LUND UNIVERSITY

Polynomial Solvers for Geometric Problems - Applications in Computer Vision and Sensor Networks

Kuang, Yubin

2014

Link to publication

Citation for published version (APA):
Kuang, Y. (2014). *Polynomial Solvers for Geometric Problems - Applications in Computer Vision and Sensor Networks*. [Doctoral Thesis (monograph), Mathematics (Faculty of Engineering)].

Total number of authors:
1

General rights
Unless other specific re-use rights are stated the following general rights apply:
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.
• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: https://creativecommons.org/licenses/

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Polynomial Solvers for Geometric Problems

Applications in Computer Vision and Sensor Networks

Yubin Kuang

Lund Institute of Technology
Lund University

Centre for Mathematical Sciences
Mathematics

Mathematics
Centre for Mathematical Sciences
Lund University
Box 118
SE-221 00 Lund
Sweden

http://www.maths.lth.se/

Doctoral Theses in Mathematical Sciences 2014:2
ISSN 1404-0034

ISBN 978-91-7473-995-4
LUTFMA-1048-2014

© Yubin Kuang, 2014

Printed in Sweden by MediaTryck, Lund 2014

Abstract

Given images of a scene taken with a moving camera or recordings of a moving smart phone playing a song by a microphone array, how hard is it to reconstruct the scene structure or the moving trajectory of the phone? In this thesis, we study and solve several fundamental geometric problems in order to provide solutions to these problems. Of particular interests are the *minimal problems*, where only minimal information or measurements are available for the problems to be solvable. In this thesis, we have identified and solved several important minimal problems in computer vision and sensor network calibration.

The key underlying technique for solving such geometric problems is solving systems of polynomial equations. In this thesis, several general techniques are developed. We utilize numerical schemes and explore symmetric structures of polynomial equations to enable fast and stable polynomial solvers.

These enable fast and robust techniques for reconstruction of the scene structures using different measurements. One of the examples is *structure from sound*. By measuring the time-of-arrivals of specific time instances of a song played on a phone, one can reconstruct the trajectory of the phone as well as the positions of the microphones up to precision of centimeters.

Preface

In this thesis, we focus on solving systems of polynomial equations for geometric problems in computer vision and sensor networks. In particular, we are interested in well-defined and solvable configurations for these geometric problems where absolute minimal information are available. These minimal configurations are not only of theoretical importance, but also crucial for robust estimation in the presence of extreme noise - outliers. In many cases, these minimal problems lead to constructing and solving systems of polynomial equations. For a particular geometric problem, there are in general many ways to parameterize the unknowns to form polynomial systems. Choosing a proper parameterization to reduce number of unknowns and degrees of the polynomial equations is an essential step to solve the problem. As for solving a polynomial system, numerical techniques are key to ensure numerical stability of polynomial solvers. In this thesis, we study parameterization for a wide range of geometric problems as well as general techniques to improve numerical stability and efficiency for state-of-the-art polynomial solvers.

Another topic of this thesis is optimization for image recognition. Image recognition relies largely on training the machine with prior information or labelled data. We study a formulation to encode such prior information as a continuous optimization problem to improve the capability of the recognition pipeline.

Main Papers

This thesis is based on the materials in the following papers:

- Y. Kuang, J.E. Solem, F. Kahl and K. Åström, Minimal Solvers for Relative Pose with a Single Unknown Radial Distortion, *accepted to IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, Ohio, 2014
- Y. Kuang, Y. Zheng and K. Åström, Partial Symmetry in Polynomial Systems and Its Application in Computer Vision, *accepted to IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, Ohio, 2014

- Y. Kuang, K. Åström, Pose Estimation with Unknown Focal Length using Points, Directions and Lines, *International Conference on Computer Vision*, Sydney, Australia, 2013
- Y. Kuang, S. Burgess, A. Torstensson and K. Åström, A Complete Characterization and Solution to Microphone Position Self-Calibration Problem, *International Conference on Acoustics, Speech and Signal Processing*, Vancouver, Canada, 2013.
- Y. Kuang and K. Åström, Stratified sensor network self-calibration from TDOA Measurements, *European Signal Processing Conference*, Morocco, 2013.
- Y. Kuang, K. Åström, Single Antenna Anchor-Free UWB Positioning based on Multipath Propagation, *International Conference on Communications*, Budapest, Hungary, 2013
- F Jiang, Y Kuang, K. Åström, Time Delay Estimation for TDOA Self-Calibration using Truncated Nuclear Norm, *International Conference on Acoustics, Speech and Signal Processing*, Vancouver, Canada, 2013
- S. Burgess, Y. Kuang, J. Wendeberg, K. Åström, Minimal Solvers for Unsynchronized TDOA Sensor Network Calibration using Far Field Approximation, *ALGOSENSOR* 2013
- Y. Kuang and K. Åström, Numerically Stable Optimization of Polynomial Solvers for Minimal Problems, *European Conference on Computer Vision*, Florence, Italy, 2012
- E. Ask, Y. Kuang and K. Åström, Exploiting p-Fold Symmetries for Faster Polynomial Equation Solving, *International Conference on Pattern Recognition*, Japan, 2012
- Y. Kuang, E. Ask, S. Bugeress, K. Åström, Understanding TOA and TDOA Network Calibration using Far Field Approximation as Initial Estimate, *in Proc. International Conference on Pattern Recognition Applications and Methods*, Vilamoura, Algarve, Portugal, 2012
- Y. Kuang, M. Byröd and K. Åström, Supervised Feature Quantization with Entropy Optimization, *IEEE Workshop on Information Theory in Computer Vision and Pattern Recognition*, Barcelona, Spain, 2011.
- Y. Kuang, K. Åström, L. Kopp, M. Oskarsson and M. Byröd, Optimizing Visual Vocabularies Using Soft Assignment Entropies, *Asian Conference in Computer Vision*, Queens Town, New Zealand, 2010.

Subsidiary Papers

- Z. Shimayijiang, S. Burgess, Y. Kuang and K. Åström, Minimal Solutions for Dual Microphone Rig Self-calibration, *submitted to European Signal Processing Conference*, 2014

- E. Ask, Y. Kuang and K. Åström, A Unifying Approach to Minimal Problems in Collinear and Planar TDOA Sensor Network Self-Calibration, *submitted to European Signal Processing Conference*, 2014

- Z. Shimayijiang, F. Andersson, Y. Kuang, and K. Åström, An Automatic System for Microphone Self-Localization Using Ambient Sound, *submitted to European Signal Processing Conference*, 2014

- Y. Zheng, Y. Kuang, S. Sugimoto, K. Åström, M. Okutomi, Revisiting the PnP Problem: A Fast, General and Optimal Solution, *International Conference on Computer Vision*, Sydney, Australia, 2013

- S. Burgess, Y. Kuang and K. Åström, TOA Sensor Network Calibration for Receiver and Transmitter Spaces with Difference in Dimension, *European Signal Processing Conference*, Morocco, 2013

- S. Burgess, Y. Kuang and K. Åström, Pose Estimation from Minimal Dual-Receiver Configuration, *International Conference on Pattern Recognition*, Japan, 2012

- S. Burgess, Y. Kuang and K. Åström, Node Localization in Unsynchronized Time of Arrival Sensor Networks, *International Conference on Pattern Recognition*, Japan, 2012

- T. Schoenemann; Y. Kuang, F. Kahl, Curvature Regularity for Multi-Label Problems - Standard and Customized Linear Programming, *in Proc. Energy Minimization Methods in Computer Vision and Pattern Recognition*, Saint Petersburg, Russia, 2011.

- Y. Kuang, D. Stork, F. Kahl, Improved Curvature-based Inpainting applied to Fine Art - Recovering van Goghs Partially Hidden Brush Strokes, *in Proc. IS&T/SPIE Electronic Imaging, Computer image analysis in the study of art II*, San Francisco, California, 2010.

Acknowledgements

First of all, I would like to thank my main supervisor, Kalle Åström for his patience, support and enthusiasm. Kalle's guidance has inspired and motivated me in many ways. I also thank my co-supervisors Fredrik Kahl for introducing me to discrete optimization, Jan Erik Solem for sharing his vision in both academia and industry, Magnus Oskarsson for many discussions in geometric vision.

I am also extremely fortunate to have the privilege to work with Erik Ask, Simon Burgess, Martin Byröd, Sebastian Haner, Fangyuan Jiang, Carl Olsson, Zhayida Simayijiang, Petter Strandmark, Linus Svärm, and Johannes Ulén at the Mathematical Imaging Group. A thank you also goes to Yinqiang Zheng at National Institute of Informatics in Japan for inspiring discussions that lead to two papers. I would also like to thank Thomas Schonemann, Anna Torstensson, Fredrik Tufvesson, Johannes Wenderberg and David Stork for collaborations.

Furthermore, I would like to thank our secretary Ann-Kristin Ottosson and later Lena Furberg for the help in practical issues. I am grateful for my parents, my grandma and Xuan for being always supportive as well as all my friends for the supports of everything else.

The works of this thesis have been supported by the strategic research projects ELLIIT and eSSENCE, Swedish Foundation for Strategic Research projects EN-GROSS and VINST (no. RIT08-0043), and European research council's GlobalVision grant (no. 209480).

Lund, April, 2014
Yubin Kuang

Contents

1 Introduction **1**
1.1 Overview 1
1.2 Contributions 3

2 Preliminaries **9**
2.1 Geometry in Computer Vision 9
2.1.1 Camera Model 9
2.1.2 Epipolar Geometry 10
2.1.3 Radial Distortion 10
2.2 Geometry in Bipartite Sensor Networks 11
2.2.1 Far-Field Approximation 13
2.3 Minimal Cases 14
2.4 Algebraic Geometry 14
2.4.1 The Action Matrix 15
2.4.2 Constructing Action Matrix 15

I Solving Systems of Polynomials Equations **19**

3 Optimizing Polynomial Solvers **21**
3.1 Motivation 21
3.2 Optimizing Size and Accuracy for Solvers 22
3.2.1 Permissible Selection 23
3.2.2 Equation Removal 24
3.2.3 Optimization Scheme 25
3.3 Experimental Validation 25
3.3.1 Nine-Point Uncalibrated Radial Distortion 26
3.3.2 Six-Point Calibrated Radial Distortion 29
3.3.3 Three-Point Stitching 29
3.4 Conclusions 31

4 Symmetries in Polynomial Systems **33**
4.1 Full Symmetry . . . 33
4.2 Partial Symmetry . . . 34
4.3 Utilizing Symmetry . . . 36
4.3.1 Zero Solutions . . . 37
4.3.2 Symmetric Action Matrix . . . 37
4.3.3 Elimination Template with Symmetry . . . 38
4.3.4 Extracting Solutions . . . 38
4.3.5 Detecting Symmetry . . . 40
4.4 Applications . . . 40
4.4.1 Optimal Perspective-n-Point . . . 41
4.4.2 Optimal Euclidean Registration . . . 42
4.4.3 PnL Problem . . . 43
4.4.4 Symmetric Systems and Solvers . . . 43
4.5 Experiments . . . 47
4.6 Conclusions . . . 50

II Geometric Problems in Computer Vision **53**

5 Pose Estimation **55**
5.1 Pose Estimation with Unknown Focal . . . 55
5.2 Problem Formulation . . . 56
5.2.1 Number of Constraints . . . 57
5.2.2 Useful Cases . . . 59
5.2.3 Parameterization . . . 59
5.3 Polynomial Solvers . . . 61
5.4 Experiments . . . 61
5.4.1 Synthetic Data . . . 61
5.4.2 Real Data . . . 65
5.5 Discussions . . . 68
5.6 Conclusions . . . 69

6 Radial Distortion **71**
6.1 Single Unknown Radial . . . 71
6.2 Problem Formulation . . . 73
6.2.1 Fundamental Matrix and Radial Distortion . . . 74
6.2.2 Essential Matrix and Radial Distortion . . . 74
6.3 Polynomial Solvers . . . 75
6.3.1 8 Point Case: $\mathbf{F} + \lambda$ . . . 75
6.3.2 7 Point Case: $\mathbf{E} + \lambda + f$ . . . 76
6.3.3 6 Point Case: $\mathbf{E} + \lambda$ . . . 78

6.3.4 Alternative Parameterization 78
6.3.5 Degenerated Cases 79
6.4 Experiments 79
6.4.1 Synthetic Experiments 79
6.4.2 Real Experiments 83
6.5 Conclusions 85

III Geometric Problems in Sensor Networks 87

7 TOA Self-Calibration 89
7.1 Background 89
7.2 The TOA Self-Calibration Problem 90
7.2.1 Minimal Cases 91
7.2.2 Solving the Polynomial Systems 93
7.3 Lower and Higher Dimensions 96
7.4 Difference in Dimension 97
7.4.1 Minimal Problems 98
7.4.2 Discussion 98
7.5 Failure Modes 98
7.6 Overdetermined Cases 99
7.6.1 General Overdetermined Solvers 99
7.6.2 Specialized Overdetermined Solvers 100
7.6.3 Nonlinear Optimization 101
7.7 Experiments 101
7.7.1 Synthetic Experiments 103
7.7.2 Real Experiments 105
7.8 Conclusions 106

8 TDOA Self-Calibration 107
8.1 Background 107
8.2 Minimal Cases for Unknown Offsets 108
8.2.1 Rank-$(k+1)$ Constraint 108
8.2.2 Rank-k Constraint 109
8.2.3 Discussions 111
8.3 Rank Optimization 111
8.3.1 Optimization Scheme 112
8.4 Solving TDOA Self-Calibration 115
8.5 Experiments 115
8.5.1 Synthetic Data 115
8.5.2 Real Data 120
8.6 Conclusions 121

9 Far-Field TDOA **123**
9.1 Far-Field TDOA Self-Calibration 123
9.1.1 Failure Modes of the Algorithm 126
9.1.2 Overdetermined Cases 128
9.2 Synthetic Experiments 129
9.2.1 Minimal Solver Accuracy 130
9.2.2 Far-Field Approximation Accuracy 130
9.2.3 Overdetermined Cases 131
9.3 Real Experiment 132
9.3.1 Signal model 133
9.3.2 Finding Correspondences among MPCs 133
9.3.3 Measurement Setup 133
9.3.4 Data Processing 134
9.3.5 Position Estimates 135
9.3.6 Discussions 138

10 Far-Field UTOA **139**
10.1 Far-Field UTOA Self-Calibration 139
10.2 A Matrix Factorization Method 140
10.3 Experimental Validation 143
10.3.1 Synthetic Experiments 143
10.3.2 Real Data 145
10.4 Conclusions 146

IV Image Retrieval **147**

11 Entropy Optimization **149**
11.1 Introduction 149
11.2 Vocabulary Optimization 151
11.2.1 Formulation 151
11.2.2 Efficient Gradient Computation 153
11.2.3 Approximate Computation 155
11.2.4 Connection to Max-Margin Clustering 155
11.3 Ground-Truth Dataset 156
11.4 Experiments 156
11.4.1 Parameter Sensitivity 157
11.4.2 Generalization 158
11.4.3 Optimization over Subspace 159
11.4.4 Image Retrieval 160
11.5 Conclusions 161

Chapter 1
Introduction

Geometric problems in computer vision and sensor networks in general involve the inverse problems of recovering certain geometric property e.g. positions and rotations of the cameras, positions of sensors, given specific measurements. For computer vision, such geometric problems can be to estimate the scene structure and the camera motions based on image measurements. For sensor network calibration, one of the important geometric inverse problems is to estimate the positions of the sensors based on time-of-arrival or time-difference-of-arrival measurements.

