM problems. As SfM often solves reconstruction from views with large baselines, we generated random camera positions and random connections in the SfM setting. In contrast, for the SLAM setting, we simulated views with a smooth trajectory and connect only nearby views. We created measurements of relative rotations (1) by multiplying the ground truth relative rotations with rotations with random axes and angles normally distributed with $\sigma = 0.1$ rad. For a fair comparison, we initialised all methods with the same initial random absolute rotations.

Varying graph densities Fig. 3 shows the runtimes averaged over 10 runs for all methods. RCD significantly outperformed SA for $d_G > 0.1$. In general, camera graphs

²Rotation averaging instances are typically overdetermined, i.e., problems with $|\mathcal{E}| > n - 1$ edges.

from real world SfM datasets are often dense. See for example d_G values for the real world instances in Table 2 with average $d_G \approx 0.53$. For larger problems, BCD was not able to terminate within reasonable time (≤ 1000 s); we did not report results for BCD for $n > 100$. Although RCD was not considerably faster than SA when $d_G < 0.04$, the convergence rate can be accelerated by using a local optimisation routine as we show in Sec. 5.2.

Varying noise levels and number of cameras In Fig. 4, we plotted the runtimes of RCD and SA on SfM camera graphs with varying noise levels σ , number of cameras n , but with fixed $d_G = 0.4$. We omitted the comparison against BCD as it did not converge within a sensible time for large problems, as demonstrated in Fig. 3. Fig. 4(a) shows that runtimes for RCD and SA were marginally affected by noise. Fig. 4(b) shows that RCD outperformed SA by two orders of magnitude ($1.5s$ vs $312.9s$ at $n = 1,800$)—this further demonstrates the superior scalability of RCD.

We repeated the above experiment with $d_G = 0.2$ which simulates SLAM graphs. See supp. material for the results.

5.2. SLAM benchmark dataset

We compared runtimes on large-scale problems from the SLAM dataset in [4]. Table 1 reports the input characteristics of each benchmarking instance and the results for all methods. Here, we initialised rotations from a random spanning tree. Note that initialisation does not affect the global optimality of tested algorithms. The spanning tree initialisation is fast and practical. We remark that in real-world

Dataset characteristics				Error [%]				Efficiency					
Name	$ \mathcal{V} $ n	$ \mathcal{E} $ m	d_G	Init.	RCD	RCDL	SA	# Epoch		Time [s]			Speedup
								RCD	RCDL	RCD	RCDL	SA	
smallgrid	125	297	0.02200	-16.13	0	-4.77E-09	-8.38E-05	46	10	0.07	0.02	0.06	2.7
garage	1661	6275	0.00340	-7.29E-05	-3.63E-06	0	-1.40E-07	29	2	3.74	0.28	4.76	17.1
sphere	2500	4949	0.00078	-1.70	-6.24E-06	0	-7.84E-07	352	2	105.70	0.66	17.07	25.7
torus	5000	9898	0.00039	-20.95	-2.55E-05	0	-1.73E-06	1620	4	1808.86	4.86	15.76	3.2
grid3D	8000	22819	0.00046	-15.41	-4.54E-06	0	-2.14E-06	409	4	1199.30	14.78	23.93	1.6

Table 1. Quantitative results for the SLAM Benchmark dataset [4]. Error of the initial solution (Init.) and each method is the % of its objective w.r.t. the lowest obtained objective among all methods. One epoch is equivalent to n iterations as described in Sec. 4. Speedup is presented for the best result of RCD and RCDL against SA.

Dataset characteristics				Error [%]				Efficiency					
Name	$ \mathcal{V} $ n	$ \mathcal{E} $ m	d_G	Init.	RCD	RCDL	SA	# Epoch		Time [s]			Speedup
								RCD	RCDL	RCD	RCDL	SA	
Alcatraz Tower	172	14706	1.00	-0.66	-8.66E-10	0	-4.64E-08	4	2	0.03	0.12	0.63	25.3
Doge Palace	241	19753	0.68	-0.89	-1.16E-08	0	-1.04E-07	8	2	0.09	0.21	1.00	11.1
King’s College	328	41995	0.78	-1.57	-5.01E-10	0	-3.81E-08	7	2	0.11	0.60	2.37	21.5
Alcatraz Garden	419	51635	0.59	-1.29	-7.70E-09	0	-5.57E-08	11	2	0.24	0.89	3.24	13.5
Linkoping	538	34462	0.24	-1.22	-3.62E-07	0	-4.03E-06	37	2	0.90	0.43	6.44	15.0
UWO	692	80301	0.33	-1.26	-1.38E-07	0	-7.88E-07	20	2	0.85	1.65	11.12	13.1
Orebro Castle	761	116589	0.40	-1.19	-9.40E-08	0	-1.04E-06	20	2	1.10	4.01	26.17	23.8
Spilled Blood	781	117814	0.39	-2.81	-3.20E-08	0	-6.61E-07	14	2	0.79	4.32	37.64	47.6
Lund Cathedral	1207	177289	0.24	-1.16	-9.62E-07	0	-1.91E-06	78	2	8.45	7.03	41.08	5.8
San Marco	1498	757037	0.67	-0.74	-6.60E-09	0	-8.97E-09	6	2	1.61	145.46	110.07	68.4

Table 2. Quantitative results for the SfM large scale real-world dataset [22]. See Table 1 for the description of each column.

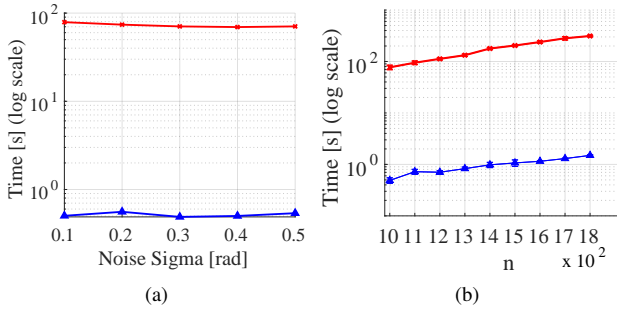


Figure 4. Runtime [s] (in log-scale) for SfM camera graphs with $d_G = 0.4$. (a) Varying σ in $[0.1, 0.5]$ rad. and $n = 1000$. (b) Varying n in $[1000, 1800]$ with $\sigma = 0.1$ rad. See Fig. 3 for the description of the legend.

applications it is unnecessary to solve camera orientations from random rotations. Again, we initialised all algorithms with the same initial estimates for fair comparison. We provided the errors (in %) of resulting objective value (including the initialisation) relative to the lowest objective value reported among all methods.

Camera graphs are very sparse for the SLAM Benchmark in Table 1 ($d_G \leq 0.022$). Although RCD was not as fast as SA on sphere, torus and grid3D (note that $d_G \leq$

0.0007 in those instances), the use of a local optimisation in RCDL permitted to outperform SA; see also Fig. 2.

5.3. Real world SfM dataset

Table 2 presents runtimes over real-world SfM datasets [22]³ where RCD outperformed SA. We remark that RCDL took substantially fewer epochs compared to RCD to converge. However, RCDL did not achieve a better runtime as local optimisation consumed on average $\approx 90\%$ of the total runtime, especially for large graph densities. Fig. 5 shows the reconstructed *Spilled Blood Cathedral* using the estimated camera orientations of RCDL after running for 1 epoch in 4.2s.

6. Conclusions

We present RCD, a fast rotation averaging algorithm that finds the globally optimal rotations under mild conditions on the noise level of the measurements. Our insights on gauge freedom has circumvented the quadratic computational burden of BCD, which is an established method for global rotation averaging. Also, since RCD maintains valid rotations instead of a dense PSD matrix, local optimisation routines can be beneficially integrated. Experimental results

³<http://www.maths.lth.se/matematiklth/personal/calle/dataset/dataset.html>

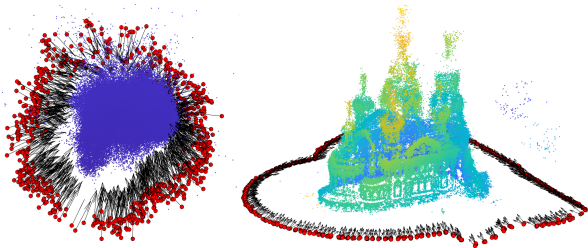


Figure 5. Reconstruction of the *Spilled Blood Cathedral* by solving the known rotation problem (KROT) [36]. *Left*: Initial camera orientations. *Right*: Result from RCDL after 1 epoch.

demonstrated the superior efficiency of RCD, which significantly outperformed state-of-the-art algorithms on a variety of problem configurations.