The main focus of this thesis is to solve such inverse problems with minimally available measurements such that the problems are solvable. In many cases, these minimal problems can be formulated as systems of polynomial equations. The solvability of these problems is then closely related to the algebraic geometry of the corresponding polynomial systems. For a solvable problem, an effective and stable polynomial solver is essential for practical application. This thesis devotes efforts both to identifying unsolved minimal problems in computer vision and sensor networks and to deriving general techniques to improve numerical stability and efficiency of polynomial solvers.

1.1 Overview

This section gives an overview of the thesis and describes the connections between different papers.

Part I: Solving Systems of Polynomial Equations

Chapter 3 This chapter presents a scheme for optimizing the numerical stability and efficiency of polynomial solvers which is mainly based on the work in [54]. The optimization scheme consists two steps (i) equation removal (ii) permissible selection. Our results show that with simple greedy schemes, one can improve the polynomial solvers for a wide range of geometric problems in computer vision, thus outperforming state-of-the-art polynomial solvers in terms of both speed and stability.

Chapter 4 The symmetric structures in the monomials are explored to enable faster and more stable polynomial solvers. Full symmetry in the polynomial systems was first exploited in [3] and was later generalized to partial symmetry in [63]. The importance and advantages of utilizing symmetry are demonstrated for geometric problems in computer vision. One of the examples with full symmetry was also discussed for the optimal pose estimation problem [104].

Part II: Geometric Problems in Computer Vision

Chapter 5 This chapter focuses on camera pose estimation problem in computer vision. Specifically, we study and identify a new set of solvable minimal problems of pose estimation with unknown focal length with line, points and direction features. The results are published in [56].

Chapter 6 We study the problem of estimating relative pose with a single unknown radial distortion on one of the two cameras. Three previously unsolved minimal problems are identified. We study ways of parameterization and develop efficient polynomial solvers for these minimal problems. The materials of this chapter are based on [62].

Part III: Geometric Problems in Sensor Networks

Chapter 7 The minimal problems of bipartite sensor network calibration using time-of-arrival (TOA) measurements are studied. We identify the minimal problems for general 2D and 3D configurations. We study a new parameterization based on rank-constraints of the measurement matrix and present a saturation scheme that is essential to solve the underlying polynomial system [60]. For cases where the dimension of the receivers and transmitters are different, we derive a linear solver based on factorization for reconstructing the sensor positions [16].

Chapter 8 In this chapter, we study the problems of recovering unknown time offsets in sensor network calibration with time-difference-of-arrival (TDOA) measurements. The key technique is to utilize the rank constraints of the measurement matrix and to formulate the unknown offsets in terms of the determinant constraints. We study thoroughly the minimal measurements needed to solve for the unknown offsets [57]. For overdetermined cases, we present a formulation that aims to minimize the rank of the compacted measurement matrix [48].

Chapter 9 We study sensor network self-calibration in far-field setup where the distances between the receivers are much smaller than their distances to any of the

transmitters. We show that in the far-field setup, TOA self-calibration and TDOA self-calibration are equivalent. In this setup, one can reconstruct the receivers positions and the directions from the transmitters to the receivers. By using the rank constraints, the minimal case of this problem can be solved linearly [53]. We also test the linear solver on UWB signals where multipath propagation is utilized to synthesize the existence of multiple transmitters [59].

Chapter 10 In this chapter, we extend the analysis of far-field setup to unsynchronized receivers and transmitters i.e. unsynchronized time-of-arrival (UTOA) measurements. We derive a minimal solver based on linear factorization that is general for 2D and 3D problems [17].

Part IV: Optimization in Image Retrieval

Chapter 11 Given matching image patches, we study the effects of vocabulary learning for local features in image retrieval. We integrate an entropy optimization into the vector quantization step and derive efficient training schemes [58, 61].

1.2 Contributions

All of the published papers or manuscripts during my study are collaborative works from all the authors. I describe in this section the contributions of the author and coauthors for all these works.

Main Papers

- Y. Kuang, J.E. Solem, F. Kahl and K. Åström, Minimal Solvers for Relative Pose with a Single Unknown Radial Distortion, *accepted to IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, Ohio, 2014

  In the process of solving two-sided unknown radial distortion, I came up with the problems of radial distortion calibration with a single unknown radial distortion. I was responsible for all the algorithms, codes, experiments as well as the writing. Jan Erik Solem, Kalle Åström and Fredrik Kahl worked on real data collection and provided helpful discussions on several issues of the paper.

- Y. Kuang, Y. Zheng and K. Åström, Partial Symmetry in Polynomial Systems and Its Application in Computer Vision, *accepted to IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, Ohio, 2014

  I developed the algorithms and wrote the solvers for partial symmetric problems. Yinqiang Zheng first brought up the partially symmetric examples in

computer vision and worked on the experiments. Kalle Åström and I derived the theoretical part of partial symmetry. I have written most of the paper and run half of the experiments.

- Y. Kuang, K. Åström, Pose Estimation with Unknown Focal Length using Points, Directions and Lines, *International Conference on Computer Vision*, Sydney, Australia, 2013

  I and Kalle Åström had the idea to extend the minimal solvers with richer features. I developed algorithms for all the solvers and most of the experiments. Kalle Åström worked on generating reconstruction from real data.

- Y. Kuang, S. Burgess, A. Torstensson and K. Åström, A Complete Characterization and Solution to Microphone Position Self-Calibration Problem, *International Conference on Acoustics, Speech and Signal Processing*, Vancouver, Canada, 2013.

  Kalle Åström, I and Anna Torstensson developed the algorithms together. I implemented and optimized the solvers and did all the synthetic experiments. I wrote most of the paper and Simon Burgess worked on real experiments.

- Y. Kuang and K. Åström, Stratified sensor network self-calibration from TDOA Measurements, *European Signal Processing Conference*, Morocco, 2013.

  I first developed the linear solver. Kalle Åström came up with the concept of rank constraints and I developed all the minimal solvers based on rank constraints. I did all the synthetic experiments. Kalle Åström and I worked together on the writing as well as real experiments.

- Y. Kuang, K. Åström, Single Antenna Anchor-Free UWB Positioning based on Multipath Propagation, *International Conference on Communications*, Budapest, Hungary, 2013

  I worked on the algorithmic part and RANSAC experiments for the paper. Kalle Åström processed the data and did the reconstruction. Fredrik Tufvesson collected the data and wrote the signal processing part of the paper.

- F Jiang, Y Kuang, K. Åström, Time Delay Estimation for TDOA Self-Calibration using Truncated Nuclear Norm, *International Conference on Acoustics, Speech and Signal Processing*, Vancouver, Canada, 2013

  I developed the algorithm with truncated nuclear norm and ran most of the experiments based on Kalle Åström's suggestions of utilizing nuclear norm. Fangyuan Jiang was responsible for both the experiments and writing.

- S. Burgess, Y. Kuang, J. Wendeberg, K. Åström, Minimal Solvers for Unsynchronized TDOA Sensor Network Calibration using Far Field Approximation, *ALGOSENSOR* 2013

I came up with the idea of a linear solver for the minimal problem and developed the algorithm together with Simon Burgess. I wrote most of codes and the section for the linear solver. Simon Burgess and Johannes Wendeberg did most of analysis, writing and experiments.

- Y. Kuang and K. Åström, Numerically Stable Optimization of Polynomial Solvers for Minimal Problems, *European Conference on Computer Vision*, Florence, Italy, 2012

  Kalle Åström first had the idea of optimizing polynomial solver. We developed and implemented the algorithms and theories together. I wrote most of the paper and conducted most of the experiments.

- E. Ask, Y. Kuang and K. Åström, Exploiting p-Fold Symmetries for Faster Polynomial Equation Solving, *International Conference in Pattern Recognition*, Japan, 2012

  Erik Ask and Kalle Åström developed the main theory and methods. I was involved in developing the method at a later stage. I worked on optimizing the solvers and was responsible for one of synthetic experiments and optimization of the solvers.

- Y. Kuang, E. Ask, S. Bugeress, K. Åström, Understanding TOA and TDOA Network Calibration using Far Field Approximation as Initial Estimate, *in Proc. International Conference on Pattern Recognition Applications and Methods*, Vilamoura, Algarve, Portugal, 2012

  Kalle Åström first came to the linear solution. I worked on the alternating optimization and other experiments. Erik Ask, and Simon Burgess were also responsible for the experiments.

- Y. Kuang, M. Byröd and K. Åström, Supervised Feature Quantization with Entropy Optimization, *IEEE Workshop on Information Theory in Computer Vision and Pattern Recognition*, Barcelona, Spain, 2011.

  I implemented the algorithms and did all the experiments and writing. Martin Byröd came up with the speed-up scheme.

- Y. Kuang, K. Åström, L. Kopp, M. Oskarsson and M. Byröd, Optimizing Visual Vocabularies Using Soft Assignment Entropies, *Asian Conference in Computer Vision*, Queens Town, New Zealand, 2010.

  Kalle Åström and I developed the formulation and implemented the algorithm together. I was responsible for most of experiments. Other coauthors contributed in comments and discussions.

Subsidiary Papers

- Z. Shimayijiang, S. Burgess, Y. Kuang and K. Åström, Minimal Solutions for Dual Microphone Rig Self-calibration, *submitted to European Signal Processing Conference*, 2014

  I came up with the formulation of dual-microphone sensors and developed two of the minimal solvers myself. Together with Zhayida Shimayijiang, Simon Burgess and Kalle Åström, we developed the rest of the solvers and experiments. I did part of the writing.

- E. Ask, Y. Kuang and K. Åström, A Unifying Approach to Minimal Problems in Collinear and Planar TDOA Sensor Network Self-Calibration, *submitted to European Signal Processing Conference*, 2014

  I developed and implemented several minimal solvers in the paper. I wrote parts of the algorithmic section of the paper. Erik Ask wrote most of the paper and did most of experiments together with Kalle Åström.

- Z. Shimayijiang, F. Andersson, Y. Kuang, and K. Åström, An Automatic System for Microphone Self-Localization Using Ambient Sound, *submitted to European Signal Processing Conference*, 2014

  I worked on revising the paper. Zhayida Shimayijiang, Fredrik Andersson and Kalle Åström and did most of the work.

- Y. Zheng, Y. Kuang, S. Sugimoto, K. Åström, M. Okutomi, Revisiting the PnP Problem: A Fast, General and Optimal Solution, *International Conference on Computer Vision*, Sydney, Australia, 2013

  I developed and optimized the customized symmetric solver for the problem and wrote about details on generating the solver. Yinqiang Zheng is the main contributor of this paper.

- S. Burgess, Y. Kuang and K. Åström, TOA Sensor Network Calibration for Receiver and Transmitter Spaces with Difference in Dimension, *European Signal Processing Conference*, Morocco, 2013

  I was responsible for experiments of the paper and the later the algorithm and codes of double-compacted solver for the problem.

- S. Burgess, Y. Kuang and K. Åström, Pose Estimation from Minimal Dual-Receiver Configuration, *International Conference on Pattern Recognition*, Japan, 2012

  I worked on optimizing and speeding up the polynomial solvers for different problems. Simon Burgess did most of the writing and experiments.

- S. Burgess, Y. Kuang and K. Åström, Node Localization in Unsynchronized Time of Arrival Sensor Networks, *International Conference on Pattern Recognition*, Japan, 2012

  Simon did most of the writing and experiments as well as theories together with Kalle Åström. I was responsible for part of the writing and the algorithm development.

- T. Schoenemann; Y. Kuang, F. Kahl, Curvature Regularity for Multi-Label Problems - Standard and Customized Linear Programming, *in Proc. Energy Minimization Methods in Computer Vision and Pattern Recognition*, Saint Petersburg, Russia, 2011.

  I worked on the multi-label formulation, algorithm implementation and writing for the part of inpainting. Thomas Schoenemann did the other works together with Fredrik Kahl.

- Y. Kuang, D. Stork, F. Kahl, Improved Curvature-based Inpainting applied to Fine Art - Recovering van Goghs Partially Hidden Brush Strokes , *in Proc. IS&T/SPIE Electronic Imaging, Computer image analysis in the study of art II*, San Francisco, California, 2010.

  David Stork started the discussions with Fredrik Kahl on recovering hidden strokes. I developed the pipeline and codes for the processing and wrote most of paper.

Chapter 2
Preliminaries

To attack any problems in the real world in a mathematical way, a proper abstraction or modeling of the real world is essential. For geometric problems in computer vision and sensor networks, the key is to build up mathematical models between sensors (e.g. cameras, microphones) and the corresponding measurements. In this chapter, we introduce several geometric concepts and models in computer vision and sensor network calibration. Then we review the key concepts in algebraic geometry and the action matrix method for solving systems of polynomial equations.

2.1 Geometry in Computer Vision

2.1.1 Camera Model

We consider the standard *pinhole camera* model widely used in computer vision [43]. In the pinhole camera model, it is assumed that all light-rays towards the camera pass through a common point $\mathbf{C}$ in space. Here $\mathbf{C}$ is also called the *camera center*. An image is formed by the intersection of these light-rays with the image plane Π. For a 3D point $\mathbf{X} = \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T$ and its corresponding 2D image projections $\mathbf{x} = \begin{bmatrix} u & v & 1 \end{bmatrix}^T$ in homogeneous coordinates, the projection equation is,

$$\eta \mathbf{x} = \mathbf{P}\mathbf{X}. \tag{2.1}$$

Here, the scalar η is the *depth* of the 3D point $\mathbf{X}$ and $\mathbf{P}$ is a 3×4 *camera matrix* which can be factorized as,

$$\mathbf{P} = \mathbf{K}\left[\mathbf{R} \mid \mathbf{t}\right]. \tag{2.2}$$

The rotation matrix $\mathbf{R}$ encodes the orientational part of the camera pose specifying in which direction the camera is pointing and $\mathbf{t}$ relates to the coordinate of the camera center $\mathbf{t} = -\mathbf{R}\mathbf{C}$. Here, $\mathbf{K}$ is the calibration matrix that compensates for the intrinsic

setup of the camera as follows:

$$\mathbf{K} = \begin{bmatrix} f & \alpha & u \\ 0 & \gamma f & v \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.3}$$

Here f is the *focal length* that indicates the distance of the camera center to the image plane. In most applications, it is assumed that the image coordinate system is perpendicular such that $\alpha = 0$ (no *skew*), the pixels are square i.e. *aspect ratio* $\gamma = 1$ and the position of the *principle point* (u, v) is at the center of the image. A camera if said to be *calibrated* is the calibration matrix $\mathbf{K}$ is completely known. Otherwise, the camera is *uncalibrated*.

2.1.2 Epipolar Geometry

In computer vision, epipolar geometry relates to geometric relations of the image points of a 3D scene between two camera views. Given two image projections $\mathbf{x} = \begin{bmatrix} u & v & 1 \end{bmatrix}^T$ and $\mathbf{x}' = \begin{bmatrix} u' & v' & 1 \end{bmatrix}^T$ of the same 3D world point from two cameras $\mathbf{P}$ and $\mathbf{P}'$, there exist bilinear constraints between the corresponding points $\mathbf{x}$ and $\mathbf{x}'$. Specifically, if both cameras are uncalibrated, then the *epipolar constraint* is

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0, \tag{2.4}$$

where $\mathbf{F} \in \mathbb{R}^{3\times 3}$ is the *fundamental matrix*. A fundamental matrix $\mathbf{F}$ is only defined up to scale and it satisfies $\det(\mathbf{F}) = 0$.

As for calibrated cameras, the bilinear constraint is determined by an *essential matrix* $\mathbf{E} \in \mathbb{R}^{3\times 3}$. The corresponding points $\mathbf{x}$ and $\mathbf{x}'$ satisfy

$$\mathbf{x}'^T \mathbf{E} \mathbf{x} = 0. \tag{2.5}$$

It is shown in [43] that a 3×3 matrix $\mathbf{E}$ is an essential matrix if and only if $\det(\mathbf{E}) = 0$ and it satisfies the trace constraints

$$2\mathbf{E}\mathbf{E}^T\mathbf{E} - \mathrm{tr}(\mathbf{E}\mathbf{E}^T)\mathbf{E} = 0,$$

which indicate that the two non-zero singular values of $\mathbf{E}$ are equal.

2.1.3 Radial Distortion

With the presence of radial distortion from camera lens, the linear pinhole camera model in (2.1) is not valid any more for image projections. To model the nonlinear effects of radial distortion, the undistorted image projection $\mathbf{x}_u = \begin{bmatrix} x_u & y_u & 1 \end{bmatrix}^T$ and distorted image projection $\mathbf{x}_d = \begin{bmatrix} x_d & y_d & 1 \end{bmatrix}^T$ are related as

$$\mathbf{x}_d = \mathcal{L}\big(r(\mathbf{x}_u)\big)\mathbf{x}_u. \tag{2.6}$$

Here $\mathcal{L}(r) = 1 + k_1 r + k_2 r^2 + k_3 r^3 + k_4 r^4 + \dots$ which can be seen as a Taylor expansion of an arbitrary smooth function and $r(\mathbf{x_u})$ is the distance of the undistorted image points to the principle point. In this thesis, we assume a one-parameter division model for radial distortion as in [36] which is an effective approximation of the general smooth function. Under the one-parameter radial distortion model, the relation between undistorted point coordinates $\mathbf{x}_u$ and radially distorted point coordinates $\mathbf{x}_d$ can be written as

$$\begin{bmatrix} x_u \\ y_u \\ 1 \end{bmatrix} \sim \begin{bmatrix} x_d \\ y_d \\ 1 + \lambda r_d^2 \end{bmatrix}, \tag{2.7}$$

where λ is the distortion coefficient and r_d is the distance of $\mathbf{x}_d$ to the distortion center.

2.2 Geometry in Bipartite Sensor Networks

We consider a particular type of sensor networks where the graph structure is *bipartite*. A graph is bipartite if the vertices of the graph can be divided into two independent sets (such that there are no edges between vertices within each set). In sensor network, these two independent sets can be regards as *receivers* and *transmitters*. Let $\mathbf{r}_i$, $i = 1, \dots, m$ and $\mathbf{s}_j$, $j = 1, \dots, n$ be the spatial coordinates of m receivers and n transmitters, respectively. In this thesis, we consider the problem of reconstructing the spatial positions of the receivers $\{\mathbf{r}_i\}$ and the transmitters $\{\mathbf{s}_j\}$ by measuring the arrival time instances $\{t_{ij}\}$ of signals for the receivers. Here we assume that the speed of the signal v is known and constant. We further assume that each receiver $\mathbf{r}_i$ has its own clock offset τ_i (time of transmission) and the same applies to each transmitter $\mathbf{s}_j$ with clock offset μ_j (time of recording). Then we have the following model between the positions and the measurements

$$\begin{aligned} \|\mathbf{r}_i - \mathbf{s}_j\|_2 &= v(t_{ij} + \tau_i - \mu_j) \\ &= f_{ij} + q_i + o_j, \end{aligned} \tag{2.8}$$

where $f_{ij} = vt_{ij}$, $q_i = v\tau_i$ and $o_j = -v\mu_j$. We call this *unsynchronized time-of-arrive* (UTOA) measurements. If the receivers are synchronized i.e. $\tau = \tau_1 = \tau_2 = \dots$, we have

$$\|\mathbf{r}_i - \mathbf{s}_j\|_2 = f_{ij} + o_j, \tag{2.9}$$

where $o_j = v(\tau - \mu_j)$. Then we know the relative arrival time or *time difference of arrival* (TDOA) of the signals from different transmitters to receivers. If we further assume that the transmitters and receivers are synchronized under the same clock, i.e.

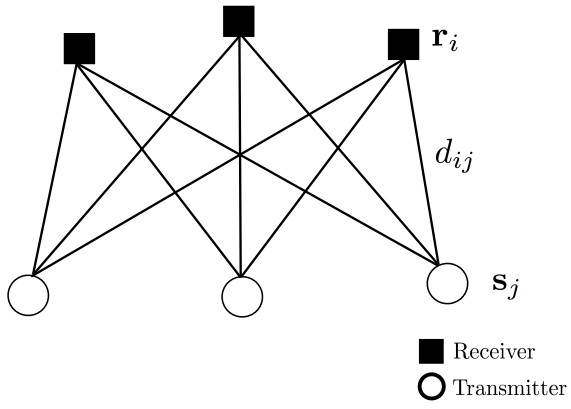


Figure 2.1: A configuration of 3 transmitters (circles) and 3 receivers (squares) with known pairwise distances d_{ij} in 2D.

$\tau_1 = \tau_2 = \cdots = \mu_1 = \mu_2 = \ldots$, then based on (2.8), we have *time of arrival* (TOA) measurements

$$\|\mathbf{r}_i - \mathbf{s}_j\|_2 = d_{ij}, \tag{2.10}$$

where $d_{ij} = vt_{ij}$ is the absolute distance between the receiver i and transmitter j (Figure 2.1). We have the following calibration problems to consider:

Problem 2.2.1 (UTOA Bipartite Network Calibration) Given relative distance measurements f_{ij} determine receiver positions $\mathbf{r}_i$, $i = 1, \ldots, m$, transmitter positions $\mathbf{s}_j$, $j = 1, \ldots, n$, offsets $q_i, i = 1, \ldots, m$ and $o_j, j = 1, \ldots, n$ such that $f_{ij} = \|\mathbf{r}_i - \mathbf{s}_j\|_2 + q_i + o_j$.

Problem 2.2.2 (TDOA Bipartite Network Calibration) Given relative distance measurements f_{ij} determine receiver positions $\mathbf{r}_i$, $i = 1, \ldots, m$, transmitter positions $\mathbf{s}_j$, $j = 1, \ldots, n$ and offsets $o_j, j = 1, \ldots, n$ such that $f_{ij} = \|\mathbf{r}_i - \mathbf{s}_j\|_2 + o_j$.

Problem 2.2.3 (TOA Bipartite Network Calibration) Given absolute distance measurements d_{ij} determine receiver positions $\mathbf{r}_i$, $i = 1, \ldots, m$ and transmitter positions $\mathbf{s}_j$, $j = 1, \ldots, n$ such that $d_{ij} = \|\mathbf{r}_i - \mathbf{s}_j\|_2$.

For all these problems, one can only reconstruct positions of receivers and transmitters up to an Euclidean transformation and mirroring.

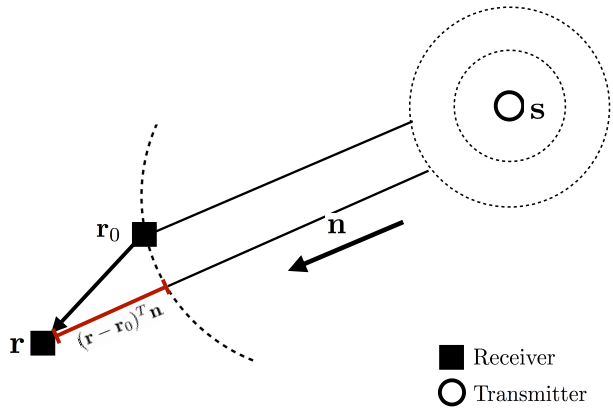


Figure 2.2: Approximation of distance between receivers and transmitters in far-field configuration.

2.2.1 Far-Field Approximation

In practice, there are spatial configurations for the sensors such that distances between receivers are small and transmitters are all very far away from the receivers. In this case, a good approximation is to assume that the directions from transmitter to each of the receivers are parallel and independent of the positions of the receivers. If we denote these directions as $\mathbf{n}$, then we have the following far-field approximation of the distance measurement between $\mathbf{r}$ and $\mathbf{s}$

$$\begin{aligned} d &= \|\mathbf{r}-\mathbf{s}\|_2 \\ &\approx \|\mathbf{r}_0-\mathbf{s}\|_2 + (\mathbf{r}-\mathbf{r}_0)^T\mathbf{n}_j \\ &= \mathbf{r}^T\mathbf{n} + \underbrace{(\|\mathbf{r}_0-\mathbf{s}\|_2 - \mathbf{r}_0^T\mathbf{n})}_{c}, \end{aligned} \tag{2.11}$$

where $\mathbf{r}_0$ is a reference receiver position (Figure 2.2). Here, we can see that the distance d can be approximately expressed as inner product of the receivers position $\mathbf{r}$ and its direction form transmitter $\mathbf{n}$ and a constant c that is related to the position of transmitter. Using this approximation, the inverse problems of interests are to reconstruct the receiver positions and the directions of transmitters to the receivers given TOA, TDOA or UDOA measurements.

2.3 Minimal Cases

We define a *minimal case* to a problem as follows:

Definition 2.3.1. *A minimal case to a problem is the case that consists of the minimal set of constraints or equations such that the problem generally has finite number of solutions.*

Minimal cases to different geometric problems are of particular interests. First of all, identifying such minimal cases as well as understanding the solution space of these cases shed light on the intrinsic structures of a problem. In practice, solving these minimal cases are essential for robust parameter estimation in computer vision [95] and sensor network calibration.

2.4 Algebraic Geometry

Many of the geometric problems studied in this thesis lead to systems of polynomial equations. Tools and techniques in algebraic geometry are key to derive solvers for polynomial systems. In this section, we review some of the key definitions in algebraic geometry as well as techniques to solve polynomial systems based on algebraic geometry. For general introduction or detailed proofs of the concepts discussed in this section, one can refer to [26] and [28]. To start with, we consider solving polynomial system as the following problem:

Problem 2.4.1 Given a set of m polynomials $f_i(\mathbf{x}) \in \mathbb{C}(\mathbf{x})$ in n variables $\mathbf{x} = (x_1, \ldots, x_n)$, determine the complete set of solutions to

$$\begin{aligned} f_1(\mathbf{x}) &= 0, \\ &\vdots \\ f_m(\mathbf{x}) &= 0. \end{aligned} \tag{2.12}$$

where $\mathbb{C}(\mathbf{x})$ denotes the collection of all polynomials with coefficients of the monomials in the field of complex numbers $\mathbb{C}$. We denote a monomial $\mathbf{x}^\gamma = x_1^{\gamma_1} x_2^{\gamma_2} \ldots x_n^{\gamma_n}$, where $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_n)$ is the vector of exponents. The degree of $\mathbf{x}^\gamma$ is defined as the sum of the exponents $|\gamma| = \gamma_1 + \cdots + \gamma_n$.

To solve multivariate polynomial systems, we utilize results and techniques from algebraic geometry. We start by introducing several concepts in algebraic geometry that are closely related to polynomial solving. The zero set (solution set) of a system of polynomial equations defines an affine *variety* V. In this thesis, we focus on solving systems of a finite variety i.e. problems with finite number of solutions. The implication of infinite-dimensional variety is also of interest in a theoretical point of

view. The set of polynomials $\{f_i(\mathbf{x})\}$ generates an *ideal* $I = \sum m_i(\mathbf{x}) f_i(\mathbf{x})$, where $m_i(\mathbf{x}) \in \mathbb{C}(\mathbf{x})$. An ideal I is said to be *radical* if I is identical to the set of all polynomials that vanish on V. The equivalence of two polynomials with respect to ideal I is defined as : if two polynomials f and g is equivalent modulo I iff $f - g \in I$ denoted as $f \sim g$. With this, we can define the *quotient space* $\mathbb{C}(\mathbf{x})/I$ as all the equivalence classes modulo I. For a finite and radical ideal I, it is shown in [28] that $\mathbb{C}(\mathbf{x})/I$ is isomorphic to $\mathbb{C}^r$, where $r = |V|$ is the number of solutions to the set of polynomials that form I.