References

- [1] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009. 2
- [2] Federica Arrigoni, Beatrice Rossi, and Andrea Fusiello. Spectral synchronization of multiple views in $SE(3)$. *SIAM SIMAX*, 9(4):1963–1990, 2016. 2
- [3] Nicolas Boumal, Amit Singer, P-A Absil, and Vincent D Blondel. Cramér-Rao bounds for synchronization of rotations. *Information and Inference: A Journal of the IMA*, 3(1):1–39, 2014. 1
- [4] Luca Carlone, Roberto Tron, Kostas Daniilidis, and Frank Dellaert. Initialization techniques for 3D SLAM: a survey on rotation estimation and its use in pose graph optimization. In *IEEE ICRA*, 2015. 1, 6, 7, 8
- [5] Avishek Chatterjee and Venu Madhav Govindu. Efficient and robust large-scale rotation averaging. In *ICCV*, 2013. 2, 3
- [6] Hainan Cui, Xiang Gao, Shuhan Shen, and Zhanyi Hu. HSfM: hybrid structure-from-motion. In *CVPR*, 2017. 1
- [7] Zhaopeng Cui and Ping Tan. Global structure-from-motion by similarity averaging. In *ICCV*, 2015. 1
- [8] Frank Dellaert, David M Rosen, Jing Wu, Robert Mahony, and Luca Carlone. Shonan rotation averaging: Global optimality by surfing $SO(p)^n$. In *ECCV*, 2020. 1, 2, 4, 6
- [9] Olof Enqvist, Fredrik Kahl, and Carl Olsson. Non-sequential structure from motion. In *ICCVW*, 2011. 3
- [10] Anders Eriksson, Carl Olsson, Fredrik Kahl, and Tat-Jun Chin. Rotation averaging and strong duality. In *CVPR*, 2018. 2, 3, 4, 5
- [11] Anders Eriksson, Carl Olsson, Fredrik Kahl, and Tat-Jun Chin. Rotation averaging with the chordal distance: Global minimizers and strong duality. *IEEE TPAMI*, 2019. 5
- [12] Johan Fredriksson and Carl Olsson. Simultaneous multiple rotation averaging using Lagrangian duality. In *ACCV*, 2012. 2
- [13] Venu Madhav Govindu. Combining two-view constraints for motion estimation. In *CVPR*, 2001. 2
- [14] Venu Madhav Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *CVPR*, 2004. 2
- [15] Richard Hartley, Khurram Aftab, and Jochen Trumpf. L1 rotation averaging using the Weiszfeld algorithm. In *CVPR*, 2011. 2, 3
- [16] Richard Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *IJCV*, 103(3):267–305, 2013. 1, 2, 3
- [17] Xinyi Li and Haibin Ling. Hybrid camera pose estimation with online partitioning for SLAM. *IEEE RAL*, 5(2):1453–1460, 2020. 1
- [18] Alex Locher, Michal Havlena, and Luc Van Gool. Progressive structure from motion. In *ECCV*, 2018. 1
- [19] Daniel Martinec and Tomas Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *CVPR*, 2007. 1
- [20] Maher Moakher. Means and averaging in the group of rotations. *SIAM SIMAX*, 24(1):1–16, 2002. 2
- [21] Pierre Moulon, Pascal Monasse, and Renaud Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *ICCV*, 2013. 1
- [22] Carl Olsson and Olof Enqvist. Stable structure from motion for unordered image collections. In *Springer SCIA*, 2011. 1, 3, 8
- [23] Álvaro Parra, Tat-Jun Chin, Anders Eriksson, and Ian Reid. Visual SLAM: Why bundle adjust? In *IEEE ICRA*, 2019. 1, 6
- [24] Pulak Purkait, Tat-Jun Chin, and Ian Reid. NeuRoRA: Neural robust rotation averaging. In *ECCV*, 2020. 2
- [25] David M Rosen, Luca Carlone, Afonso S Bandeira, and John J Leonard. SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group. *IJRR*, 38(2-3):95–125, 2019. 2
- [26] David M Rosen, Charles DuHadway, and John J Leonard. A convex relaxation for approximate global optimization in simultaneous localization and mapping. In *IEEE ICRA*, 2015. 1
- [27] Andrzej Ruszczyński. *Nonlinear optimization*. Princeton university press, 2011. 2
- [28] Jos F Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1-4):625–653, 1999. 2
- [29] Chengzhou Tang, Oliver Wang, and Ping Tan. GSLAM: Initialization-robust monocular visual SLAM via global structure-from-motion. In *IEEE 3DV*, 2017. 1
- [30] Yulun Tian, Kasra Khosoussi, David M Rosen, and Jonathan P How. Distributed certifiably correct pose-graph optimization. *arXiv preprint arXiv:1911.03721*, 2019. 2, 4
- [31] Roberto Tron, Bijan Afsari, and René Vidal. Intrinsic consensus on $SO(3)$ with almost-global convergence. In *IEEE CDC*, 2012. 2
- [32] Roberto Tron, Xiaowei Zhou, and Kostas Daniilidis. A survey on rotation optimization in structure from motion. In *CVPRW*, 2016. 1, 2
- [33] Kyle Wilson, David Bindel, and Noah Snavely. When is rotations averaging hard? In *ECCV*, 2016. 2
- [34] Heng Yang and Luca Carlone. One ring to rule them all: Certifiably robust geometric perception with outliers. *Advances in Neural Information Processing Systems*, 33, 2020. 3

- [35] Christopher Zach, Manfred Klopschitz, and Marc Pollefeys. Disambiguating visual relations using loop constraints. In *CVPR*, 2010. 3
- [36] Qiangong Zhang, Tat-Jun Chin, and Huu Minh Le. A fast resection-intersection method for the known rotation problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3012–3021, 2018. 9
- [37] Siyu Zhu, Runze Zhang, Lei Zhou, Tianwei Shen, Tian Fang, Ping Tan, and Long Quan. Very large-scale global SfM by distributed motion averaging. In *CVPR*, 2018. 1

Supplementary Material for: Rotation Coordinate Descent for Fast Globally Optimal Rotation Averaging

Alvaro Parra Bustos*
University of Adelaide

alvaro.parrabustos@adelaide.edu.au

Shin-Fang Chng*
University of Adelaide

shinfang.chng@adelaide.edu.au

Tat-Jun Chin
University of Adelaide

tat-jun.chin@adelaide.edu.au

Anders Eriksson
University of Queensland

a.eriksson@uq.edu.au

Ian Reid
University of Adelaide

ian.reid@adelaide.edu.au

A. Demonstration program

To run the demonstration program, please follow the instructions in the README.md file in folder demo_RCD.

B. Further details

B.1. Conditions on the noise level for the strong duality of Eq. (5)

For the following rotation averaging problem (Eq. (5) in the main text)

$$\min_{R_1, \dots, R_n \in SO(3)} \sum_{(i,j) \in \mathcal{E}} d_{\text{chordal}}(R_j R_i^T, \tilde{R}_{ij})^2, \quad (1)$$

we present a bound on the *angular* residual errors

$$\alpha_{ij} = d_{\angle}(R_j^* R_i^{*T}, \tilde{R}_{ij}) \quad (2)$$

such that its *strong duality* holds.

The main result of [Theorem 4.1, 10] is the proof of the strong duality of Problem (1) if

$$|\alpha_{ij}| \leq \alpha_{\max} \quad \forall (i, j) \in \mathcal{E}, \quad (3)$$

where

$$\alpha_{\max} = 2 \arcsin \left(\sqrt{\frac{1}{4} + \frac{\lambda_2(L_G)}{2d_{\max}}} - \frac{1}{2} \right). \quad (4)$$

$\lambda_2(L_G)$ and d_{\max} in (4) are related to the structure of the camera graph. More precisely, α_{\max} depends on the connectivity of the camera graph represented by its Fiedler value $\lambda_2(L_G)$ (the second smallest eigenvalue of its Laplacian L_G), and its maximal vertex degree d_{\max} (c.f. to [10] and [11] for more details).

From the dependency of α_{\max} on the structure of the camera graph, it can be established that the most favourable

case (admitting the largest residuals) is the complete graph for which $\alpha_{\max} \approx 42.9^\circ$. The other extreme case is a cycle with $\alpha_{\max} = \pi/n$, which induces a low angular bound for a large number of cameras although [10] suggested that this bound was “quite conservative”.

Although conditions were presented in terms of the angular distance, we remark that a chordal bound can also be established for the chordal residuals $\{d_{\text{chordal}}(R_j^* R_i^{*T}, \tilde{R}_{ij})\}$ of Problem 1 as both distances are related [16]:

$$d_{\text{chordal}}(R, S) = 2\sqrt{2} \sin \left(\frac{d_{\angle}(R, S)}{2} \right). \quad (5)$$

B.2. Zero duality gap between (P) and (DD)

Eriksson *et al.* [10] has proven that under mild conditions on the noise level (See Sec. B.1), there is zero duality gap between their primal problem (P_{orig}) and their SDP relaxation (DD_{orig}). Since we defined our primal problem (P) and its SDP relaxation (DD) following a different convention for the relative rotation definition than [10], here we show that our (P) and (DD) problems are equivalent to their counterparts in [10]. Hence the zero duality gap extends to them.

We defined our primal problem as follows. By rewriting the chordal distance using trace, (1) becomes (Eq. (8) in the main text)

$$\min_{R_1, \dots, R_n \in SO(3)} - \sum_{(i,j) \in \mathcal{E}} \text{tr}(R_j^T \tilde{R}_{ij} R_i). \quad (6)$$

By the transpose invariance of the trace, (6) is equivalent to

$$\min_{R_1, \dots, R_n \in SO(3)} - \sum_{(i,j) \in \mathcal{E}} \text{tr}(R_i^T \tilde{R}_{ij}^T R_j). \quad (7)$$

Our primal definition comes from rewriting (6) more compactly as

$$\min_{R \in SO(3)^n} -\text{tr}(R^T \tilde{R} R) \quad (\text{P})$$

using matrix notations, where

$$R = [R_1^T R_2^T \cdots R_n^T]^T \in SO(3)^n \quad (8)$$

contains the target variables, and \tilde{R} encodes the transposes of the relative rotations. \tilde{R} is then defined as

$$\tilde{R} = \begin{bmatrix} 0_3 & a_{12} R_{12}^T & \cdots & a_{1n} R_{1n}^T \\ a_{21} R_{21}^T & 0_3 & \cdots & a_{2n} R_{2n}^T \\ \vdots & & 0_3 & \vdots \\ a_{n1} R_{n1}^T & a_{n2} R_{n2}^T & \cdots & 0_3 \end{bmatrix}, \quad (9)$$

where a_{ij} are the elements of the adjacency matrix A of \mathcal{G} .

We now show that (P) is equivalent to the primal in [10], which is defined as (Eq. (11) in [10])

$$\min_{Q \in SO(3)^n} -\text{tr}(Q \tilde{Q} Q^T), \quad (\text{P}_{\text{orig}})$$

where Q is a ‘‘row’’ vector containing rotation matrices

$$Q = [Q_1, \dots, Q_n], \quad (10)$$

and \tilde{Q} encodes the relative measurements as

$$\tilde{Q} = \begin{bmatrix} 0_3 & a_{12} Q_{12} & \cdots & a_{1n} Q_{1n} \\ a_{21} Q_{21} & 0_3 & \cdots & a_{2n} Q_{2n} \\ \vdots & & 0_3 & \vdots \\ a_{n1} Q_{n1} & a_{n2} Q_{n2} & \cdots & 0_3 \end{bmatrix}. \quad (11)$$

However, relative rotations Q_{ij} in [10] are defined such that (Eq. (4) in [10])

$$Q_{ij} = Q_i^T Q_j. \quad (12)$$

Contrast to our definition from Eq. (1) in the main text where we define relative rotations in the ideal case as

$$R_{ij} = R_j R_i^T. \quad (13)$$

The following equivalences can then be established:

$$R_i = Q_i^T \text{ and } R_{ij} = Q_{ij}^T, \quad (14)$$

which implies that $Q = R^T$, $\tilde{Q} = \tilde{R}$, and therefore (P) is equivalent to (P_{orig}) in the sense that their objective values are the same and their optimisers are related by a translation.

Similarly, our SDP relaxation

$$\min_{Y \in \mathbb{R}^{3n \times 3n}} -\text{tr}(\tilde{R} Y) \quad (\text{DD})$$

$$\text{s.t. } Y_{i,i} = I_3, \quad i = 1, \dots, n. \quad (15a)$$

$$Y \succeq 0, \quad (15b)$$

is equivalent to its counterpart in [10]. In effect, they are the same as matrices encoding rotations are the same for both problems ($\tilde{Q} = \tilde{R}$).

B.3. Validity of Algorithm 1 as equivalent to BCD in Eriksson et al. [10]

Here we show that BCD as presented in Algorithm 1 in the main text is equivalent to the original BCD algorithm for rotation averaging proposed in [10]. To facilitate presentation, we call BCD-Ours to Algorithm 1 in the main text and BCD-Orig to Algorithm 1 in [10].

The improvement of BCD-Ours over BCD-Orig is that instead of creating a temporary large square matrix

$$B = \begin{bmatrix} Y_{(1:k-1);(1:k-1)}^{(t)} & Y_{(1:k-1);(k+1:n)}^{(t)} \\ Y_{(k+1:n);(1:k-1)}^{(t)} & Y_{(k+1:n);(k+1:n)}^{(t)} \end{bmatrix} \quad (16)$$

as in BCD-Orig, BCD-Ours creates a temporary vector which allows to operate directly on $Y^{(t)}$ as we will show next.