2.4.1 The Action Matrix

The action matrix method is a multivariate extension of the companion matrix for solving uni-variate polynomial equations. To start with, we consider first the linear mapping $T_{a(\mathbf{x})} : f(\mathbf{x}) \mapsto a(\mathbf{x}) f(\mathbf{x})$ in the r-dimensional $\mathbb{C}[\mathbf{x}]/I$ where $a(\mathbf{x}) \in \mathbb{C}(\mathbf{x})$. In this thesis, we limit the choice of $a(\mathbf{x})$ to be a monomial instead of any polynomial and we call $a(\mathbf{x})$ as *action monomial*. Given that $\mathbb{C}(\mathbf{x})/I$ is finite dimensional, one can choose a linear basis of monomials $\mathcal{B} = \{\mathbf{x}^{\alpha_1}, \ldots, \mathbf{x}^{\alpha_r}\}$ for $\mathbb{C}(\mathbf{x})/I$. Now the mapping $T_{a_{(\mathbf{x})}}$ can be represented as a $r \times r$ matrix $\mathbf{M}_a$, which is the so-called *action matrix*. The solutions of the polynomial system are closely related to the action matrix. The eigenvalues of $\mathbf{M}_a$ are the values of $a(\mathbf{x})$ evaluated at the solution points i.e. V. Also, the eigenvectors of $\mathbf{M}_a^T$ are the values of the basis monomials in $\mathcal{B}$ evaluated at the solution points.

2.4.2 Constructing Action Matrix

In this section, we describe several important techniques for constructing the action matrix in a numerically stable manner including the single eliminate scheme and the basis selection technique.

Single Elimination

The single elimination technique has been widely adapted e.g. [20, 64]. It starts by multiplying the equations in (2.13) by a set of multiplication monomials and produces an equivalent and expanded set of equations. This is in contrast to the Buchberger's algorithm for computing Gröbner bases where equations are generated incrementally. By stacking the coefficients of the expanded set of equations in a coefficient matrix $\mathbf{C}_{\text{exp}}$ which is usually called *elimination template*, we have

$$\mathbf{C}_{\text{exp}} \mathbf{X}_{\text{exp}} = 0. \tag{2.13}$$

where $\mathbf{X}_{\text{exp}}$ is a vector of the set of monomials in the expanded equations. In general, the set of multiplication monomials can be chosen such that the resulting equations are all up to a certain degree.

Basis Selection

To select $\mathcal{B}$ in a numerically stable way, several techniques for basis selection were described in [20]. In the following, we present the column-pivoting scheme for basis selection which has been shown to be a good tradeoff between speed and numerical stability. For all the polynomial solvers derived in this thesis, we have applied the column-pivoting basis selection to improve the numerical stability of solvers.

To enable basis selection, one first partition the set of all monomials $\mathcal{M}$ occurring in the expanded set of equations as $\mathcal{M} = \mathcal{E} \bigcup \mathcal{R} \bigcup \mathcal{P}$. Specifically, $\mathcal{P}$ represent a subset of the set of monomials (*permissible* set) that remain in $\mathcal{M}$ after multiplying with $a(\mathbf{x})$. The *reducible* set $a(\mathbf{x})\mathbf{x}^{\alpha_k} \notin \mathcal{P}$ for $\mathbf{x}^{\alpha_k} \in \mathcal{P}$ is denoted as $\mathcal{R}$. We denote the remaining monomials as the excessive set $\mathcal{E}$. By reordering the monomials such that $\mathcal{E} > \mathcal{R} > \mathcal{P}$, we obtain

$$\begin{bmatrix} \mathbf{C}_{\mathcal{E}} & \mathbf{C}_{\mathcal{R}} & \mathbf{C}_{\mathcal{P}} \end{bmatrix} \begin{bmatrix} \mathbf{X}_{\mathcal{E}} \\ \mathbf{X}_{\mathcal{R}} \\ \mathbf{X}_{\mathcal{P}} \end{bmatrix} = 0. \tag{2.14}$$

Traditionally, the Gröbner bases are used to generate $\mathcal{B}$ for some ordering on the monomials (e.g. *grevlex* [64]). In this case, the permissible set is the same as basis set. The key idea of of [19] is to select $\mathcal{B}$ adaptively from a permissible set $\mathcal{P}$ where $|\mathcal{P}| > r$. Note that, in practice, the subset $\mathcal{P}$ is chosen from the permissible set as the last $n_{\mathcal{P}}(> r)$ monomials based on *grevlex* ordering. To achieve this, a scheme involving two linear elimination steps was proposed. The first step eliminate the monomials in $\mathcal{E}$:

$$\begin{bmatrix} \mathbf{U}_{\mathcal{E}_1} & \mathbf{C}_{\mathcal{R}_1} & \mathbf{C}_{\mathcal{P}_1} \\ 0 & \mathbf{U}_{\mathcal{R}_2} & \mathbf{C}_{\mathcal{P}_2} \\ 0 & 0 & \mathbf{C}_{\mathcal{P}_3} \end{bmatrix} \begin{bmatrix} \mathbf{X}_{\mathcal{E}} \\ \mathbf{X}_{\mathcal{R}} \\ \mathbf{X}_{\mathcal{P}} \end{bmatrix} = 0, \tag{2.15}$$

where $\mathbf{U}_{\mathcal{E}_1}$ and $\mathbf{U}_{\mathcal{R}_2}$ are upper triangular. One can safely remove the top rows of the coefficient matrix involving the $\mathcal{E}$, which gives

$$\begin{bmatrix} \mathbf{U}_{\mathcal{R}_2} & \mathbf{C}_{\mathcal{P}_2} \\ 0 & \mathbf{C}_{\mathcal{P}_3} \end{bmatrix} \begin{bmatrix} \mathbf{X}_{\mathcal{R}} \\ \mathbf{X}_{\mathcal{P}} \end{bmatrix} = 0. \tag{2.16}$$

In the second elimination step, the goal is to reduce $\mathbf{C}_{\mathcal{P}_3}$ into upper triangular matrix. In [19], column-pivoting QR is utilized, which introduces a permutation $\mathbf{C}_{\mathcal{P}_3}\Pi$, where Π is a permutation matrix. With such permutation, the numerical stability is improved. The basis is thus selected as the last r monomials after the reordering of Π i.e. $\begin{bmatrix} \mathbf{X}_{\mathcal{P}'} & \mathbf{X}_{\mathcal{B}} \end{bmatrix}^T$. This gives

$$\begin{bmatrix} \mathbf{U}_{\mathcal{R}_2} & \mathbf{C}_{\mathcal{P}_4} & \mathbf{C}_{\mathcal{B}_1} \\ \mathbf{0} & \mathbf{U}_{\mathcal{P}_3} & \mathbf{C}_{\mathcal{B}_2} \end{bmatrix} \begin{bmatrix} \mathbf{X}_{\mathcal{R}} \\ \mathbf{X}_{\mathcal{P}'} \\ \mathbf{X}_{\mathcal{B}} \end{bmatrix} = 0. \tag{2.17}$$

To this end, monomials in $\mathcal{R}$ and $\mathcal{P}'$ are linear combinations of monomials in $\mathcal{B}$:

$$\begin{bmatrix} \mathbf{X}_{\mathcal{R}} \\ \mathbf{X}_{\mathcal{P}'} \end{bmatrix} = - \begin{bmatrix} \mathbf{U}_{\mathcal{R}_2} & \mathbf{C}_{\mathcal{P}_4} \\ \mathbf{0} & \mathbf{U}_{\mathcal{P}_3} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{C}_{\mathcal{B}_1} \\ \mathbf{C}_{\mathcal{B}_2} \end{bmatrix} \mathbf{X}_{\mathcal{B}}. \tag{2.18}$$

By finding the corresponding indices of the monomials $\{a(\mathbf{x})\mathbf{x}^{\alpha_k} \mid \forall \mathbf{x}^{\alpha_i} \in \mathcal{B}\}$ in $[\mathbf{X}_{\mathcal{R}} \quad \mathbf{X}_{\mathcal{P}'}]$, the action matrix $\mathbf{M}_a$ can be extracted from the linear mapping (2.18).

Part I

Solving Systems of Polynomials Equations

Chapter 3
Optimizing Polynomial Solvers

The state-of-the-art method for constructing polynomial solvers is based on the use of numerical linear algebra on large but sparse coefficient matrix that represents the original equations multiplied with a set of monomials. The key observation in this chapter is that the speed and numerical stability of a solver depends heavily on (i) the set of multiplication monomials (ii) the set of so called permissible monomials from the basis monomials of the corresponding quotient ring are chosen. We show in this chapter that, optimizing with respect to these two factors can gain both computational efficiency and significant improvements in numerical stability compared to state-of-the-art polynomial solvers.

3.1 Motivation

The single elimination template and the column-pivoting basis selection techniques discussed in Section 2.4.2 give generally good performance with respect to speed and numerical accuracy for most minimal problems in computer vision [20]. However, there are two parameters that are chosen adaptively for different problems (1) the set of multiplication monomials and (2) the set of permissible monomials. For different problems, there is no general criterion on how to choose the set of multiplication monomials to construct the single elimination template. In [19, 20], an elimination template is constructed for a specific problem by fine tuning manually the set of monomials based on numerical stability. On the other hand, in [64], the multiplication monomials are chosen automatically by removing equations from an initial large elimination template while keeping the solver well-conditioned. For the permissible set, it was suggested that one should choose it as large as possible, since this in theory should provide better chance getting stable column-pivoting QR. One may ask, will we gain more numerical stabilities by adding more equations to an elimination template or using a smaller set of permissible monomials?

We present here an example to show the effects of these two factors on current state-of-the-art polynomial solvers. The problem we study is the estimation of fundamental matrix for uncalibrated cameras with radial distortion [21], which will be

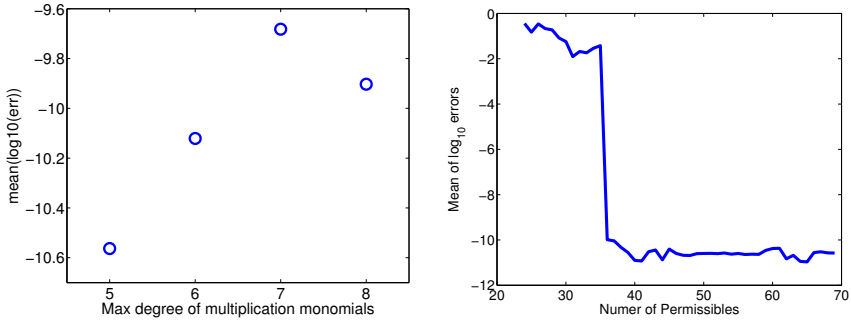


Figure 3.1: Numerical accuracy (mean of log_{10}(errors)) of the solver in [21] for uncalibrated camera with radial distortion. Left: With multiplication monomials of degree 5 (393×390), 6 (680×595), 7 (1096×865) and 8 (1675×1212). The numbers in the parenthesis are the sizes of corresponding C_{exp}. Right: Vary the number of permissible monomials. Note the minimal size of the permissible set is 24 which is the number of solutions.

discussed in more details in Section 3.3.1. In [21], the original equations are multiplied with set of monomials so that the highest degree of resulting equations is 5. To shed light on the effects of more equations, we allow the highest degree to be 8. We note that adding equations by simply increased degrees will actually hurt the overall numerical accuracy (Figure 3.1, Left). On the other hand, we also vary the size of permissible set used in the solver, where we choose the last few monomials in *grevlex* order similar to [21]. We can see in Figure 3.1 (Right), by reducing the size of permissible set in this way, the solver retains numerical stability for permissible set of large sizes, while lose its accuracy for smaller permissible set (in this case when size is smaller than 36). We will show in the next few sections that one can actually gain numerical accuracy by adding equation if we carefully choose the multiplication monomials. Similarly, we can also achieve similar or even improved accuracy with smaller permissible set.

3.2 Optimizing Size and Accuracy for Solvers

In this section, we describe our method for optimizing numerical accuracy of polynomial solvers. We focus on improving the numerical accuracy of the basis selection method in [19] with respect to the permissible set and the equations of the template.

3.2.1 Permissible Selection

As discussed in 2.4.2, the permissible set $\mathcal{P}$ is the set of monomials where the optimal basis can be selected from. In [19], it is suggested to choose the permissible set as large as possible. Specifically, the permissible set is usually chosen as last few monomials with *grevlex* order. In general, larger permissible set will gives better accuracy since it gives high freedom in choosing the optimal basis. However, for some problems, it is possible to obtain slightly better numerical accuracy with smaller set of permissible as indicated by the local minima in Figure 3.1 (Right) . However, it is not clear how the size of permissible set will affect the overall numerical accuracy of the final solvers. On the other hand, it should also be noted that the larger the permissible set is, we need in theory more equations for the template to be solvable. This is related to the fact that we need at least $|\mathcal{P}| + |\mathcal{R}| - r$ equations to yield the system in (2.17). Therefore, to generate solvers with smaller elimination template, one step to go is to select a set of permissible monomials that is more compact while keeping the solvers numerically stable.

Choosing the optimal permissible set of size K is a combinatorial problem, and it is difficult to fully access the optimality of such sets with respect to numerical accuracy. Our approach here is to start with a over-complete set of permissible monomials. We then run the solvers as in [19] with basis selection for N random problem examples. We collect for each run the basis monomials selected for the specific problem in $\mathbf{B} = \{\mathcal{B}_1, \ldots, \mathcal{B}_N\}$. In this way, we have information of what monomials are selected as basis monomials for random problems. Typically, in our experiments, we find no unique optimal set of monomials among the permissibles, which are selected by all the random problems. One criterion for selecting a smaller permissible set of size K is to choose the most selected K monomials. The drawback of this is that such monomials as a set might have never been selected as basis in any random runs. To avoid such cases, we formulate this as a optimization problem. Essentially, a subset $\mathcal{P}^*$ of $\mathcal{P}$ would retain numerical accuracy if it is a superset of as many as basis sets as possible. Therefore, for optimal $\mathcal{P}^*$ of size K, we have the following optimization problem.

Problem 3.2.1 Given a set $\mathcal{P}$ and a collection of sets $\mathbf{B} = \{\mathcal{B}_1, \ldots, \mathcal{B}_N\}$ where $B_i \subseteq \mathcal{P}$ for $i = \{1, \ldots N\}$, find $\mathcal{P}^* \subseteq \mathcal{P}$,

$$\begin{aligned} &\max_{\mathcal{P}^*} \sum_{i=1}^{N} \delta(\mathcal{B}_i \subseteq \mathcal{P}^*), \\ &s.t.\ |\mathcal{P}^*| = K, \end{aligned} \tag{3.1}$$

where $\delta(.)$ is 1 for true boolean expression and 0 otherwise.

This is a hard problem itself. However, since $|\mathcal{P}|$ is generally small for most minimal problems, we can solve it with branch and bound in reasonable time.

3.2.2 Equation Removal

In all previous polynomial solvers applying single elimination template, the equations in an elimination template are generated with redundancy. This will affect both the speed and the overall numerical accuracy of the solvers. This is because all previous methods involve steps of generating upper triangular form of partial or full template, with LU or QR factorization or Gauss-Jordan elimination. For large template with many equations, this step is slow and might be numerically unstable. For [19–21], the multiplication monomials are manually tuned for different minimal problems. Both [64] and [78] investigated the possible techniques of automatic removing equations while keeping the solvers numerical stable. They showed that one can remove equations along with corresponding monomials without losing numerical stability and even gain better numerical performance with single precision for certain minimal problem.

To further improve numerical stability of polynomial solvers, we apply a simple greedy search on what equations to remove while keeping the numerical accuracy of solvers as good as possible. Given a redundant set of equations, we try to find the best equation to remove and then iterate until no equations can be removed i.e. when removing any of the equations makes the template unsolvable. Specifically, we first remove a candidate equation from the original template. We then measure the mean log_{10}-errors of the resulting template on a random set of problem examples (as training data). After running through all equations, we remove the equation without which the resulting solver gives the lowest mean errors. Generally, we can use any polynomial solver to evaluate the errors at each removal step. We have chosen the solver with basis selection based on column pivoting [21] since it generally gives better numerical accuracy.

By exploiting the fact that the set of excessive monomials do not contribute to the construction of the action matrix, we can remove certain equations without any local search and maintain the numerically stability of template. Specifically, when expanding the equations, there can be some excessive monomials that appear only in one of the equations. We call these excessive monomials as *singular excessive monomials*. For the first elimination in (2.15), if an equation contains singular monomials, removing it will not affect the overall numerical condition of the LU or QR with proper pivoting. Therefore, any equation containing singular excessive can be safely removed. One can always apply this trimming step within each local search step to speed up the removal process.

It could be good to start with even more equations which might introduce better numerical conditioning for the first linear elimination in (2.15). To achieve this, the greedy removal step is crucial to select equations to improve the overall numerical accuracy. We have not exploited the numerical routines to remove columns as in [78]. Here, we simply remove columns that become all zeros after certain equations are removed.

3.2.3 Optimization Scheme

Given the tools introduced in the previous sections, we should be able to optimize the polynomial solvers by finding the best combination of permissible set and equations for a specific problem. This is a difficult combinatorial problem. Therefore, as a first attempt, we here first optimize the permissible set by fixing the set of equations. We select the optimal permissible sets of different sizes on the original template (no equation removal). Then we apply greedy search to remove equations with these optimal permissible sets. All these training are done with a fixed set of random problem examples. By investigating the error distributions of all these solvers, the size of template (number of equations, number of monomials), we will be able to choose these optimized solvers with trade-off in accuracy and speed. Generally, we have the following steps for optimizing polynomial solver for a minimal problem:

Algorithm 3.2.1 Polynomial Solver Optimization

Input: Expanded coefficient matrix $\mathbf{C}_{\text{exp}}$, initial permissible set $\mathcal{P}$, a training set of problem examples, size of the permissible set K and threshold parameter ϵ.
Output: Permissible set $\mathcal{P}_K^*$ and elimination template $\mathbf{C}_{\text{slim}}$.

1. Perform permissible selection to choose $\mathcal{P}_K^*$ for $r \leq K \leq |\mathcal{P}|$ as in Problem 2.
2. For each $\mathcal{P}_K^*$, perform equation removal, initialize $\mathbf{C}_{\text{slim}} = \mathbf{C}_{\text{exp}}$,
   (a) Remove equation(s) containing singular excessive monomials
   (b) For each remaining rows i in $\mathbf{C}_{\text{slim}}$, evaluate the average errors of the solver with $\mathbf{C}_{\text{slim}/i}$ on a random subset of problem examples
   (c) Update C_{slim} by removing the row i^* that gives lowest mean error
   (d) Go back to (a)-(c) if the lowest average error in (c) is less than ϵ or a predefined number of iterations is reached.

Note here if the resulting solver after equation removal fails during the iterations in step 2, we set the error to be infinite. Therefore, if we choose ϵ to be large enough, it will be similar to using solvability as removal criterion in [64].

3.3 Experimental Validation

In this section, we apply our method on different minimal geometry problems in computer vision. The first two problems are the estimation of fundamental or essential matrix with radial distortion for uncalibrated or calibrated cameras [21, 66]. The third problem is three-point stitching with radial distortion [18, 49, 78]. All these prob-

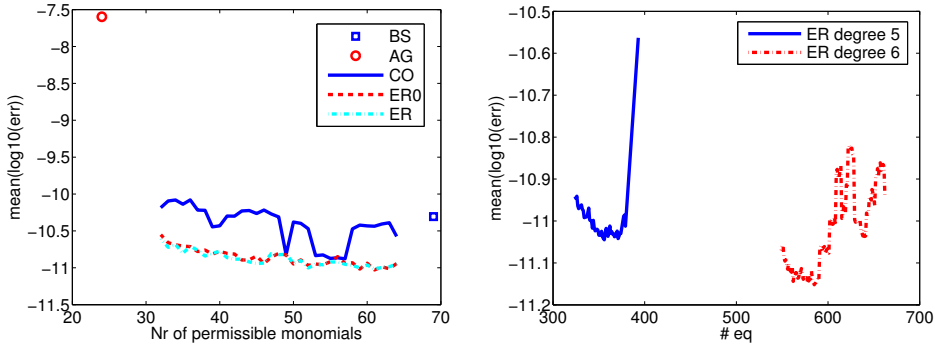


Figure 3.2: Nine-point uncalibrated cameras with radial distortion. Left: Effects of permissible selection and equation removal on numerical accuracy of the polynomials solver. BS - basis selection with column-pivoting [19] , AG - automatic generator [64]), CO - permissible selection with combinatorial optimization, ER0 - ours equation removal without removing equations with singular excessive monomials, ER - same as ER0 but with extra removal on singular excessive equations. Right: Effects of removing equations with fixed permissible (a) ER - degree 5, 393 $\times$ 390 template and (b) ER - degree 6, 680 $\times$ 595 template.

lems has been studied before and were shown to be challenging in the sense that it is difficult to derive numerically stable solvers. We compare our method with previous methods with respect to both numerical accuracy as well as the size of template. The state-of-the-art method is the column-pivoting basis selection [20] which is also the building block of our method. We also compare with other two methods [64, 78] with template reduction procedures. In our implementation of a general solver with column-pivoting basis selection, we have used QR factorization to perform all the elimination steps. Our method involves a training stage where we perform equation removal and permissible selection on a fixed set of random problem examples. For all the comparisons, we test the resulting solvers on an independent set of random problems.

3.3.1 Nine-Point Uncalibrated Radial Distortion

The problem of estimating fundamental matrix with radial distortion was studied in [66]. Numerically stable solutions were provided by [21, 64]. The minimal case for this problem is 9-point correspondences. In general, there are 10 equations and 10 unknowns. By linear elimination, one can reduced the polynomial system to 4 equations and 4 unknowns [66]. The reduced system has 24 solutions. In [64], the equations are first expanded to 497 equations which are then reduced to 179 equations

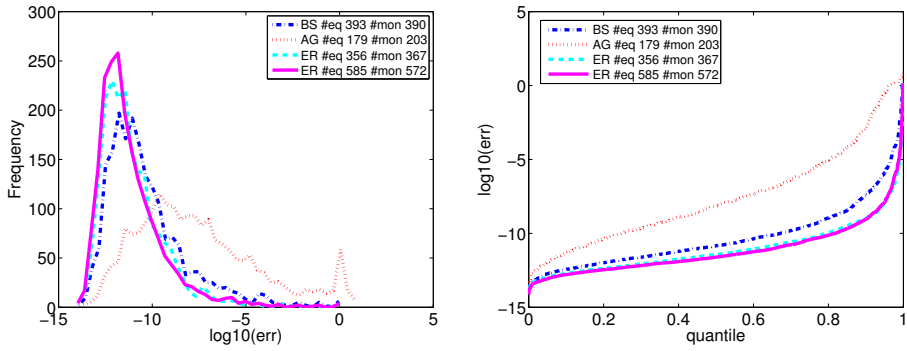


Figure 3.3: Histogram (left) and quantiles (right) of log_{10} (errors) of different solvers for the uncalibrated radial distortion problem BS - [19], AG - [64] and ER - equation removal (ours) with different initial templates.

and 203 monomials. There is no basis selection in this method and the basis set is chosen from the output of Macaulay2, which is of size 24. On the other hand, in [21], a fine-tuned template of 393 equations and 390 monomials are used. In this case, we use the template in [21] as our starting point. We denote this as the original template. For the set of permissible monomials, we choose also exactly the same set as in [21] which is of size 69. We note that further increasing the set of permissible makes the template unsolvable.

Selecting Permissibles

Following the steps in Section 3.2.1, we first run the solver for 393 equations and 390 monomials, and with the set of permissible of size 69 on 5000 random problem examples. First of all, we notice that there are monomials that are never selected as basis by any problem examples. Ideally, we can easily remove these monomials from the permissible set. However, we need to force $\{x_1, x_2, x_3, x_4, 1\}$ to be in the permissible set even if they are never selected. This is because without these monomials we can not extract the solutions of the unknowns $\{x_1, x_2, x_3, x_4\}$ with the method in [20][1]. Therefore, excluding $\{x_1, x_2, x_3, x_4, 1\}$, we are able to remove 8 monomials that are never selected from the permissible set. To this end, we have the possible permissible set of size 61. We then continue to solve the problem in (3.1) with varying K. We manage to solve the problem for K up to 27 with branch and bound. We can see the effects of permissible selection in Figure 3.2 (Right, blue - solid), the solver is still stable using only smallest permissible set of size 32 (we force the 5 monomials to

[1]Note that in general, this is not required for action matrix method e.g. see [27] or Chapter 4.

be included) which is better than without basis selection (Figure 3.1). The improvement gained by permissible selection is more significant for permissible set of smaller size. We also note that permissible selection alone fails to improve the accuracy of the solver consistently.

Removing Equations

We first study the interplay between equation removal and permissible selection. To do this, we perform equation removal on the template with the optimal permissible sets of different sizes from branch and bound. We can see that by performing several steps of equation removal (10 steps in this case), one get consistent improvement of solvers for different optimal permissible sets (ER0 in Figure 3.2). This suggests that one need to combine permissible selection and equation removal to get compact and numerical stable solvers. We also show that removing equations containing singular excessive monomials have little effect on the numerical stability (ER in Figure 3.2). Therefore, we can always apply this to speed up the removal process.

To further understand the mechanism of removing equations, we fix the size of permissible set to 69. We work with both template $\mathbf{T}_1$ (393×390) and the further expanded template $\mathbf{T}_2$ (680×595 up to degree 6). We then proceed to remove equations using local search described in Section 3.2.2. For each removal step, we evaluate the mean log-errors on 100 random samples and we remove the equation without which gives the best numerical accuracy. We can see that the greedy search improves both templates at the beginning, and the templates get worse when more equations are removed (Figure 3.2, right)[2]. Ideally, by optimization, the large template should converge to better template with smaller size. This reason that we do not achieve this ideal situation there is because of the nature of greedy optimization as well as the randomness involved in the training data. Nevertheless, we can see that by removing equations using the greedy search, we can improve significantly the numerical behavior of $\mathbf{T}_2$. Note that the initial mean log_{10}(errors) of $\mathbf{T}_2$ (Figure 3.1, left - degree 6) is around -10.10. We can reduce the size $\mathbf{T}_2$ to 585×572 (local minima in Figure 3.2, right) while in the meantime improve the accuracy to -11.15. This is even slightly better than the best template reduced from $\mathbf{T}_1$ (356×367). This indicates that one could improve numerical accuracy with large template if one carefully optimize the set of equations. We also show the distribution of test errors for solvers producing the lowest errors from our training comparing to state of the art in Figure 3.3. We can see that both our optimized solvers improve over [19] on accuracy. From the iteration of optimization, one get a spectrum of solvers which can be chosen with preference to speed or accuracy.

[2]The initial errors (see Figure 3.1, left) are omitted for better visualization

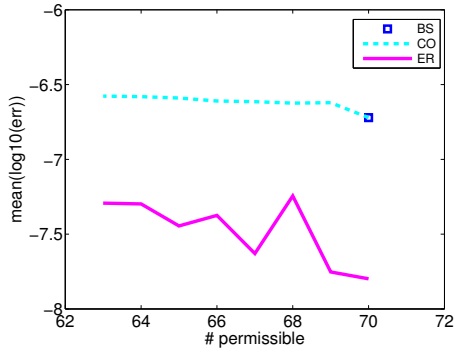


Figure 3.4: Six-point uncalibrated cameras with radial distortion. Effects of permissible selection and equation removal. BS - [19], CO - permissible selection with combinatorial optimization, ER - 5 steps of equation removal with permissible monomials from CO.

3.3.2 Six-Point Calibrated Radial Distortion

For estimation of essential matrix for calibrated cameras with radial distortion, the minimal case is 6 points [66]. The polynomial system for the minimal problem consists of 16 equations in 9 unknowns. It can be reduced to 11 equations in 4 unknown with linear elimination, which has 52 solutions. In [64], a solver with template of 238 equations and 290 monomials is automatically generated. While in [21], a template with 320 equation and 363 monomials is used after fine tuning the degree of monomials multiplied with. For this problem, we do not find the tuning parameters for the smaller template (320 by 363). Therefore, we work with a larger template reported in the paper (356 by 378, with monomials up to degree eight).

We notice in Figure (3.4) that the initial template (356 by 378) is fairly unstable compared to the one reported in [21]. Here we perform first permissible selection on the template. We observe that reducing the permissible size hurts the numerical accuracy even with permissible selection. The equation removal step gives consistent improvement on all permissible sizes. We illustrate this example here to show that the local method can be applied to numerically unstable template and still gives improvements.