Note that B are the elements in $Y^{(t)}$ that are kept constant during the current iteration in BCD-Orig and BCD-Ours. On the other hand, the updated components for $Y^{(t)}$ in BCD-Orig are obtained from the optimiser X^* of an SDP problem (Problem (26) in the main text) which has the following explicit solution:

$$X^* = BC \left[(C^T BC)^{\frac{1}{2}} \right]^\dagger, \quad (17)$$

where $C \in \mathbb{R}^{3(n-1) \times 3}$ is the k -th column of \tilde{R} without its k -th row, i.e.,

$$C = \begin{bmatrix} \tilde{R}_{(1:k-1);(k:k)}^{(t)} \\ \tilde{R}_{(k+1:n);(k:k)}^{(t)} \end{bmatrix}. \quad (18)$$

Instead of computing the updates from (17), BCD-Ours solves

$$S = Z \left[(W^T Z)^{\frac{1}{2}} \right]^\dagger, \quad (19)$$

where $W \in \mathbb{R}^{3(n-1) \times 3}$ is the k -th column of \tilde{R} , i.e.,

$$W = \tilde{R}_{:,k}, \quad (20)$$

and

$$Z = Y^{(t)} W \quad (21)$$

is a temporary vector.

We will show next that X^* is equal to S without its k -th element. Since BCD-Ours ignores the k -th element of S during the update (Line 7 in BCD-Ours), BCD-Ours and BCD-Orig produce the same output.

Note first that the pseudo-inverse parts of (17) and (19) are the same since

$$C^T BC = W^T Z \quad (22)$$

as the k -th element in W is zero (W is the k -th column of \tilde{R} which has diagonal elements equal to 0_3). Similarly BC is equal to Z if removing the k -th element of Z . Hence (19) produces X^* after removing the k -th element of S .

C. Additional Results

C.1. Varying noise levels and number of cameras in SLAM graphs

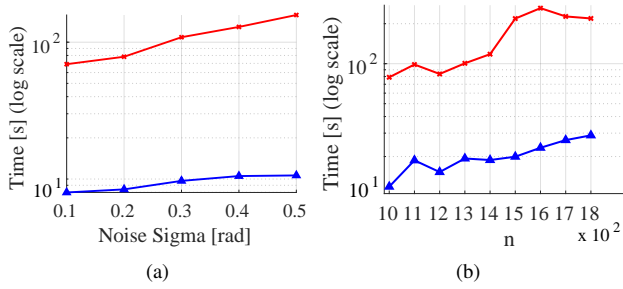


Figure 1. Runtime [s] (in log-scale) for SLAM camera graphs with $d_G = 0.2$. (a) Varying σ in $[0.1, 0.5]$ rad. and $n = 1000$. (b) Varying n in $[1000, 1800]$ with $\sigma = 0.1$ rad. See Fig. 3 in the main text for the description of the legend.

Chapter 5

Resolving Marker Pose Ambiguity by Robust Rotation Averaging with Clique Constraints

The work contained in this chapter has been published as the following paper

Shin-Fang Chng, Naoya Sogi, Pulak Purkait, Tat-Jun Chin and Kazuhiro Fukui:
Resolving Marker Pose Ambiguity by Robust Rotation Averaging with Clique Constraints. IEEE International Conference on Robotics and Automation (ICRA) 2020.

Statement of Authorship

Title of Paper	Resolving Marker Pose Ambiguity by Robust Rotation Averaging with Clique Constraints
Publication Status	<input checked="" type="checkbox"/> Published <input type="checkbox"/> Accepted for Publication <input type="checkbox"/> Submitted for Publication <input type="checkbox"/> Unpublished and Unsubmitted work written in manuscript style
Publication Details	Shin-Fang Chng, Naoya Sogi, Pulak Purkait, Tat-Jun Chin and Kazuhiro Fukui. "Resolving Marker Pose Ambiguity by Robust Rotation Averaging with Clique Constraints." IEEE International Conference on Robotics and Automation (ICRA) 2020.

Principal Author

Name of Principal Author (Candidate)	Shin-Fang Chng		
Contribution to the Paper	Proposed the main idea. Conducted experiments and collected onsite data. Wrote the paper.		
Overall percentage (%)	60		
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.		
Signature	_____	Date	6 January 2021

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

- i. the candidate's stated contribution to the publication is accurate (as detailed above);
- ii. permission is granted for the candidate to include the publication in the thesis; and
- iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Naoya Sogi		
Contribution to the Paper	Provided discussions and suggestions. Collected onsite data.		
Signature	_____	Date	6 January 2021

Name of Co-Author	Pulak Purkait		
Contribution to the Paper	Provided discussions and source code during the experiment.		
Signature	_____	Date	6th Jan 2021

Name of Co-Author	Tat-Jun Chin		
Contribution to the Paper	Provided major discussions and suggestions about the method and the experiments. Modified the draft.		
Signature		Date	21 Feb 2021

Name of Co-Author	Kazuhiro Fukui		
Contribution to the Paper	Provided suggestions to improve the draft.		
Signature		Date	6 January 2021

Resolving Marker Pose Ambiguity by Robust Rotation Averaging with Clique Constraints*

Shin-Fang Ch'ng¹, Naoya Sogi², Pulak Purkait¹, Tat-Jun Chin¹ and Kazuhiro Fukui²

Abstract—Planar markers are useful in robotics and computer vision for mapping and localisation. Given a detected marker in an image, a frequent task is to estimate the 6DOF pose of the marker relative to the camera, which is an instance of planar pose estimation (PPE). Although there are mature techniques, PPE suffers from a fundamental ambiguity problem, in that there can be more than one plausible pose solutions for a PPE instance. Especially when localisation of the marker corners is noisy, it is often difficult to disambiguate the pose solutions based on reprojection error alone. Previous methods choose between the possible solutions using a heuristic criterion, or simply ignore ambiguous markers.

We propose to resolve the ambiguities by examining the consistencies of a set of markers across multiple views. Our specific contributions include a novel rotation averaging formulation that incorporates long-range dependencies between possible marker orientation solutions that arise from PPE ambiguities. We analyse the combinatorial complexity of the problem, and develop a novel lifted algorithm to effectively resolve marker pose ambiguities, without discarding any marker observations. Results on real and synthetic data show that our method is able to handle highly ambiguous inputs, and provides more accurate and/or complete marker-based mapping and localisation.

I. INTRODUCTION

In many robotic vision pipelines, fiducial markers are often employed to simplify feature extraction. In particular, planar markers [1]–[6], which are designed to be easily detected and associated across images, find extensive use in laboratory and commercial settings (factories, warehouses, mines, etc.). In applications that perform planar marker-based SfM or SLAM [7]–[10], there is a basic need to estimate the 6DOF pose of an observed marker relative to the camera coordinate frame. This is often solved as a special case of planar pose estimation (PPE), which functions by determining the relative pose between a plane of known dimensions and its projection onto the image [11]–[13].

While in theory 6DOF pose can be determined uniquely from four non-colinear but co-planar points, the situation is less clear in non-ideal conditions where perspective effects are not apparent, e.g., when the imaged marker is small or the marker is at a distance which is significantly larger than the focal length. In such conditions there is a two-fold *rotational ambiguity* that corresponds to an unknown reflection of the plane about the z-axis of the camera [11]–[13]. For one observed planar marker (specifically its four corners), state-of-the-art PPE methods [12], [13] may return two physically

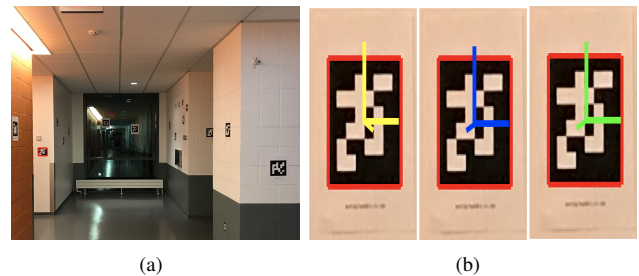


Fig. 1. (a) A detected marker with bounding box from a frame in the dataset of [9]. (b) The two poses \mathbf{p}' (yellow) and \mathbf{p}'' (blue) returned by PPE [13] have reprojection errors 0.00011 and 0.00013 resp. Though \mathbf{p}' has the lower error, it is an incorrect pose, cf. the ground truth pose (green).

plausible pose solutions, with one of them being the correct one (i.e., the one closer to the ground truth pose).

Fig. 1 shows an example from the dataset of [9]. Note that the two solutions returned by PPE can be very different, thus it is unwise to arbitrarily choose one of the two poses, or take the midpoint of the two solutions as the pose estimate.

A common way to disambiguate the two returned poses \mathbf{p}' and \mathbf{p}'' is to compute the reprojection error of each pose

$$r(\mathbf{p}) = \sum_{k=1}^4 \|f(\mathbf{K}, \mathbf{c}_k, \mathbf{p}) - \mathbf{u}_k\|_2^2, \mathbf{p} \in \{\mathbf{p}', \mathbf{p}''\} \quad (1)$$

where $\{\mathbf{c}_k\}_{k=1}^4$ and $\{\mathbf{u}_k\}_{k=1}^4$ are the reference 3D position and 2D observation of the 4 corners of the detected marker, \mathbf{K} is the camera intrinsic parameter and $f(\mathbf{K}, \mathbf{c}, \mathbf{p})$ projects \mathbf{c} onto the image with camera pose \mathbf{p} . The PPE pose with the lower reprojection error is then selected.

However, comparing reprojection errors is not fool-proof [10], [14], for if the corner localisation is noisy, $r(\mathbf{p}')$ and $r(\mathbf{p}'')$ can be very close. In fact, the correct solution can have the higher reprojection error; see Fig. 1.

In practice, marker pose ambiguity occurs regularly [8]. Fig. 2(a) is the histogram of the reprojection error ratio

$$\frac{\min[r(\mathbf{p}'), r(\mathbf{p}'')]}{\max[r(\mathbf{p}'), r(\mathbf{p}'')]} \quad (2)$$

of the PPE-derived poses for all the markers detected in sequence Hotel2(H2) from [15]. About 25% of the PPE solutions are considered ambiguous (ratio value ≥ 0.6 [8]).

While current theory and algorithms for PPE [12], [13] have characterised the ambiguity issue and are able to compute all physically plausible solutions stably, using the PPE outputs under ambiguity, particularly in marker-based SfM or SLAM pipelines, remains a fundamental challenge. In the following, we further survey efforts to deal with marker pose ambiguity, before outlining the proposed solution.

*This work was supported by the ARC Centre of Excellence on Robotic Vision CE140100016 and the Mawson Lakes Fellowship Program.

¹School of Computer Science, The University of Adelaide, Australia.

²Department of Computer Science, University of Tsukuba, Japan.

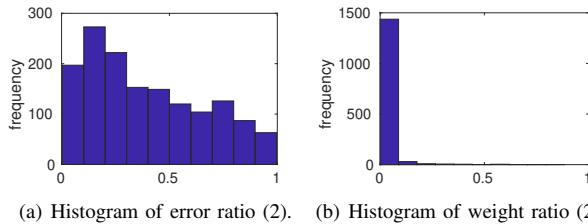


Fig. 2. Histogram of reprojection error ratio (2) and weight ratio (21) from proposed method (Sec. IV-C) for all markers detected in **Hotell2** [15].