3.3.3 Three-Point Stitching

For two cameras with common focal point, we can solve for the unknown rotation, the unknown but common radial distortion and focal length. The minimal case for this problem is 3 point correspondences [49]. A numerically stable solver based on action

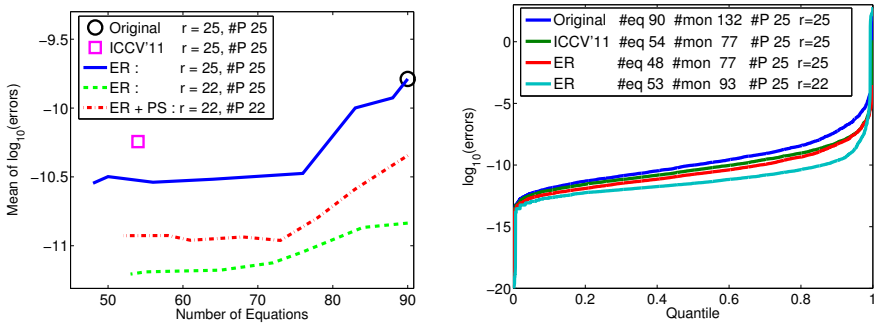


Figure 3.5: Performance of different polynomials solvers on three-point stitching problem. Left: log_{10} (errors) with respect to number of equations. Original - 90×132 template as in [18], ICCV'11- optimization with sequential equation and monomial removal [78] , ER - our equation removal and PS - permissible selection. Right: (best view in colors) Quantile for the log_{10}(errors).

matrix method was proposed in [18]. The polynomial system contains 2 equations in 2 unknowns with 18 solutions. By multiplying the equations with monomials up to degree eight, the expanded template consists of 90 equations in 132 monomials. For this problem, a truncation technique [20] is also applied to obtain numerical solver. It is essentially a way to gain numerical stability by allowing false solutions. Specifically, instead of 18 (number of solutions), the size of the basis r is chosen to 25. In this case, the size of permissible set is also 25. To further improve numerical stability and speed, in [78], a reduction procedure is performed on the same template which is trimmed down to 54 equation in 77 monomials. The resulting smaller template tends to give more stable solvers that are also faster.

Removing Equations

For original solver in [18], the permissible set used was of the same size of the basis (both are 25). Therefore, we do equation removal without permissible selection. We will explore in the next section on possible ways for permissible selection. We can see that one can actually remove fairly many equations and also gain numerical accuracy compared to the original template. Note that our optimized template is also smaller (48×77 for the smallest one) than the one in [78] with slightly better numerical accuracy (mean log-errors -10.56 v.s. -10.27). For fair comparison, similar to [78], we only modify the publicly available solver from [18] with corresponding equations and monomials removed.

For the solver in [18], it turns out the choice for dimension of the basis also affects the numerical accuracy. We studied this by running solvers for different pairs of $|\mathcal{P}| =$

25 and r. We find that the best combination of $|\mathcal{P}|$ and r is using 25 permissible with $r = 22$. It is shown that by simply reducing r, we gain almost one order of magnitude of improvement on accuracy (Figure 3.5 , green - dash line) . Note that this is a local search itself and there might exist better combinations. We leave this as future research direction. We can further reduce the size of the template by removing equations. In this case, the smallest and actually also the best template we have obtained consists of 53 equations in 93 monomials. The equation removal step again improves the numerical accuracy of the solver further.

Permissible Selection

Given that we have a working solver with $|P| > r$ i.e. $|\mathcal{P}| = 25$ and $r = 22$, we can investigate the potential of permissible selection. By solving Problem (2) with $K = 22, 23, 24$, we find that having a large permissible set is always better for this problem. It is here of interest to see what numerical accuracy can we get with the smallest set of permissible monomials. Using 22 permissible selected by branch and bound and equation removal step, the resulting solver (Figure 3.5, red - dotted line) is not as good as the best solver of size 53 with 25 permissibles and $r = 22$. It is a slightly slimmer solver (52×84) with a trade-off in slight decreased numerical stability.

3.4 Conclusions

We have made several observations on the stability of polynomial equation solving using the action matrix method. Firstly, it is shown that adding more equations can improve numerical accuracy, but it only does so up to a point. Adding too many equations can actually decrease numerical accuracy. Secondly, it is shown that the choice of permissible monomials also affects the numerical precision.

We propose two optimization schemes that exploit these observations. Thus we are able to produce solvers that range from very compact, while still retaining good numerical accuracy to solvers that involve larger set of multiplication monomials and larger set of permissible monomials to optimize for numerical accuracy. Our method is easy to implement and is general for different problems. Therefore, it can serve as an initial tool for improving minimal problem involving large templates.

There are several interesting avenues of future research. First, the interplay between numerical linear algebra routines used and the choice of multiplication monomials and permissible monomials should be better understood. Second, the understanding of the mechanisms for numerical accuracy could open up for further improvements.

Chapter 4

Symmetries in Polynomial Systems

In this chapter, we explore another direction for improving the stability and speed of polynomial solvers - symmetry. By detecting and utilizing symmetry in the monomial structure of polynomial equations, we can reduce the size of elimination template for polynomial solvers and develop stable solution extraction techniques for the action matrix method. We first discuss full symmetry in the polynomial systems which is further generalized to partial symmetry i.e. where symmetry lies in a subset of the variables. We derive theoretical results as well novel numerical schemes to utilize these symmetries.

4.1 Full Symmetry

We first define the *full symmetry* with respect to a polynomial system as follows:

Definition 4.1.1. *A polynomial $f(\mathbf{x})$ has full symmetry of type p on the set of variables $\mathbf{x}$, if for each monomial in the polynomial, the sum of the exponents on $\mathbf{x}$ has the same remainder q modulo p as*

A polynomial system is said to have full symmetry of type p on $\mathbf{x}$ if all polynomial equations in the system have symmetry of type p. To understand the characteristics of the solution set to a polynomial system with full symmetry, we define the *symmetry operator* $S^j_{\mathbf{x},p}$ on $\mathbf{x}$ of type p as

$$S^j_{\mathbf{x},p}(\mathbf{x}) = e^{i2j\pi/p}\mathbf{x}, \tag{4.1}$$

where $j \in \mathbb{Z}^+$. By definition, we have $S^p_{\mathbf{x},p}(\mathbf{x}) = \mathbf{x}$. With this, we can define full symmetry of a solution set as follows:

Definition 4.1.2. *A solution set is said to have full symmetry of type p on $\mathbf{x}$ if for each solution $\mathbf{x}^*$, the point $S^j_{\mathbf{x},p}(\mathbf{x}^*)$ is also a solution.*

Example 1. We illustrate the full symmetry of type-2 in the following polynomial system:

$$\begin{cases} x_1^2 - x_2^2 &= 0 \\ x_2^3 x_1 + 1 &= 0 \end{cases}. \tag{4.2}$$

This polynomial system has obviously two real solutions, $(1, -1)$ and $(-1, 1)$. We see that the two solutions are symmetric and only differ in sign.

4.2 Partial Symmetry

In the following, we introduce the concepts of *partial symmetry* in a polynomial and in a solution set, respectively. To facilitate the discussion, for each polynomial $f(\mathbf{x})$ we divide the variables into two sets $\mathbf{x} = \{\mathbf{x}_s, \mathbf{x}_t\}$ such that

$$f(\mathbf{x}) = \sum_k c_k \mathbf{x}_s^{\gamma_k} \mathbf{x}_t^{\beta_k}. \tag{4.3}$$

Here, $c_k \in \mathbb{C}$ are the coefficients of monomials.

Definition 4.2.1. *A polynomial $f(\mathbf{x})$ has partial symmetry of type p on a subset of variables $\mathbf{x}_s$ if for each monomial in the polynomial, the sum of the exponents corresponding to $\mathbf{x}_s$ has the same remainder q modulo p.*

Before we discuss partial symmetry of a solution set, we define the *partial symmetry operator* on $\mathbf{x}_s$ of type p:

$$S^j_{\mathbf{x}_s,p}(\mathbf{x}) = (e^{i2j\pi/p}\mathbf{x}_s, \mathbf{x}_t), \tag{4.4}$$

where $j \in \mathbb{Z}^+$. By definition, we have $S^p_{\mathbf{x}_s,p}(\mathbf{x}) = \mathbf{x}$.

Definition 4.2.2. *A solution set is said to have partial symmetry of type p on a subset of variables $\mathbf{x}_s$ if for each solution $\mathbf{x}^*$, the point $S^j_{\mathbf{x}_s,p}(\mathbf{x}^*)$ is also a solution.*

Example 2. The following polynomial system has partial symmetry ($p = 2$) to x_1, and there is no symmetry for the unknown x_2.

$$\begin{aligned} x_1^2 - x_2^2 - 2 &= 0 \\ x_1^2 - 3x_2 &= 0 \end{aligned} \tag{4.5}$$

For this system, the two pairs of partial symmetric solutions are $\{[\sqrt{6} \;\; 2]^T, [-\sqrt{6} \;\; 2]^T\}$ and $\{[\sqrt{3} \;\; 1]\}^T, [-\sqrt{3} \;\; 1]^T\}$.

With the definitions and theorem of partial symmetry, we can interpret that full symmetry is a special case of partial symmetry. Specifically, it corresponds to the

cases where $\mathbf{x}_s = \mathbf{x}$. The following theorem shows that partial symmetry in a polynomial system has a solution set that is partially symmetric. On the other hand, the existence of a partially symmetric solution set implies that the polynomial system is partially symmetric.

Theorem 4.2.1. *A system of polynomial equations, where every polynomial has partial symmetry of type p on the subset of variables $\mathbf{x}_s$ has a solution set with partial symmetry of type p on $\mathbf{x}_s$. Vice versa, each system of polynomial equations, whose solution set has partial symmetry of type p on $\mathbf{x}_s$, can be written as a set of polynomial equations with partial symmetry of type p on $\mathbf{x}_s$.*

Proof. We first prove that partial symmetry in polynomial systems indicates partially symmetric solution sets. Assume that for a polynomial equation $f(\mathbf{x}) = 0$ in the system, the sum of the exponents $|\gamma_k|$ has constant remainder q modulo p i.e. $\forall k$, $q \equiv |\gamma_k| \bmod p$. Let $b = e^{i2\pi/p}$, we have $b^{|\gamma_k|} = e^{i2\pi|\gamma_k|/p} = b^{q_i}$. Then for $f\big(S^1_{\mathbf{x}_s,p}(\mathbf{x})\big) = f\big((b\mathbf{x}_s, \mathbf{x}_t)\big)$, we have

$$\begin{aligned} f\big((b\mathbf{x}_s, \mathbf{x}_t)\big) &= \sum_k c_k (b\mathbf{x}_s)^{\gamma_k} \mathbf{x}_t^{\beta_k} \\ &= \sum_k b^{|\gamma_k|} c_k \mathbf{x}_s^{\gamma_k} \mathbf{x}_t^{\beta_k} \\ &= \sum_k b^q c_k \mathbf{x}_s^{\gamma_k} \mathbf{x}_t^{\beta_k} \\ &= b^q \sum_k c_k \mathbf{x}_s^{\gamma_k} \mathbf{x}_t^{\beta_k} \\ &= b^q f(\mathbf{x}). \end{aligned} \tag{4.6}$$

Thus if $f(\mathbf{x}^*) = 0$, then $f\big((b\mathbf{x}_s^*, \mathbf{x}_t^*)\big) = b^q f(\mathbf{x}^*) = 0$. One can prove the same for $f\big(S^j_{\mathbf{x}_s,p}(\mathbf{x}^*)\big) = f\big((b^j\mathbf{x}_s, \mathbf{x}_t)\big)$ by induction. This proves the assertion in one direction.

We then prove the existence of partially symmetric solutions indicates partial symmetry in the corresponding polynomial systems. Assume for a certain p that for every solution $\mathbf{x}^* = (\mathbf{x}_s^*, \mathbf{x}_t^*)$ holds also

$$f\big((b^j\mathbf{x}_s^*, \mathbf{x}_t^*)\big) = 0, \quad j = 0, \ldots, p-1. \tag{4.7}$$

Divide the polynomial into p parts according to $|\gamma_\mathbf{k}| \bmod p$ so that

$$f(\mathbf{x}) = g_0(\mathbf{x}) + g_1(\mathbf{x}) + \ldots + g_{p-1}(\mathbf{x}). \tag{4.8}$$

Then we have

$$\begin{aligned} f\big((\mathbf{x}_s, \mathbf{x}_t)\big) &= g_0(\mathbf{x}) + g_1(\mathbf{x}) + \ldots + g_{p-1}(\mathbf{x}), \\ f\big((b\mathbf{x}_s, \mathbf{x}_t)\big) &= g_0(\mathbf{x}) + bg_1(\mathbf{x}) + \ldots + b^{p-1}g_{p-1}(\mathbf{x}), \\ \vdots \quad & \quad \vdots \qquad \vdots \\ f\big((b^{p-1}\mathbf{x}_s, \mathbf{x}_t)\big) &= g_0(\mathbf{x}) + b^{p-1}g_1(\mathbf{x}) + \ldots + bg_{p-1}(\mathbf{x}). \end{aligned}$$

which is equivalent to the following linear system

$$\mathbf{F}(\mathbf{x}) = \mathbf{H}_p\mathbf{G}(\mathbf{x}) \tag{4.9}$$

where

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} f\big((\mathbf{x}_s, \mathbf{x}_t)\big) & f\big((b\mathbf{x}_s, \mathbf{x}_t)\big) & \ldots & f\big((b^{p-1}\mathbf{x}_s, \mathbf{x}_t)\big) \end{bmatrix}^T, \tag{4.10}$$

$$\mathbf{G}(\mathbf{x}) = \begin{bmatrix} g_0(\mathbf{x}) & g_1(\mathbf{x}) & \ldots & g_{p-1}(\mathbf{x}) \end{bmatrix}^T \tag{4.11}$$

and

$$\mathbf{H}_p = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & b & b^2 & \ldots & b^{p-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & b^{p-1} & b^{p-2} & \ldots & b \end{bmatrix}. \tag{4.12}$$

From (4.7), it can be seen that $\mathbf{F}(\mathbf{x}^*) = \mathbf{0}$. Since $\mathbf{H}_p$ is invertible for all p (it is basically the matrix representing the discrete Fourier transform of p-vectors), it follows that $g_j(\mathbf{x}^*) = 0$, for all $j = 0, \ldots, p-1$. Thus if there exist a set of partially symmetric solutions, then it follows that each polynomial $f_i(\mathbf{x})$ can be split into p parts i.e. $g_{ij}(\mathbf{x})$, where each part has $\mathbf{x}^*$ as the solution. □

With the definitions of partial symmetry in polynomial system and solution set, we can interpret that full symmetry is a special case of partial symmetry. Specifically, full symmetry corresponds to partial symmetry with $\mathbf{x}_s = \mathbf{x}$.

4.3 Utilizing Symmetry

In this section, we present general techniques to integrate symmetry into the action matrix method. Specifically, we discuss (i) the concept of symmetric action matrix (ii) the construction of elimination template for problems with symmetry (iii) the solution extraction step given a symmetric action matrix.

4.3.1 Zero Solutions

Before we start the discussion on utilizing symmetry, we discuss a scheme for polynomial systems with zero solution(s). The existence of zero solution can be common of polynomial system with symmetry e.g. a polynomial system of type-2 symmetry with only 1st and 3rd monomials. That is, if none of the polynomial equations in a system has a constant term, then there always exists one zero solution. For a polynomial system with r solutions, one can use the constant 1 as the rth basis monomial in $\mathcal{B}$. To simplify the discussion, assume that x_1 is the action monomial and is chosen as the $(r-1)^{th}$ basis monomial. Then we obtain an action matrix in the form of

$$\mathbf{M} = \begin{bmatrix} a_{1,1} & \dots & a_{1,r-1} & 0 \\ \dots & \dots & \dots & 0 \\ a_{r-1,1} & \dots & a_{r-1,r-1} & 1 \\ 0 & \dots & 0 & 0 \end{bmatrix}. \tag{4.13}$$

The last column of the action matrix corresponds to the mapping of the constant term to x_1. Since no other reduction involves the constant, it follows immediately that $\begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix}^T$ is an eigenvector with eigenvalue 0, i.e. the zero solution. Furthermore, any of the $n-1$ eigenvectors to $\mathbf{M}_a = \{a_{ij}\}$ can be used to produce a corresponding eigenvector to $\mathbf{M}_a^T$. Thus without loss of generality we can consider the eigenvalue problem for the modified action matrix $\mathbf{M}_a$ instead. In practice if there is a zero solution it can be extracted before solving the full system. We assume in the following that the action matrix is reduced to the form where trivial zero solution is removed.

4.3.2 Symmetric Action Matrix

Recall from Theorem 4.2.1 that for a type-p partially symmetric polynomial in $\mathbf{x}_s$, there exist a set of type-p partially symmetric solutions in the form of $(e^{i2\pi j/p}\mathbf{x}_s, \mathbf{x}_t)$ where $j = 0, \dots, p-1$. This suggests that there is a p-fold ambiguity in the r solutions, which can be utilized to simplify the action matrix construction step. To simplify the discussion, we assume that there is no zero solution or one have used the scheme in [3] to remove the zero solution. The idea is then to construct a linear mapping $T_{a(\mathbf{x})} : f(\mathbf{x}) \mapsto a(\mathbf{x})f(\mathbf{x})$ that preserve the underlying partial symmetry of the system. To achieve this, we first need to choose the action monomial $a(\mathbf{x})$ such that the sum of exponents of $\mathbf{x}_s$ in $a(\mathbf{x})$ is p. This follows that the p ambiguous solutions collapse into a single solution point in $a(\mathbf{x})$ which effectively reduces the dimension of the solution space to $r_p = \frac{r}{p}$ [1]. Thus, instead of considering the original solution space, we can express the reduced solution space with a monomial basis $\mathcal{B}_p$ of size r_p. To this end, the size of the reduced action matrix $\mathbf{M}_{r_p}$ with partial symmetry is $r_p \times r_p$.

[1] r is divisible by p, if there exists no zero solution.

The computational advantage of this reduction is that the eigenvalue decomposition step for the action matrix is done on a much smaller matrix.

4.3.3 Elimination Template with Symmetry

The next step is to construct an elimination template with partially symmetry in mind. The idea again is to generate the expanded set of equations that preserve the symmetry in $\mathbf{x}_s$. To facilitate the elimination step for partially symmetric systems, we propose a scheme to achieve this: choose the multiplication monomials that results in the same remainder modulo p (for the symmetric subset) across the set of different equations (not only within each equation). This scheme ensures good overlapping of the expanded monomials between different equations which improves the efficiency (reduced size of the elimination template) and stability of the elimination step. We will illustrate the scheme in Section 4.4.4 with detailed examples.

4.3.4 Extracting Solutions

In action matrix method, once we have constructed the action matrix $\mathbf{M}$, we can extract the solutions from the eigenvectors of $\mathbf{M}^T$. We also know that the each eigenvector $\mathbf{v}$ are values of the basis monomials $\mathcal{B}$ evaluated at a solutions up to an unknown scalar λ. Specifically, we have for each element v_k in $\mathbf{v}$

$$\lambda v_k = x_1^{\gamma_{k1}} x_2^{\gamma_{k2}} \dots x_n^{\gamma_{kn}}. \tag{4.14}$$

To extract the solution from $\mathbf{v}$, the simplest scenario is that all the first-order monomials as well as the constant term i.e. $\{x_1, \dots, x_n, 1\}$ are in the basis monomials. In this case, the solutions can be extracted by reading off the corresponding values of $\{\lambda x_1, \dots, \lambda x_n, \lambda\}$ from $\mathbf{v}$ and the solutions for $\{x_1, \dots, x_n\}$ can be calculated via division by λ.

For type-p partially symmetric cases where $p > 1$, the general idea for solution extraction is to find a mapping from monomials in $\mathcal{B}$ to x_i^p for $i = 1, \dots, n$ and the p-fold ambiguity of x_i's can be solved directly. For example, if we know $v_1 = \lambda x_1 x_2$ and $v_2 = \lambda x_1 x_2^3$ are in the basis, one can calculate the solution values $x_2^2 = v_2/v_1$. The two solutions of x_2 can then be extracted directly. In general, this mapping is not unique. To find one of such mappings automatically, we describe a two-step scheme involving (i) a random sampling step and (ii) solving an integer linear system.

We first introduce an equivalent expression of (4.14). We can treat the unknown constant λ as a proxy unknown x_{n+1} such that

$$v_k = x_1^{\gamma_{k1}} x_2^{\gamma_{k2}} \dots x_n^{\gamma_{kn}} x_{n+1}^{\gamma_{k,n+1}}, \tag{4.15}$$

where $x_{n+1} = 1/\lambda$ and $\gamma_{k,n+1} = 1$.

For $\mathbf{x} = \begin{bmatrix} x_1 & \dots & x_{n+1} \end{bmatrix}^T$ and an integer matrice $\mathbf{A} \in \mathbb{Z}^{r\times(n+1)}$, define the exponential $\mathbf{x}^{\mathbf{A}}$ of $\mathbf{x} : \mathbb{R}^{n+1} \mapsto \mathbb{R}^r$ as

$$\mathbf{x}^{\mathbf{A}} := \exp(\mathbf{A}\log(\mathbf{x})). \tag{4.16}$$

Although the logarithm is ill-defined for complex numbers, the exponential is well-defined since $\mathbf{A}$ has integer coefficients.

Let $\mathbf{A}$ be the integer matrix with elements $A_{kj} = \gamma_{kj}$ as in (4.15). Then we can express the eigenvector $\mathbf{v}$ in the form of exponential map, i.e. $\mathbf{v} = \mathbf{x}^{\mathbf{A}}$. The problem of calculating $\mathbf{x}$ from $\mathbf{v}$ can thus be written as : find all $\mathbf{x}$ such that $\mathbf{v} = \mathbf{x}^{\mathbf{A}}$. If there exists an integer matrix $\mathbf{B} \in \mathbb{Z}^{(n+1)\times(n+1)}$, such that $\mathbf{B}\mathbf{A}_{\text{sub}} = \mathbf{I}$, then there is only one solution. Here, $\mathbf{A}_{\text{sub}}$ is a $(n+1)\times(n+1)$ sub-matrix of $\mathbf{A}$. This solution can be written as $\mathbf{x} = \mathbf{v}_{\text{sub}}^{\mathbf{B}}$, since

$$\mathbf{v}_{\text{sub}}^{\mathbf{B}} = (\mathbf{x}_{\text{sub}}^{\mathbf{A}})^{\mathbf{B}} = \mathbf{x}^{\mathbf{B}\mathbf{A}_{\text{sub}}} = \mathbf{x}^{\mathbf{I}} = \mathbf{x},$$

where $\mathbf{v}_{\text{sub}}$ the subset of $\mathbf{v}$ corresponding to $\mathbf{A}_{\text{sub}}$. One way of generating such a matrix is to search for $(n+1)\times(n+1)$ sub-matrices in $\mathbf{A}$ with determinant 1 or -1. The inverse of such a sub-matrix $\mathbf{A}_{\text{sub}}$, its inverse is also an integer matrix and $\mathbf{B}$ is constructed as the inverse of $\mathbf{A}_{\text{sub}}$. When $\mathbf{A}_{\text{sub}}$ is an identity matrix, it is equivalent to choosing the subset as $\{x_1, \dots, x_n, 1\}$.

In the general case, one may search for an invertible sub-matrix $\mathbf{A}_{\text{sub}}$, whose absolute value of its determinant is as low as possible. For cases with full or partial symmetry of type-p, it is possible to find a sub-matrix with $p = |\det(\mathbf{A}_{\text{sub}})|$. We know that the inverse of $\mathbf{A}_{\text{sub}}$ can be written as

$$\mathbf{A}_{\text{sub}}^{-1} = \frac{\operatorname{adj}(\mathbf{A}_{\text{sub}})}{|\det(\mathbf{A}_{\text{sub}})|},$$

where $\operatorname{adj}(.)$ is the adjoint operator of a matrix. By setting $\mathbf{B} = |\det(\mathbf{A}_{\text{sub}})|\mathbf{A}_{\text{sub}}^{-1}$, we have $\mathbf{B}\mathbf{A}_{\text{sub}} = p\mathbf{I}$. This gives $\mathbf{v}_{\text{sub}}^{\mathbf{B}} = (\mathbf{x}^{\mathbf{A}_{\text{sub}}})^{\mathbf{B}} = \mathbf{x}^{(\mathbf{B}\mathbf{A}_{\text{sub}})} = \mathbf{x}^{p\mathbf{I}}$. From this, we can recover the absolute values of $\mathbf{x}$ (denoted as $\mathbf{x}_0$). Now it is possible to solve for $\mathbf{x}$ up to an unknown phase parameterized by $\mathbf{k}$

$$\mathbf{x} = \mathbf{x}_0 \odot \exp(i\mathbf{k}2\pi/p), \tag{4.17}$$

where $\mathbf{k} \in \mathbb{Z}_p^{n+1}$ and $\odot$ denotes the element-wise multiplication. Thus the absolute value $\mathbf{x}_0$ of $\mathbf{x}$ is well defined and the phase is known up to a type-p uncertainty. Given $\mathbf{v} = \mathbf{x}^{\mathbf{A}}$, by plugging in (4.17) and take logarithm on both sides, we have

$$\log(\mathbf{v}) = \mathbf{A}\log(\mathbf{x}_0) + \mathbf{A}\mathbf{k}(i2\pi/p) + ij2\pi,$$

which can be written

$$\mathbf{A}\mathbf{k} = \underbrace{p(\log(\mathbf{v}) - \mathbf{A}\log(\mathbf{x}_0))/(i2\pi)}_{\mathbf{z}} + pj,$$

which can be interpreted as a system of integer linear equations $\mathbf{A}\mathbf{k} = \mathbf{z}$. It is straightforward to write a solver for such problems based on Gaussian elimination on $\mathbb{Z}_p$. In the case of p being prime is particularly simple. The solution has in general one free parameter and can be written as

$$\mathbf{k} = \mathbf{k}_0 + c\mathbf{k}^* \ , \ \text{with } c \in \mathbb{Z}_p \ ,$$

which gives the solutions after substitution in (4.17) . Note that in our discussions above, we have chosen the mappings $\mathbf{A}_{\text{sub}}$ from monomials in $\mathcal{B}$. In fact, given that the values of monomials (up to an unknown constant) in $\mathcal{R}$ and $\mathcal{P}'$ can be calculated from (2.18), we can actually choose the mappings from $\{\mathcal{R}, \mathcal{P}', \mathcal{B}\}$.

The choice of $\mathbf{A}_{\text{sub}}$ which is found in our algorithm by randomly choosing $(n+1)$ rows from $\mathbf{A}$. Therefore, the resulting mapping for solution extraction is not unique. It has been seen in our experiments that the stability of these mappings varies and most of them can be very unstable. This is due to the fact that most of these mappings involve (i) evaluation of the solutions for monomials of high degrees (ii) numerical operations e.g. division of monomial of high degrees. Therefore, while one can easily use the general technique introduced here for symmetric cases, it is preferable to select problem-specific mappings either by observing the structure in the problems or post-selection step based on criterion like avoiding the existence of monomial of high degrees in the mappings. We will discuss such selection steps for different problems in Section 4.4.

4.3.5 Detecting Symmetry

Based on the Theorem 4.2.1, we describe a simple strategy for detecting partially symmetric polynomial system. Specifically, if d is lowest degree of any monomials in all the polynomial equations, we check for each subset of $\mathbf{x}_s$, whether type-p ($2 \leq p \leq d$) partial symmetry is fulfilled. This is done by checking whether the remainders of the sum of exponents for $\mathbf{x}_s$ modulo p are the same for all monomials in each of the polynomial equations. It is an exhaustive scheme and involves combinatorial search over all subsets of variables. However, for general problems in computer vision where the number of variables are among 2 to 10, the search is completely feasible.

4.4 Applications

In this section, we discuss full symmetry and partial symmetry in the context of geometric problems in computer vision. We present several formulations of these problems that yield polynomial systems with either full symmetry or partial symmetry.