A. Related work

Tanaka et al. [16], [17] modified the conventional planar marker design to directly incorporate orientation information. They attach two one-dimensional moire patterns onto the marker to obtain appearance variation for pose disambiguation, as well as lenticular lenses that introduce 3D deviations to the marker surface. Though this largely alleviates the ambiguity problem, the marker fabrication is non-trivial.

For planar target camera tracking, a filtering method with a well-tuned camera motion model [14], [18] can be exploited to disambiguate the marker poses. However, this assumes temporal continuity in the images, which may not be valid in SfM with wide baseline images; moreover, there are no mature *filtering methods* for marker SLAM. Jin et al. [19] showed improved marker pose estimation accuracy by fusing depth information. However, this requires an RGBD camera.

Marker-based SfM/SLAM is an active research area [7]–[10], [20]. Marker ambiguity is not dealt with explicitly in [7], [9], [20], though [9] combined feature-based SfM with marker-based SfM. Munoz-Salinas et al. applied the ratio test of [13] in their marker-based SfM [8] and SLAM pipeline [10]. Basically, if the ratio (2) is below a threshold (default is 0.6 [8]), the PPE solution with the lower reprojection error is used in subsequent SfM/SLAM processing; else, the marker detection is discarded. A weakness of this approach is the sensitivity to the threshold. If it is too low, many marker detections will be excluded, leading to data wastage or even SfM/SLAM failure. Contrarily, a high threshold risks using bad marker poses (recall that the pose with the lower reprojection error may not be the correct one) for SfM/SLAM. Sec. VI will demonstrate this shortcoming.

B. Our contributions

Unlike previous works that have used a *per-marker* approach to resolve marker ambiguity, we exploit *multi-view constraints* for disambiguation. From the input marker detections, we first construct a *multigraph* of relative rotation measurements, which incorporates all PPE pose ambiguities. Then, we formulate a novel rotation averaging problem with clique constraints that respects *consistency* (details later) between subsets of relative pose measurements. We examine the combinatorial complexity of the new problem, and develop a lifted optimisation method to efficiently solve it. Then, a series of small maximal weighted clique problems are solved to make the final pose selections. Our method

allows all valid PPE pose combinations to be examined, and leads to more accurate and/or complete marker-based SfM.

II. PROBLEM FORMULATION

Consider T input images $\{I_t\}_{t=1}^T$ that observed a set of N markers $\{\mathcal{M}_i\}_{i=1}^N$ of known sizes in a static scene. We assume calibrated cameras. A standard marker detection and id algorithm [4], [21] is applied to each image. Denote by

$$\mathcal{A}^t = \{i \in \{1, \dots, N\} \mid \mathcal{M}_i \text{ was detected in } I_t\} \quad (3)$$

as the set of markers detected in I_t . Using a PPE technique [12], [13] on the corners of \mathcal{M}_i detected in I_t , the *marker-to-camera* (M2C) relative pose of \mathcal{M}_i to I_t is computed, which can potentially yield two solutions

$$\{\tilde{\mathbf{p}}_i^{(t,0)}, \tilde{\mathbf{p}}_i^{(t,1)}\} = \{\tilde{\mathbf{p}}_i^{(t,a)}\}_{a=0,1}. \quad (4)$$

Without loss of generality, we assume that each marker observation has exactly two relative pose solutions. Note that the pose ambiguity is due to orientation ambiguity, thus the translation component is the same, i.e.,

$$\tilde{\mathbf{p}}_i^{(t,0)} = (\tilde{\mathbf{t}}_i^{(t)}, \tilde{\mathbf{R}}_i^{(t,0)}), \quad \tilde{\mathbf{p}}_i^{(t,1)} = (\tilde{\mathbf{t}}_i^{(t)}, \tilde{\mathbf{R}}_i^{(t,1)}). \quad (5)$$

Given the set of all M2C relative pose measurements

$$\bigcup_{t=1}^T \bigcup_{i \in \mathcal{A}^t} \{\tilde{\mathbf{p}}_i^{(t,a)}\}_{a=0,1}, \quad (6)$$

our overall aim is SfM, i.e., find the absolute poses of the markers $\{\mathbf{p}_i\}_{i=1}^N$ and cameras $\{\mathbf{q}_t\}_{t=1}^T$. To do so, pose ambiguity must be resolved, i.e., for each (i, t) such that $i \in \mathcal{A}^t$, choose *either* $\tilde{\mathbf{p}}_i^{(t,0)}$ *or* $\tilde{\mathbf{p}}_i^{(t,1)}$ for SfM computations.

Previous pipelines [8], [10] make the choice using per-marker heuristics, or discard the marker observation. This “preprocessing” yields the *reduced measurement set*

$$\bigcup_{t=1}^T \bigcup_{i \in \mathcal{B}^t} \{\tilde{\mathbf{p}}_i^{(t)}\}, \quad (7)$$

where each $\tilde{\mathbf{p}}_i^{(t)}$ is *either* $\tilde{\mathbf{p}}_i^{(t,0)}$ *or* $\tilde{\mathbf{p}}_i^{(t,1)}$, and $\mathcal{B}^t \subseteq \mathcal{A}^t$. The reduced measurement set is then subjected to the rest of the SfM/SLAM pipeline. Our new method exploits multi-view consistency to disambiguate the PPE marker poses in a way that avoids premature decisions; details as follows.

III. MULTIGRAPH WITH ROTATIONAL AMBIGUITY

Since the ambiguity lies in the orientations, it is natural to model the ambiguity using only the M2C relative rotations

$$\bigcup_{t=1}^T \bigcup_{i \in \mathcal{A}^t} \{\tilde{\mathbf{R}}_i^{(t,a)}\}_{a=0,1}. \quad (8)$$

To this end, we construct a multigraph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where the vertices \mathcal{V} is the set of markers $\{1, \dots, N\}$, and the edges \mathcal{E} indicate covisibility between the markers. More specifically, if \mathcal{M}_i and \mathcal{M}_j are detected in I_t , four edges

$$\langle i, j \rangle^{(t,00)}, \langle i, j \rangle^{(t,01)}, \langle i, j \rangle^{(t,10)}, \langle i, j \rangle^{(t,11)} \quad (9)$$

connect vertices i and j in \mathcal{G} ; assuming $i < j$, the edges correspond to the *marker-to-marker (M2M)* relative rotations

$$\begin{aligned}\tilde{\mathbf{R}}_{i,j}^{(t,00)} &= (\tilde{\mathbf{R}}_j^{(t,0)})^T \tilde{\mathbf{R}}_i^{(t,0)}, & \tilde{\mathbf{R}}_{i,j}^{(t,01)} &= (\tilde{\mathbf{R}}_j^{(t,1)})^T \tilde{\mathbf{R}}_i^{(t,0)}, \\ \tilde{\mathbf{R}}_{i,j}^{(t,10)} &= (\tilde{\mathbf{R}}_j^{(t,0)})^T \tilde{\mathbf{R}}_i^{(t,1)}, & \tilde{\mathbf{R}}_{i,j}^{(t,11)} &= (\tilde{\mathbf{R}}_j^{(t,1)})^T \tilde{\mathbf{R}}_i^{(t,1)}.\end{aligned}\quad (10)$$

Fig. 3 shows an example. Since multiple edges connect two vertices, \mathcal{G} is a multigraph. We summarise (9) and (10) as

$$\left\{ \langle i, j \rangle^{(t,ab)} \right\}_{ab=00,01,10,11}, \quad \left\{ \tilde{\mathbf{R}}_{i,j}^{(t,ab)} \right\}_{ab=00,01,10,11}, \quad (11)$$

where ab is a bit string composed of two binary indicators $a, b \in \{0, 1\}$. The edges in \mathcal{G} are undirected; if $i < j$, the edge $\langle j, i \rangle^{(t,ab)}$ has the associated M2M relative rotation

$$\tilde{\mathbf{R}}_{j,i}^{(t,ab)} = (\tilde{\mathbf{R}}_j^{(t,a)})^T \tilde{\mathbf{R}}_i^{(t,b)}. \quad (12)$$

Thus, in our notation

$$\langle i, j \rangle^{(t,ab)} = \langle j, i \rangle^{(t,ba)} \neq \langle j, i \rangle^{(t,ab)}. \quad (13)$$

The set of all edges \mathcal{E} (without repetitions) is thus

$$\mathcal{E} = \bigcup_{t=1}^T \bigcup_{\substack{i,j \in \mathcal{A}^t \\ i < j}} \left\{ \langle i, j \rangle^{(t,ab)} \right\}_{ab=00,01,10,11}. \quad (14)$$

Similarly, the set of unique M2M relative rotations is

$$\bigcup_{t=1}^T \bigcup_{\substack{i,j \in \mathcal{A}^t \\ i < j}} \left\{ \tilde{\mathbf{R}}_{i,j}^{(t,ab)} \right\}_{ab=00,01,10,11}. \quad (15)$$

The existence of four M2M relative rotations per $\langle i, j \rangle$ pair is a direct consequence of ambiguity in marker pose estimation, and the bit string ab selects a particular combination of M2C relative rotations to derive the M2M relative rotation.

Note that our multigraph construction method is a significant extension of that in [8], in that our multigraph incorporates all ambiguous marker poses, whereas [8] generates \mathcal{G} from the preprocessed data (7) with no ambiguities.

A. Consistent cliques

We assume that the multigraph \mathcal{G} is connected, i.e., there is a path that connects every pair of vertices (markers) in \mathcal{G} .

Definition 1 (*Consistent clique*) Given multigraph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ as defined above, a consistent clique for image I_t is a fully connected subgraph $\mathcal{C} = \{\mathcal{V}', \mathcal{E}'\}$ such that

- $\mathcal{V}' = \mathcal{A}^t \subseteq \mathcal{V}$;
- Every two vertices $i, j \in \mathcal{V}'$ are connected by **exactly** one edge $\langle i, j \rangle^{(t,ab)}$, where ab is one of $\{00, 01, 10, 11\}$.
- For every two vertices $j, k \in \mathcal{V}'$ that are connected to vertex i , the associated edges $\langle i, j \rangle^{(t,ab)}$ and $\langle i, k \rangle^{(t,cd)}$ satisfy the condition $a = c$.

Fig. 3 provides examples. Intuitively, a consistent clique \mathcal{C} for image I_t corresponds to a set of M2M relative rotations that are composed using a constant selection of one of the two M2C relative poses for each marker detected in I_t .

Since there are multiple valid combinations of constant M2C relative pose selections, there are multiple consistent

cliques for an image. Assuming that V markers are detected in each image, there are $\mathcal{O}(2^V)$ number of consistent cliques per image. For T images, there are thus $\mathcal{O}(2^{VT})$ unique combinations of consistent cliques across the images.

IV. DISAMBIGUATION WITH ROTATION AVERAGING

Based on the multigraph, our technique resolves the ambiguities by first solving a novel rotation averaging formulation, then - based on the averaging results - building and solving a maximum weighted clique problem. The key outcome of this step is marker pose disambiguation; Sec. V will incorporate this step into a marker-based SfM pipeline.

A. Rotation averaging with clique constraints

While standard rotation averaging is defined over a graph of relative rotations [22], [23], extending the formulation to a multigraph of relative rotations is straightforward, and existing algorithms (we used [23]) can be applied with minor adjustments. Let $\{\mathbf{R}_i\}_{i=1}^N$ be the absolute rotations of the markers. A rotation averaging problem over multigraph \mathcal{G} is

$$\min_{\{\mathbf{R}_i\}_{i=1}^N} \sum_{t=1}^T \sum_{\substack{i,j \in \mathcal{A}^t \\ i < j}} \sum_{a,b \in \{0,1\}} \rho \left(\left\| \tilde{\mathbf{R}}_{i,j}^{(t,ab)} - \mathbf{R}_j \mathbf{R}_i^T \right\|_F \right), \quad (16)$$

where ρ is a robust norm. The motivation behind (16) is to attempt to identify the incorrect poses from PPE as the contributors to outlying measurements in the averaging task.

However, our tests (Sec. VI) suggest that this approach is ineffective for disambiguation, most probably because (16) does not enforce clique consistency (Def. 1). Thus, error terms that are regarded as inliers could correspond to choosing *both* PPE poses for the same marker detection.

To enforce clique consistency into rotation averaging, we introduce a set of binary indicator variables

$$\mathcal{S} = \bigcup_{t=1}^T \{s_i^t \in \{0, 1\} \mid i \in \mathcal{A}^t\}, \quad (17)$$

where the setting $s_i^t = 0$ implies selecting M2C relative rotation $\tilde{\mathbf{R}}_i^{(t,0)}$ the detection of \mathcal{M}_i in I_t , while $s_i^t = 1$ implies selecting $\tilde{\mathbf{R}}_i^{(t,1)}$. We then formulate the clique-constrained rotation averaging problem

$$\begin{aligned}\min_{\{\mathbf{R}_i\}_{i=1}^N, \mathcal{S}} \sum_{t=1}^T \sum_{\substack{i,j \in \mathcal{A}^t \\ i < j}} s_i^t s_j^t \left\| \tilde{\mathbf{R}}_{i,j}^{(t,11)} - \mathbf{R}_j \mathbf{R}_i^T \right\|_F + \\ s_i^t (1 - s_j^t) \left\| \tilde{\mathbf{R}}_{i,j}^{(t,10)} - \mathbf{R}_j \mathbf{R}_i^T \right\|_F + \\ (1 - s_i^t) s_j^t \left\| \tilde{\mathbf{R}}_{i,j}^{(t,01)} - \mathbf{R}_j \mathbf{R}_i^T \right\|_F + \\ (1 - s_i^t) (1 - s_j^t) \left\| \tilde{\mathbf{R}}_{i,j}^{(t,00)} - \mathbf{R}_j \mathbf{R}_i^T \right\|_F.\end{aligned}\quad (18)$$

Intuitively, \mathcal{S} selects the M2C relative rotations to compose the M2M relative rotations in a consistent way. Searching over \mathcal{S} thus allows different consistent cliques in all images to be examined. Finally, since $\{\mathbf{R}_i\}_{i=1}^N$ are shared across images, multi-view consistency is exploited to choose the best combinations of the PPE relative rotations.

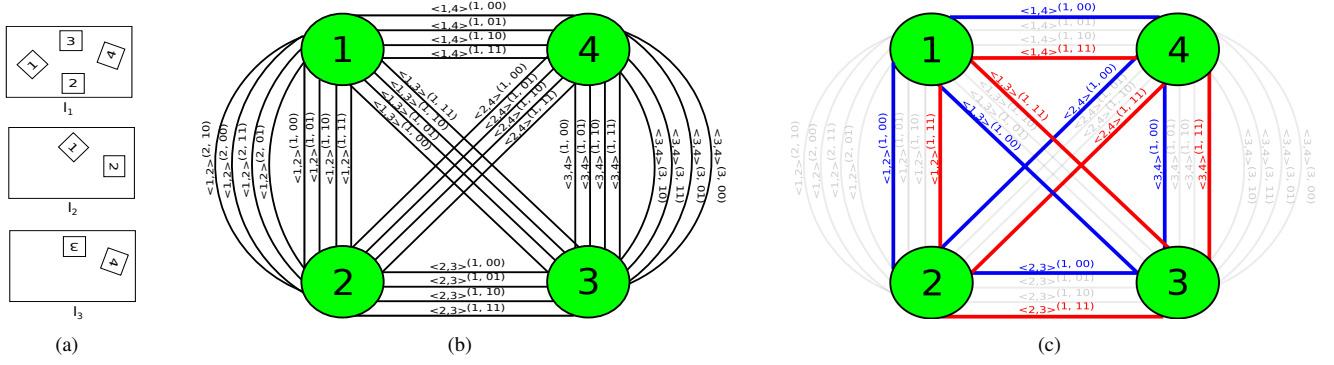


Fig. 3. Multigraph and consistent cliques. (a) The scene has 4 markers \$\{\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4\}\$ captured in 3 images \$\{I_1, I_2, I_3\}\$. All markers were detected in \$I_1\$, while only a subset was detected in \$I_2\$ and \$I_3\$. (b) Multigraph with the edges labelled following (9). Since \$\mathcal{M}_1\$ and \$\mathcal{M}_2\$ were covisible in \$I_1\$ and \$I_2\$, there are 8 edges connecting vertices 1 and 2 (similarly, \$\mathcal{M}_3\$ and \$\mathcal{M}_4\$ in \$I_1\$ and \$I_3\$). (c) Two consistent cliques (red and blue) for image \$I_1\$.

B. Efficient algorithm using lifting approach

A naive method to solve (18) is to enumerate \$\mathcal{S}\$, and for each \$\mathcal{S}\$ instantiation, collect the non-zero terms in (18) and solve the resulting rotation averaging problem. Then, return the \$\mathcal{S}\$ with the lowest optimised error as the disambiguation decision. Since there are \$\mathcal{O}(2^{2V})\$ possible instantiations of \$\mathcal{S}\$ (assuming \$V\$ markers seen per image), this is infeasible.

To enable an efficient algorithm for (18), we apply the lifting approach [24]. First, we relax the indicator variables \$s_i^t \in [0, 1]\$ and replace them in (18) with a sigmoid function

$$\Phi(s) = 1/(1 + e^{-s}), \quad (19)$$

which yields the “smoothed” version of (18)

$$\begin{aligned} \min_{\{\mathbf{R}_i\}, \mathcal{S}} \sum_{t=1}^T \sum_{\substack{i,j \in \mathcal{A}^t \\ i < j}} \Phi(s_i^t) \Phi(s_j^t) & \left\| \tilde{\mathbf{R}}_{i,j}^{(t,11)} - \mathbf{R}_j \mathbf{R}_i^T \right\|_F + \\ \Phi(s_i^t) (1 - \Phi(s_j^t)) & \left\| \tilde{\mathbf{R}}_{i,j}^{(t,10)} - \mathbf{R}_j \mathbf{R}_i^T \right\|_F + \\ (1 - \Phi(s_i^t)) \Phi(s_j^t) & \left\| \tilde{\mathbf{R}}_{i,j}^{(t,01)} - \mathbf{R}_j \mathbf{R}_i^T \right\|_F + \\ (1 - \Phi(s_i^t)) (1 - \Phi(s_j^t)) & \left\| \tilde{\mathbf{R}}_{i,j}^{(t,00)} - \mathbf{R}_j \mathbf{R}_i^T \right\|_F. \end{aligned} \quad (20)$$

Intuitively, the contribution of an error term in (20) is now weighted according to correctness of the corresponding M2C relative poses that define the error term.

Problem (20) can be solved using an iterative non-linear optimiser (e.g., *fmincon* in MATLAB). We initialise \$\{\mathbf{R}_i\}\$ via a minimum spanning tree on \$\mathcal{G}\$, choosing the M2M relative rotations with the lower combined reprojection errors for chaining, and \$\mathcal{S}\$ is set to reflect these choices. As we will show in Sec. VI, our method is not biased by such an initialisation, since it is capable of providing more accurate disambiguation than comparing reprojection errors alone.

C. Selecting the marker poses

Let \$\hat{\mathcal{S}}\$ by the optimised relaxed indicator variables from solving (20). For the same sequence used in Fig. 2(a), we plot in Fig. 2(b) the histogram of the ratios

$$\frac{\min(\Phi(\hat{s}_i^t), 1 - \Phi(\hat{s}_i^t))}{\max(\Phi(\hat{s}_i^t), 1 - \Phi(\hat{s}_i^t))} \quad (21)$$

for all \$\hat{s}_i^t \in \hat{\mathcal{S}}\$. Similar to (2), the ratio (21) indicates how “disambiguable” the PPE poses are for each marker detection (smaller ratios are better), but now based on the value of \$\hat{s}_i^t\$. Although \$\hat{\mathcal{S}}\$ is not discrete, the percentage of marker poses that are still ambiguous is now significantly reduced.

To conclusively select one PPE pose per detected marker, a simple solution would be to threshold each \$\hat{s}_i^t \in \hat{\mathcal{S}}\$ with 0.5; however, we would like to avoid such a per-marker decision. To this end, for each image \$I_t\$ we construct the multigraph \$\mathcal{G}_t = \{\mathcal{V}_t, \mathcal{E}_t\}\$, where \$\mathcal{V}_t = \mathcal{A}^t\$, and

$$\mathcal{E}_t = \left\{ \langle i, j \rangle^{(t,ab)} \mid i, j \in \mathcal{A}^t, ab \in \{00, 01, 10, 11\} \right\}. \quad (22)$$

Note that \$\mathcal{G}_t\$ is a submultigraph of \$\mathcal{G}\$, and there exist \$\mathcal{O}(2^V)\$ consistent cliques in \$\mathcal{G}_t\$ (see Sec. III-A). Further, each edge \$\langle i, j \rangle^{(t,ab)}\$ in \$\mathcal{G}_t\$ has the weight

$$\hat{w}_{i,j}^{(t,ab)} = \begin{cases} (1 - \Phi(\hat{s}_i^t))(1 - \Phi(\hat{s}_j^t)) & \text{if } ab = 00; \\ (1 - \Phi(\hat{s}_i^t))\Phi(\hat{s}_j^t) & \text{if } ab = 01; \\ \Phi(\hat{s}_i^t)(1 - \Phi(\hat{s}_j^t)) & \text{if } ab = 10; \\ \Phi(\hat{s}_i^t)\Phi(\hat{s}_j^t) & \text{if } ab = 11. \end{cases} \quad (23)$$

Given \$\mathcal{G}_t\$, define edge indicator variables

$$\mathcal{Z}_t = \left\{ z_{i,j}^{(t,ab)} \in \{0, 1\} \mid i, j \in \mathcal{A}^t, ab \in \{00, 01, 10, 11\} \right\}.$$

and the maximum weighted clique (MWC) problem

$$\max_{\mathcal{Z}_t} \sum_{\substack{i,j \in \mathcal{A}^t \\ i < j}} \sum_{ab \in \{00, 01, 10, 11\}} z_{i,j}^{(t,ab)} \hat{w}_{i,j}^{(t,ab)} \quad (MWC_t)$$

$$\text{s.t. } \{ \langle i, j \rangle^{(t,ab)} \mid z_{i,j}^{(t,ab)} = 1 \} \text{ is consistent.}$$

Basically, the aim of \$MWC_t\$ is to find a consistent clique in \$I_t\$ with the largest edge weights. Though MWC is intractable in general [25], each \$MWC_t\$ instance is small, since the number \$V\$ of detected markers in \$I_t\$ is small (usually \$V \le 9\$).

We use the efficient clique solver of [26] on each \$MWC_t\$. The optimised \$\hat{\mathcal{Z}}_t\$ provides a consistent selection of the PPE poses for all markers detected in \$I_t\$. Specifically, for each \$\mathcal{M}_i\$ detected in \$I_t\$, find a \$\hat{z}_{i,j}^{(t,ab)}\$ that is nonzero, and set \$\tilde{\mathbf{p}}_i^{(t)} = \tilde{\mathbf{p}}_i^{(t,0)}\$ if \$a = 0\$, or \$\tilde{\mathbf{p}}_i^{(t)} = \tilde{\mathbf{p}}_i^{(t,1)}\$ otherwise.

Algorithm 1 summarises the proposed method for marker pose disambiguation.

Algorithm 1 Method for marker pose disambiguation

Input: M2C relative poses (6) with PPE ambiguity.

- 1: Construct a multigraph \mathcal{G} from the input (Sec. III).
- 2: $\{\hat{\mathbf{R}}_i\}, \{\hat{s}_i^t\} \leftarrow$ Solve (20) based on \mathcal{G} (Sec. IV-B).
- 3: **for** $t = 1, \dots, T$ **do**
- 4: $\{\hat{z}_{i,j}^{(t,ab)}\} \leftarrow$ Solve MWC_t from $\{\hat{s}_i^t\}$ (Sec. IV-C).
- 5: $\{\hat{\mathbf{p}}_i^{(t)}\} \leftarrow$ Based on $\{\hat{z}_{i,j}^{(t,ab)}\}$, select one of two M2C poses for all markers in I_t (Sec. IV-C).

Output: One M2C relative pose per detected marker.

V. MARKER-BASED SFM PIPELINE

To carry out marker-based SfM using our marker pose disambiguation method, we largely follow the pipeline of the state-of-the-art MarkerMapper [8]. Briefly, a robust pose graph optimisation is first invoked on the resolved M2C relative poses (7) from Algorithm 1 to yield absolute marker poses $\{\mathbf{p}_i\}_{i=1}^N$ - in our case, the absolute rotation component is initialised using the output $\{\hat{\mathbf{R}}_i\}$ from solving (20). Then, each camera pose \mathbf{q}_t is initialised using single pose averaging from the M2C poses, before all marker $\{\mathbf{p}_i\}_{i=1}^N$ and camera poses $\{\mathbf{q}_t\}_{t=1}^T$ are refined simultaneously by bundle adjustment on the observed corners of all detected markers. We refer to [8] for details of the SfM pipeline.

VI. RESULTS

To assess the efficacy of the proposed marker pose disambiguation technique, we compared the following methods:

- **Reprojection error (M1):** For each marker detection, select the PPE solution with the lower reprojection error.
- **Strict ratio test (M2):** The threshold of 0.1 is applied on the reprojection error ratio (2) (see Sec. I-A for details).
- **Default ratio test (M3):** The threshold of 0.6 is applied on the reprojection error ratio (the default setting in [8]).
- **Robust rotation averaging and post hoc clique consistency enforcement (M4):** Solve (16) by IRLS [23], then use the IRLS-optimised weights for the error terms as inputs to our M2C pose selection method in Sec. IV-C.
- **Proposed method (Ours):** As described in Sec. IV.

When applying the above disambiguation methods to per-form marker-based SfM, we simply used them to preprocess the input marker detections, then execute the rest of the pipeline of MarkerMapper [8] (see Sec. V). All the experiments were conducted on a 3.5GHz CPU and 8GB of RAM.

A. Experiments on hybrid data

1) *Data generation:* We used the **ScanNet Dataset** [15] that contained a number of sequences with ground truth 6DOF camera poses and depth. A test sequence was created from an original sequence by warping a number of ArUco markers [4], [5] based on known/ground truth M2C relative poses $\bar{\mathbf{p}}_i^{(t)}$ onto parts of the images that correspond to planar surfaces; see supplementary video for a sample sequence. Using the ground truth camera absolute pose $\bar{\mathbf{q}}_t$, the ground truth marker absolute pose is $\bar{\mathbf{p}}_i = \bar{\mathbf{q}}_t^{-1} \bar{\mathbf{p}}_i^{(t)}$.

2) *Marker detection:* Using the steps above, we generated five testing sequences from Bedroom(**B**), Hotel1(**H1**), Hotel2(**H2**), Office1(**O1**) and Office2(**O2**). We used [4] to detect, identify and localise the corners of each marker in each frame; see Table I for the number of frames and unique detected markers in each sequence. Though the markers were synthetically warped into the images, our analysis suggests that corner localisation suffered from errors of 1–7 pixels.

3) *Ground truth M2C pose selection:* On the noisy corner localisations, PPE [13] is invoked, which yields two M2C relative poses $\{\tilde{\mathbf{p}}_i^{(t,a)}\}_{a=0,1}$ for each detected marker. To decide the ground truth selection, we compute the angular difference $\{\theta_i^{(t,a)}\}_{a=0,1}$ between $\{\tilde{\mathbf{R}}_i^{(t,a)}\}_{a=0,1}$ and $\tilde{\mathbf{R}}_i^{(t)}$ as $\theta_i^{(t,a)} = \frac{180}{\pi} \arccos(1 - 0.25 \|\mathbf{I} - \tilde{\mathbf{R}}_i^{(t,a)} (\tilde{\mathbf{R}}_i^{(t)})^T\|_F)$. The ground truth selection of the PPE poses is taken as the one with the lower angular difference $\min\{\theta_i^{(t,a)}\}_{a=0,1}$.

4) *Results:* For the hybrid data experiment, we evaluated all the approaches on two main aspects; see supplementary video for demonstration of our pose disambiguation method.

a) *Precision in pose disambiguation:* For each testing sequence, precision in pose disambiguation is defined as $\frac{\text{\# number of correct PPE pose selections}}{\text{\# marker detections where a pose disambiguation decision was made}}$. Table I shows that **Ours** generally has higher precision than the others. The fact that **M4** (the control method) is much poorer than **Ours** proves that enforcing the proposed clique-consistency is crucial for disambiguating the PPE poses. Amongst the per-marker disambiguation methods (**M1–M3**), **M1** has the lowest precision, validating observations in previous works that comparing reprojection errors alone is not foolproof. Adding a ratio test to avoid decisions on cases that are too ambiguous helps to improve precision in **M2** and **M3**. In particular, the precision of **M2** is on par with **Ours**. However, as we show next, this gain by **M2** comes at a cost.

b) *Completeness and accuracy of SfM:* To assess the effects of marker pose disambiguation on SfM, we evaluate

- the number of markers mapped and cameras localised; and
- the error (in deg and cm) of the marker and camera poses estimated by marker-based SfM from the disambiguated PPE poses in Table I,II respectively. Although **M2** is precise, it yields a much sparser map than the others; moreover, as it has pruned away many useful detections, there are insufficient data to allow accurate SfM. Using our pose disambiguation technique leads to more complete and accurate maps.

B. Real world dataset experiment

Testing was performed on sequences from [9]. We selected 3 indoor scenes with different difficulty levels: *ece floor 4 wall*, *ece floor5 stairs* and *cee night cw*. There are $N \geq 50$ unique markers placed in the scene in each sequence. To enable comparisons, we invoked [9] (denoted as **FM**) which conducts both feature- and marker-based SfM on the sequences. Since SfM with **M2** failed in all 3 sequences due to insufficient data for optimisation, comparison is not made.

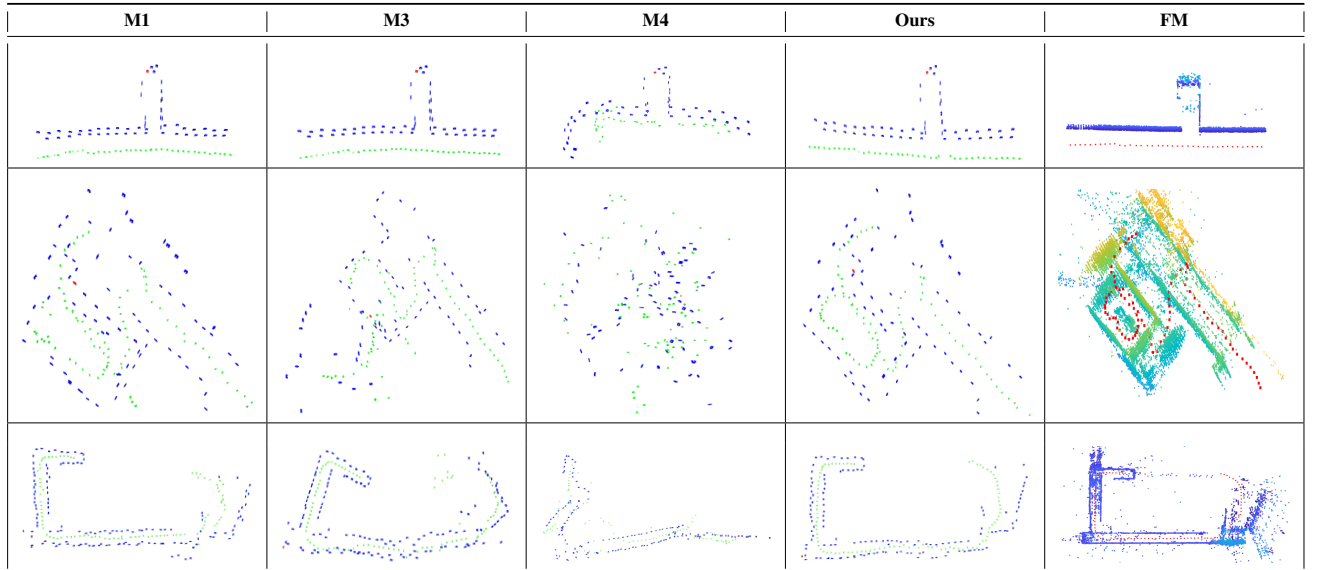
Qualitative results in Table III show that **Ours** is more accurate than **M1** and **M3** in marker-based SfM - of course, **Ours** is visibly not as complete as **FM**, but the latter uses

TABLE I
 PRECISION IN POSE DISAMBIGUATION ON HYBRID DATA.

Seq	N	T	Precision(%)					# markers mapped					# cameras localised				
			M1	M2	M3	M4	Ours	M1	M2	M3	M4	Ours	M1	M2	M3	M4	Ours
B	3	31	94.32	100	92.31	31.82	100	3	0	3	3	3	31	0	31	31	31
H1	5	41	80.68	100	82.61	22.16	100	5	0	5	5	5	41	0	40	41	41
O1	7	51	77.08	96.97	78.8	14.58	96.52	7	7	7	7	7	51	41	51	51	51
O2	6	91	92.64	100	98.95	37.94	99.41	6	4	6	6	6	91	46	91	91	91
H2	14	151	93.42	98.94	97.89	48.16	100	14	13	14	14	14	151	101	151	151	151

 TABLE II
 SFM ACCURACY FOR DIFFERENT POSE DISAMBIGUATION METHODS ON HYBRID DATA.

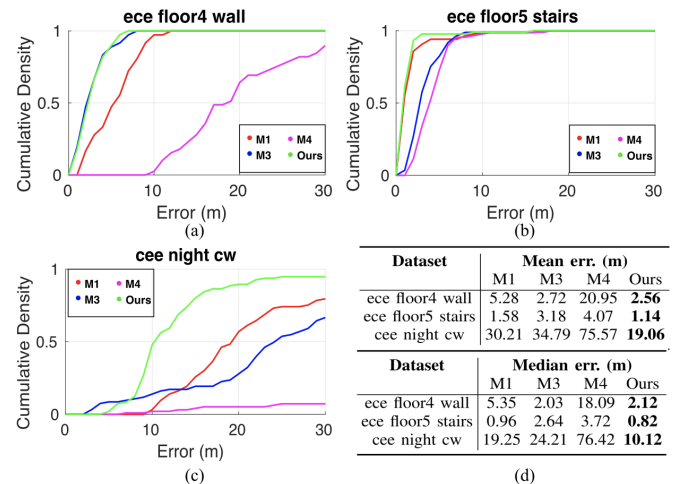
Seq	Average marker pose error ($^{\circ}$, cm)									Average camera pose error ($^{\circ}$, cm)										
	M1		M2		M3		M4		Ours		M1		M2		M3		M4		Ours	
B	5.4	11.7	-	-	6.3	15.0	19.0	37.5	2.3	2.2	7.0	15.9	-	-	11.9	19.5	32.0	10.0	0.8	2.0
H1	11.7	13.0	-	-	12.5	15.0	39.1	26.3	3.3	8.6	14.8	27.5	-	-	17.6	41.6	37.9	28.8	5.0	3.2
O1	26.2	30.3	15.2	8.0	25.4	29.0	55.3	120.9	3.5	4.3	17.3	69.8	7.6	16.0	19.2	69.4	85.8	49.7	5.7	13.7
O2	8.7	6.6	4.4	4.2	4.1	2.6	28.0	63.2	4.2	2.4	6.2	10.5	0.8	2.4	17.4	4.0	41.6	40.1	1.3	3.4
H2	4.3	5.1	7.7	3.1	5.4	5.5	20.3	14.2	3.6	4.9	4.3	3.8	2.2	2.3	3.3	3.1	32.0	10.0	3.4	2.4

 TABLE III
 QUALITATIVE RESULT: RECONSTRUCTION RESULTS FOR MARKER-BASED SFM METHODS **M1**, **M3**, **M4**, AND **Ours**, AS WELL AS FEATURE- AND MARKER-BASED SFM METHOD **FM** [9]. ROW 1: *ece floor4 wall*, ROW 2: *ece floor5 stairs*, ROW 3: *cee night cw*. FOR THE MARKER-BASED METHODS, RED = RECONSTRUCTED REFERENCE MARKER, BLUE: RECONSTRUCTED MARKERS, GREEN: ESTIMATED CAMERA POSITIONS.


features on top of markers, which entails heavier computations. Using the estimated camera positions by **FM** as reference, we obtain the position errors (in m) computed by the marker-based SfM methods - normalised and plotted as a cumulative density in Fig. 4a-c. It is apparent that **Ours** is much more accurate in camera localisation, especially in the most challenging sequence *cee night cw*. Fig 4d lists the mean and median position error, relative to **FM**.

VII. CONCLUSION

This work addresses the practically crucial marker pose ambiguities by inspecting the consistencies of a set of markers across multi-view, which enables the formulation of *clique-constrained rotation averaging*. Future work will be extending the current method in a sliding window fashion for real-time compliant robotics applications.


 Fig. 4. Comparison of camera position error (relative to **FM**) of **M1**, **M3**, **M4** and **Ours** in terms of (a-c) Cumulative density. (d) Mean & Median.

REFERENCES

- [1] J. Wang and E. Olson, "AprilTag 2: Efficient and robust fiducial detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4193–4198.
- [2] M. Fiala, "ARTag, an improved marker system based on artoolkit," *National Research Council Canada, Publication Number: NRC*, vol. 47419, p. 2004, 2004.
- [3] M. Fiala, "ARTag, a fiducial marker system using digital techniques," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2. IEEE, 2005, pp. 590–596.
- [4] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.
- [5] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image and vision Computing*, vol. 76, pp. 38–47, 2018.
- [6] D. Hu, D. DeTone, and T. Malisiewicz, "Deep ChArUco: Dark ChArUco Marker Pose Estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8436–8444.
- [7] K. Shaya, A. Mavrinac, J. L. A. Herrera, and X. Chen, "A self-localization system with global error reduction and online map-building capabilities," in *International Conference on Intelligent Robotics and Applications*. Springer, 2012, pp. 13–22.
- [8] R. Muñoz-Salinas, M. J. Marín-Jiménez, E. Yeguas-Bolívar, and R. Medina-Carnicer, "Mapping and localization from planar markers," *Pattern Recognition*, vol. 73, pp. 158–171, 2018.
- [9] J. DeGol, T. Bretl, and D. Hoiem, "Improved structure from motion using fiducial marker matching," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 273–288.
- [10] R. Muñoz-Salinas, M. J. Marín-Jiménez, and R. Medina-Carnicer, "SPM-SLAM: Simultaneous localization and mapping with squared planar markers," *Pattern Recognition*, vol. 86, pp. 156–171, 2019.
- [11] D. Oberkampf, D. F. DeMenthon, and L. S. Davis, "Iterative pose estimation using coplanar points," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1993, pp. 626–627.
- [12] G. Schweighofer and A. Pinz, "Robust pose estimation from a planar target," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 12, pp. 2024–2030, 2006.
- [13] T. Collins and A. Bartoli, "Infinitesimal plane-based pose estimation," *International Journal of Computer Vision*, vol. 109, no. 3, pp. 252–286, 2014.
- [14] P.-C. Wu, J.-H. Lai, J.-L. Wu, and S.-Y. Chien, "Stable pose estimation with a motion model in real-time application," in *2012 IEEE International Conference on Multimedia and Expo*. IEEE, 2012, pp. 314–319.
- [15] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5828–5839.
- [16] H. Tanaka, Y. Sumi, and Y. Matsumoto, "A solution to pose ambiguity of visual markers using moire patterns," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 3129–3134.
- [17] H. Tanaka, K. Ogata, and Y. Matsumoto, "Solving pose ambiguity of planar visual marker by wavelike two-tone patterns," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 568–573.
- [18] Y. Uematsu and H. Saito, "Improvement of accuracy for 2d marker-based tracking using particle filter," in *17th International Conference on Artificial Reality and Telexistence (ICAT 2007)*. IEEE, 2007, pp. 183–189.
- [19] P. Jin, P. Matikainen, and S. S. Srinivasa, "Sensor fusion for fiducial tags: Highly robust pose estimation from single frame rgbd," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 5770–5776.
- [20] M. Neunert, M. Bloesch, and J. Buchli, "An open source, fiducial based, visual-inertial motion capture system," in *2016 19th International Conference on Information Fusion (FUSION)*. IEEE, 2016, pp. 1523–1530.
- [21] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
- [22] R. Hartley, J. Trunpf, Y. Dai, and H. Li, "Rotation averaging," *International journal of computer vision*, vol. 103, no. 3, pp. 267–305, 2013.
- [23] A. Chatterjee and V. Madhav Govindu, "Efficient and robust large-scale rotation averaging," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 521–528.
- [24] N. Sünderhauf and P. Protzel, "Towards a robust back-end for pose graph slam," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 1254–1261.
- [25] E. Tomita and T. Seki, "An efficient branch-and-bound algorithm for finding a maximum clique," in *International Conference on Discrete Mathematics and Theoretical Computer Science*. Springer, 2003, pp. 278–289.
- [26] D. Eppstein and D. Strash, "Listing all maximal cliques in large sparse real-world graphs," in *International Symposium on Experimental Algorithms*. Springer, 2011, pp. 364–375.

Chapter 6

Conclusions and Future Work

Robotic perception plays a significant role in enabling intelligent and reliable autonomous machines. This thesis has made significant progress on state estimation problems, particularly problems which involve estimating rotations that reside in the manifold space.

On a practical level, efficient and/or outlier-robust algorithms have been proposed to solve the pose estimation and SLAM/SfM problems, for example, the nonlinear optimization algorithm for outlier-robust INS/GPS fusion, the global rotation averaging algorithm for large-scale SLAM/SfM, and the clique-constrained rotation averaging for marker-based SLAM.

6.1 Future Work

6.1.1 The Outlier-Robust INS/GPS Fusion

The current sensor fusion configuration setup considered in Chapter 3 is INS/GPS fusion. As adding the accelerometer bias to proposed system would introduce unobservable model, which in turn might lead to suboptimal solution and significantly reduce the robustness of the estimation method, the proposed framework does not consider the accelerometer bias. Since the rectification of the accelerometer bias is an important practical problem, it is desirable to have a deeper analysis to investigate the effect of accelerometer bias to the accuracy of the solution.

Another exciting future direction is to extend the optimization framework in Chapter 3 to ultimately fuse a vision sensor with the current sensor setup, which will lead to a new avenue of research. Challenges such as integrating the rotation averaging with the existing framework given different relative rotations, incorporating and estimating the accelerometer bias compensation, and so on would arise.

6.1.2 The Rotation Coordinate Descent Algorithm

In Chapter 4, the rotations were updated every iteration in a sequential manner, such that $k = (1, \dots, n)$ in Algorithm 2 Step 3. Our empirical results suggested that there exists a shorter trajectory to the optimal solution for $k = 1, \dots, n$. Therefore, it is desirable to perform a deeper analysis of the effect of k on the convergence rate of the algorithm and to characterise the optimal sequence of k .

Although an explicit noise bound for strong duality was derived, the corresponding bound was established based on a non-robust cost function. An interesting future work is to derive the bound of a robust cost function, and possibly contribute to a globally optimal rotation averaging algorithm which can tolerate high noise levels.

As the translation estimation is also an important area in visual SLAM/SfM. It is also exciting to investigate whether the proposed method can be extended to incorporate the estimation of translation, to realise a full pose estimation.

6.1.3 The Clique-Constrained Rotation Averaging Algorithm

In contrast to previous work which applied a heuristic criterion to deal with the fundamental ambiguity in the marker-based SLAM/SfM, Chapter 5 proposed a principled way which necessitates an optimisation subroutine. A bottleneck is the underlying optimisation is a batch optimisation, which may become less efficient as the problem size (i.e., the number of markers and the number of views) increases. Therefore, a practical strategy is to extend the current method to a sliding window optimization.

Bibliography

- [1] https://en.wikipedia.org/wiki/Inertial_measurement_unit.
- [2] <http://copter.ardupilot.com/wiki/common-apm-navigation-extended-kalman-filter>.
- [3] <http://www.hitl.washington.edu/artoolkit/>.
- [4] <https://en.wikipedia.org/wiki/Manifold>.
- [5] https://en.wikipedia.org/wiki/Lie_group.
- [6] <https://en.wikipedia.org/wiki/Quaternion>.
- [7] https://en.wikipedia.org/wiki/Maximum_likelihood_estimation.
- [8] https://en.wikipedia.org/wiki/Redescending_M-estimator.
- [9] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. <http://ceres-solver.org>.
- [10] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *IEEE robotics & automation magazine*, 13(3):108–117, 2006.
- [11] Axel Barrau and Silvère Bonnabel. The Invariant Extended Kalman Filter as a stable observer. *IEEE Trans. on Automatic Control*, 62(4):1797–1812, 2017.
- [12] Jean-Daniel Boissonnat. Shape reconstruction from planar cross sections. *Computer vision, graphics, and image processing*, 44(1):1–29, 1988.
- [13] Silvere Bonnabel, Philippe Martin, and Pierre Rouchon. A non-linear symmetry-preserving observer for velocity-aided inertial navigation. In *American Control Conf.*, pages 2910–2914, 2006.
- [14] Silvère Bonnabel, Philippe Martin, and Erwan Salaün. Invariant extended kalman filter: theory and application to a velocity-aided attitude estimation problem. In *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 1297–1304. IEEE, 2009.

-
- [15] Nicolas Boumal. A riemannian low-rank method for optimization over semidefinite matrices with block-diagonal constraints. *arXiv preprint arXiv:1506.00575*, 2015.
- [16] Nicolas Boumal, Vlad Voroninski, and Afonso Bandeira. The non-convex burer-monteiro approach works on smooth semidefinite programs. In *Advances in Neural Information Processing Systems*, pages 2757–2765, 2016.
- [17] Guillaume Bourmaud, Rémi Mégret, Marc Arnaudon, and Audrey Giremus. Continuous-discrete extended kalman filter on matrix lie groups using concentrated gaussian distributions. *Journal of Mathematical Imaging and Vision*, 51(1):209–228, 2015.
- [18] Guillaume Bourmaud, Rémi Mégret, Audrey Giremus, and Yannick Berthoumieu. Discrete extended kalman filter on lie groups. In *21st European Signal Processing Conference (EUSIPCO 2013)*, pages 1–5. IEEE, 2013.
- [19] Guillaume Bourmaud, Rémi Mégret, Audrey Giremus, and Yannick Berthoumieu. From intrinsic optimization to iterated extended kalman filtering on lie groups. *Journal of Mathematical Imaging and Vision*, 55(3):284–303, 2016.
- [20] G. Bradski. The OpenCV Library. *Dr. Dobb’s Journal of Software Tools*, 2000.
- [21] Torleiv H Bryne, Jakob M Hansen, Robert H Rogne, Nadezda Sokolova, Thor I Fossen, and Tor A Johansen. Nonlinear Observers for Integrated INS/GNSS Navigation: Implementation Aspects. *IEEE Control Systems*, 37(3):59–86, 2017.
- [22] Álvaro Parra Bustos, Tat-Jun Chin, Anders Eriksson, and Ian Reid. Visual slam: Why bundle adjust? In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2385–2391. IEEE, 2019.
- [23] Luca Carlone and Frank Dellaert. Duality-based verification techniques for 2d slam. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 4589–4596. IEEE, 2015.
- [24] Luca Carlone, David M Rosen, Giuseppe Calafiore, John J Leonard, and Frank Dellaert. Lagrangian duality in 3d slam: Verification techniques and optimal solutions. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 125–132. IEEE, 2015.
- [25] Luca Carlone, Roberto Tron, Kostas Daniilidis, and Frank Dellaert. Initialization techniques for 3d slam: a survey on rotation estimation and its use in

- pose graph optimization. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 4597–4604. IEEE, 2015.
- [26] SC Chan, ZG Zhang, and KW Tse. A new robust kalman filter algorithm under outliers and system uncertainties. In *2005 IEEE International Symposium on Circuits and Systems*, pages 4317–4320. IEEE, 2005.
- [27] Avishek Chatterjee and Venu Madhav Govindu. Robust relative rotation averaging. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):958–972, 2017.
- [28] Avishek Chatterjee and Venu Madhav Govindu. Efficient and robust large-scale rotation averaging. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 521–528, 2013.
- [29] Gregory S Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*, volume 2. Springer Science & Business Media, 2011.
- [30] John L Crassidis, F Landis Markley, and Yang Cheng. Survey of Nonlinear Attitude Estimation Methods. *Journal of guidance, control, and dynamics*, 30(1):12–28, 2007.
- [31] Joseph DeGol, Timothy Bretl, and Derek Hoiem. Improved structure from motion using fiducial marker matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 273–288, 2018.
- [32] Frank Dellaert, David M Rosen, Jing Wu, Robert Mahony, and Luca Carlone. Shonan rotation averaging: Global optimality by surfing $so(p)^n$. In *European Conference on Computer Vision*, pages 292–308. Springer, 2020.
- [33] Anders Eriksson, Carl Olsson, Fredrik Kahl, and Tat-Jun Chin. Rotation averaging and strong duality. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 127–135, 2018.
- [34] Anders Eriksson, Carl Olsson, Fredrik Kahl, and Tat-Jun Chin. Rotation averaging with the chordal distance: Global minimizers and strong duality. *IEEE transactions on pattern analysis and machine intelligence*, 2019.
- [35] Mark Fiala. ARTag, an improved marker system based on artoolkit. *National Research Council Canada, Publication Number: NRC*, 47419:2004, 2004.
- [36] Mark Fiala. ARTag, a fiducial marker system using digital techniques. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 590–596. IEEE, 2005.

- [37] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation. In *Robotics: Science and Systems*, 2015.
- [38] Johan Fredriksson and Carl Olsson. Simultaneous multiple rotation averaging using lagrangian duality. In *Asian Conference on Computer Vision*, pages 245–258. Springer, 2012.
- [39] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.
- [40] Venu Madhav Govindu. Combining two-view constraints for motion estimation. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 2, pages II–II. IEEE, 2001.
- [41] Venu Madhav Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. IEEE, 2004.
- [42] Håvard Fjær Grip, Thor I Fossen, Tor A Johansen, and Ali Saberi. Globally exponentially stable attitude and gyro bias estimation with application to GNSS/INS Integration. *Automatica*, 51:158–166, 2015.
- [43] Brian Hall. *Lie groups, Lie algebras, and representations: an elementary introduction*, volume 222. Springer, 2015.
- [44] Richard Hartley, Khurram Aftab, and Jochen Trumpf. L1 rotation averaging using the weiszfeld algorithm. In *CVPR 2011*, pages 3041–3048. IEEE, 2011.
- [45] Richard Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *International journal of computer vision*, 103(3):267–305, 2013.
- [46] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [47] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992.
- [48] Alireza Khosravian. *State Estimation for Systems on Lie groups with Nonideal Measurements*. PhD thesis, The Australian National University (Australia), 2016.

- [49] Alireza Khosravian, Jochen Trumpp, Robert Mahony, and Tarek Hamel. Velocity aided attitude estimation on $SO(3)$ with sensor delay. In *IEEE Conf. on Decision and Control (CDC)*, pages 114–120, 2014.
- [50] Alireza Khosravian, Jochen Trumpp, Robert Mahony, and Christian Lageman. Observers for invariant systems on Lie groups with biased input measurements and homogeneous outputs. *Automatica*, 55:19–26, 2015.
- [51] Arthur J Krener. The convergence of the extended kalman filter. In *Directions in mathematical systems theory and optimization*, pages 173–182. Springer, 2003.
- [52] Robert Mahony, Tarek Hamel, and J-M Pflimlin. Complementary filter design on the special orthogonal group $SO(3)$. In *Proc. of the IEEE Transactions on Decision and Control, CDC*, 2005.
- [53] Robert Mahony, Tarek Hamel, and Jean-Michel Pflimlin. Nonlinear Complementary Filters on the Special Orthogonal Group. *IEEE Trans. on Automatic Control*, 53(5):1203–1218, 2008.
- [54] Daniel Martinec and Tomas Pajdla. Robust rotation and translation estimation in multiview reconstruction. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.
- [55] C Masreliez. Approximate non-gaussian filtering with linear state and observation relations. *IEEE Transactions on Automatic Control*, 20(1):107–110, 1975.
- [56] Leonard A McGee and Stanley F Schmidt. Discovery of the Kalman filter as a practical tool for aerospace and industry. Technical Report 86847, NASA Ames Research Center, 1985.
- [57] Rafael Munoz-Salinas, Manuel J Marín-Jimenez, and R Medina-Carnicer. SPM-SLAM: Simultaneous localization and mapping with squared planar markers. *Pattern Recognition*, 86:156–171, 2019.
- [58] Rafael Munoz-Salinas, Manuel J Marin-Jimenez, Enrique Yeguas-Bolivar, and Rafael Medina-Carnicer. Mapping and localization from planar markers. *Pattern Recognition*, 73:158–171, 2018.
- [59] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.
- [60] Tomáš Polóni, Boris Rohal-Ilkiv, and Tor Arne Johansen. Moving Horizon Estimation for Integrated Navigation Filtering. *IFAC-PapersOnLine*, 48(23):519–526, 2015.

-
- [61] David M Rosen, Luca Carlone, Afonso S Bandeira, and John J Leonard. Se-sync: A certifiably correct algorithm for synchronization over the special euclidean group. *The International Journal of Robotics Research*, 38(2-3):95–125, 2019.
- [62] Irvin C Schick and Sanjoy K Mitter. Robust recursive estimation in the presence of heavy-tailed observation noise. *The Annals of Statistics*, pages 1045–1080, 1994.
- [63] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.
- [64] Hauke Strasdat, JMM Montiel, and Andrew J Davison. Real-time monocular SLAM: Why filter? In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2010.
- [65] Hauke Strasdat, José MM Montiel, and Andrew J Davison. Visual slam: why filter? *Image and Vision Computing*, 30(2):65–77, 2012.
- [66] Jos F Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1-4):625–653, 1999.
- [67] Jo-Anne Ting, Evangelos Theodorou, and Stefan Schaal. A Kalman filter for robust outlier detection. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2007.
- [68] John Wang and Edwin Olson. AprilTag 2: Efficient and robust fiducial detection. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4193–4198. IEEE, 2016.
- [69] H Durrant Whyte. Simultaneous localisation and mapping (slam): Part i the essential algorithms. *Robotics and Automation Magazine*, 2006.
- [70] Christopher Zach, Manfred Klopschitz, and Marc Pollefeys. Disambiguating visual relations using loop constraints. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1426–1433. IEEE, 2010.