All the problems discussed below involve the rotation matrix $\mathbf{R}$. One choice for

parameterizing $\mathbf{R}$ is using quaternion with $\{a, b, c, d\}$

$$\begin{bmatrix} a^2+b^2-c^2-d^2 & 2bc-2ad & 2ac+2bd \\ 2ad+2bc & a^2-b^2+c^2-d^2 & 2cd-2ab \\ 2bd-2ac & 2ab+2cd & a^2-b^2-c^2+d^2 \end{bmatrix}. \tag{4.18}$$

However, the scale of $\mathbf{R}$ can still be choose freely. There are in general two ways to fix the scale of the parameterization. One is the Cayley representation where one of the variable in $\{a, b, c, d\}$ is set to 1 [45, 76]. The other way is to enforce the unit-norm constraint such that $\|\mathbf{q}\|_2 = 1$ where $\mathbf{q} = [a, b, c, d]^T$. There are advantages and drawbacks for both choices. For the Cayley representation, it generally leads to simpler polynomial systems but it can be degenerate in all cases of 180 degree rotations around the x-, y- and z-axis (corresponding to where one of the variables in $\{a, b, c, d\}$ is 0). As for the fix with unit-norm constraint, it avoids the degeneracy completely but generally results in polynomial systems that are more difficult to solve. In this section, we explore the symmetric structures in the non-Cayley representation. While the formulations avoid certain degeneracy, the resulting polynomial systems have not been solved before due to the difficulty in constructing efficient and stable polynomial solvers. We illustrate the proposed techniques in details for these examples and obtain faster and more stable solvers than previous state-of-the-art general polynomial solvers.

4.4.1 Optimal Perspective-n-Point

Given n ($n \geq 3$) 3D reference points in the object framework and their corresponding 2D projections, determining the orientation and the position of a fully calibrated perspective camera is known as the perspective-n-point (PnP) problem [43]. In [104], a formulation (OPnP) that aims to find the global optimal of an algebraic error is presented. This formulation avoids several degeneracies that are suffered by other previous formulations and implicitly impose the normal constraints for the quaternion parameterization.

Problem 4.4.1 (Optimal Perspective-n-Point) Given n 3D points $\mathbf{X}_i = \left[x_i, y_i, z_i\right]^T$, and their corresponding projections $\mathbf{x}_i = \left[u_i, v_i\right]^T$, find the optimal rotation $\mathbf{R}^*$ and the translation $\mathbf{t}^* = \left[t_x, t_y, t_z\right]^T$ of the camera such that

$$\begin{aligned} \{\mathbf{R}^*, \mathbf{t}^*\} = \arg\min_{\mathbf{R},\mathbf{t}} \quad & \sum_{i=1}^{n} \left((1 + \mathbf{r}_3^T \tilde{\mathbf{X}})u_i - \mathbf{r}_1 \mathbf{X}_i - t_x\right)^2 \\ & \sum_{i=1}^{n} \left((1 + \mathbf{r}_3^T \tilde{\mathbf{X}})v_i - \mathbf{r}_2 \mathbf{X}_i - t_y\right)^2 \end{aligned} \tag{4.19}$$

where $\mathbf{r}_i$ is the ith row of $\mathbf{R}$ ($i = 1, 2, 3$) and $\tilde{\mathbf{X}}_i = \mathbf{X}_i - \bar{\mathbf{X}}$ with $\bar{\mathbf{X}} = \sum_{i=1}^{n} \mathbf{X}_i/n$ denoting the centroid of the 3D points. Given $\mathbf{R}$, the optimal $\mathbf{t}^*$ can be calculated as:

$$\mathbf{t}^* = \begin{bmatrix} \bar{u} + \mathbf{r}_3^T(\frac{\sum_{i=1}^{n} u_i \tilde{\mathbf{X}}_i}{n}) - \mathbf{r}_1^T \bar{\mathbf{X}} \\ \bar{v} + \mathbf{r}_3^T(\frac{\sum_{i=1}^{n} v_i \tilde{\mathbf{X}}_i}{n}) - \mathbf{r}_2^T \bar{\mathbf{X}} \\ 1 - \mathbf{r}_3^T \bar{X} \end{bmatrix} \tag{4.20}$$

By substituting (4.20) to (4.19) and parameterizing $\mathbf{R}$ with the quaternion (4.18), we can arrive at the following optimization problem:

$$\min_{a,b,c,d} \|\mathbf{M}\alpha\|_2^2 \tag{4.21}$$

where $\alpha = \left[a^2, ab, ac, ad, b^2, bc, bd, c^2, cd, d^2, 1\right]^T$ and $\mathbf{M} \in \mathbf{R}^{2n \times 11}$ is a coefficient matrix calculated based on the substitution.

4.4.2 Optimal Euclidean Registration

We then study the Euclidean registration problem given point-point, point-line or point-plane correspondences [80]. This problem has been solved using a branch-and-bound method that exploits the quasi-convexity property of the minimization problem. The mixture of correspondences can be formulated into a unified error function [80]. While the symmetry as well as the solver introduced here can be applied to the unified error function, for simplicity, we discuss the point-plane correspondences only.

Problem 4.4.2 (Optimal Euclidean Registration) Given n 3D points $\mathbf{X}_i$, and their corresponding planes in another coordinate system, each of which is represented by the normal $\mathbf{e}_i$ and a supporting point $\mathbf{Y}_i$, to find the optimal rotation $\mathbf{R}^*$ and translation $\mathbf{t}^*$ such that

$$\{\mathbf{R}^*, \mathbf{t}^*\} = \arg\min_{\mathbf{R},\mathbf{t}} \sum_{i=1}^{n} \left(\mathbf{e}_i^T(\mathbf{R}\mathbf{X}_i + \mathbf{t} - \mathbf{Y}_i)\right)^2. \tag{4.22}$$

Given $\mathbf{R}$, the translation $\mathbf{t}$ can be directly solved as

$$\mathbf{t} = \left(\sum_{i=1}^{n} \mathbf{e}_i \mathbf{e}_i^T\right)^{-1} \sum_{i=1}^{n} \mathbf{e}_i \mathbf{e}_i^T \left(\mathbf{Y}_i - \mathbf{R}\mathbf{X}_i\right). \tag{4.23}$$

After parameterizing $\mathbf{R}$ by the quaternion with norm constraints and plugging $\mathbf{t}$ back into (4.22), we obtain a constraint optimization problem

$$\begin{aligned} \{\mathbf{q}^*\} = \arg\min_{\mathbf{q}} \sum_{i=1}^{n} \left(e_i^T(\mathbf{R}(\mathbf{q})\mathbf{X}_i + \mathbf{t}(\mathbf{q}) - \mathbf{Y}_i)\right)^2, \\ s.t.\ \|\mathbf{q}\|_2^2 = 1. \end{aligned} \tag{4.24}$$

4.4.3 PnL Problem

The Perspective-n-Line (PnL) problem is to estimate the absolute pose of a calibrated camera by using n ($n > 3$) known lines and their image projections. It was studied in [76].

Problem 4.4.3 (Perspective-n-Line) Given n lines with direction $\mathbf{l}_i$ in the world framework, and their corresponding image lines, each of which determines a plane passing through the optical center with normal $\mathbf{e}_i$, to find the optimal rotation $\mathbf{R}^*$ such that

$$\{\mathbf{R}^*\} = \arg\min_{\mathbf{R}} \sum_{i=1}^{n} \left(\mathbf{e}_i^T \mathbf{R} \mathbf{l}_i\right)^2 . \tag{4.25}$$

After the rotation $\mathbf{R}$ is determined, the estimation of translation $\mathbf{t}$ becomes trivial. To parameterize $\mathbf{R}$ by the unit quaternion $\mathbf{q}$ would lead to a constrained optimization problem

$$\begin{aligned} \{\mathbf{q}^*\} = \quad & \arg\min_{\mathbf{q}} \textstyle\sum_{i=1}^{n} \left(\mathbf{e}_i^T \mathbf{R}(\mathbf{q}) \mathbf{l}_i\right)^2 , \\ & s.t. \ \|\mathbf{q}\|_2^2 = 1. \end{aligned} \tag{4.26}$$

4.4.4 Symmetric Systems and Solvers

To solve for global optimal of the geometric or algebraic errors defined in this section, we use the first order optimality condition i.e. to find all stationary points of the error functions [93]. To do that, we calculate the partial derivative of the error functions with respect to the unknowns $\{a, b, c, d\}$ in the quaternion as well as the Lagrange multiplier w for problems with explicit unit-norm constraints. For problems where unit-norm constraints are handled implicitly i.e. (OPnP), we will see in the following discussion that, the resulting polynomial system is fully symmetric while the other two problems are partially symmetric.

Full Symmetry

The resulting polynomial systems with first order optimality for the OPnP problem (4.4.1) consist of 4 equations and can be written in the following matrix form

$$\mathbf{U}\mathbf{w} = 0, \tag{4.27}$$

where $\mathbf{U}$ is a 4×24 coefficient matrix calculated for each specific problem and $\mathbf{w} = [\, a^3, a^2b, a^2c, a^2d, ab^2, abc, abd, ac^2, acd, ad^2, b^3, b^2c, b^2d, bc^2, bcd, bd^2, c^3, c^2d, cd^2, d^3, a, b, c, d \,]$.

It is verified with algebraic geometry tools [39] that a polynomial system with these monomial structures have 81 solutions . The state-of-the-art automatic generated solver [64] without considering symmetry has an elimination template of size 576×656. We observe first that there exist no constant term in any of the equations. Therefore, there is a trivial all-zero solution. On the other hand, for all the variables $\{a, b, c, d\}$, only monomials of 3rd and 1st degrees appear in the equations. From Theorem 4.2.1, we know that this polynomial system is fully symmetric of type-2. Thus, the remaining 80 solutions consist of 40-pairs of symmetric solutions.

To construct a solver that exploit the symmetry, we have derived the following problem-specific procedures. The general technique discussed in Section 4.3.4 for extracting the symmetric solutions from the action matrix is unfortunately both slow and unstable for this problem. First, we handle the zero solution implicitly by ignoring the last row and last column of the action matrix. To extract the 40 independent solutions, we first construct the elimination template by multiplying the 4 original equations with monomials of even degrees. We choose the action monomial to be a 2nd degree monomial to preserve the symmetry. In this way, the two-fold symmetry is preserved. Specifically, we have found that it is sufficient to multiply all monomials of even degrees up to degree 6. This results in an elimination template of size 520×420. Using the similar strategy in [64], the size of the elimination template is reduced to 348×376. We have chosen ac as the action monomial. With this elimination template and the basis selection technique, we are able to stably construct an 40×40 action matrix. The size of permissible set is chosen to be 100. The eigenvectors $\mathbf{v}$'s of transpose of the action matrix are solutions to the set of basis monomials up to an unknown constant. Note that, both the basis and permissible monomials are only of odd degrees by construction.

For the solution extraction step, we further utilize the problem structure to improve numerical stability and speed of the solver. We first utilize the fact that the monomial set $\{a, a^3, a^2b, a^2c, a^2d\}$ are either in the basis or permissible monomial set. This means that we can extract the values of these monomials easily from $\mathbf{v}$'s up to a common unknown constant λ. To start with, we can extract the solutions for a^2 by using values of λa and λa^3. In this way, we can trivially get the two solutions of a. To resolve for the rest of the variables, the values for $\{ab, ac, ad\}$ are calculated with the division of $\{\lambda a^2b, \lambda a^2c, \lambda a^2d\}$ by λa. Given these, we can extract the corresponding solutions for $\{b, c, d\}$ of the two symmetric solutions of a. The same procedure for each of the eigenvector gives the 40 pairs of symmetric solutions.

The potential issue for the aforementioned extraction step is that the division operation becomes degenerate (or unstable) when a is 0 (or close to 0). The way to fix this potential numerical instability is to have one extra extraction step that uses the information on other monomials. Specifically, in a similar manner, we extract another 3 sets of (40 solutions) with the monomial set $\{b, b^3, b^2a, b^2c, b^2d\}$, $\{c, c^2, c^2a, c^2b, c^2d\}$ and $\{d, d^3, d^2a, d^2b, d^2c\}$. Thus, we actually extract four sets of solutions. Since at least

one of $\{a, b, c, d\}$ is non-zero and sufficiently large in absolute value (at least so when compared with the smallest one in absolute value), it is guaranteed that at least one set of the solutions avoids the numerical instability issue. Though it seems that we extract more solutions than needed, most of solutions can be discarded by checking repetitiveness, keeping only the real solutions and checking the Hessian of the solutions. The cost for computing these extra solutions is also negligible compared to the QR factorization or eigenvalue decomposition step. We would like to emphasize that this strategy is designed to improve numerical stability so that the developed GB solver is stable even when some variables of a, b, c, d are 0 (or close to 0).

Partial Symmetry

For optimal Euclidean registration problem in Section 4.4.2, we have the polynomial system of the form

$$\begin{bmatrix} \mathbf{V} & \mathbf{0}_{4\times 4} & -\mathbf{I} & \mathbf{0}_{4\times 1} \\ \mathbf{0}_{1\times 24} & \mathbb{1}_{1\times 4} & \mathbf{0}_{1\times 4} & -1 \end{bmatrix} \mathbf{w} = \mathbf{0}, \tag{4.28}$$

where $\mathbf{V}$ is a 4×24 coefficient matrix calculated from a specific problem and $\mathbf{w}$ is a monomial vector $[a^3, a^2b, a^2c, a^2d, ab^2, abc, abd, ac^2, acd, ad^2, b^3, b^2c, b^2d, bc^2, bcd, bd^2, c^3, c^2d, cd^2, d^3, a, b, c, d, a^2, b^2, c^2, d^2, wa, wb, wc, wd, 1]^T$.

As for the PnL problem in Section 4.4.3, we have the following polynomial system

$$\begin{bmatrix} \mathbf{U} & \mathbf{0}_{4\times 4} & -\mathbf{I} & \mathbf{0}_{4\times 1} \\ \mathbf{0}_{1\times 20} & \mathbb{1}_{1\times 4} & \mathbf{0}_{1\times 4} & -1 \end{bmatrix} \mathbf{w} = \mathbf{0}, \tag{4.29}$$

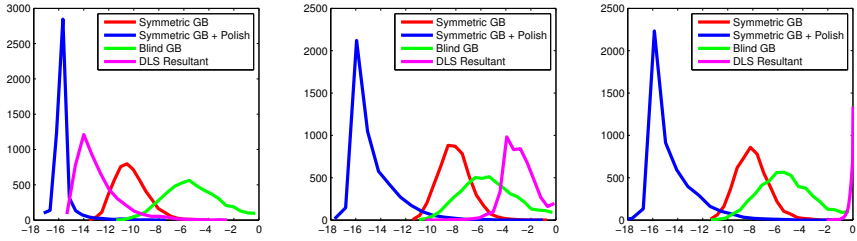
where $\mathbf{U}$ is a 4×20 coefficient matrix calculated from each specific problem and $\mathbf{w}$ is a vector of monomials $[a^3, a^2b, a^2c, a^2d, ab^2, abc, abd, ac^2, acd, ad^2, b^3, b^2c, b^2d, bc^2, bcd, bd^2, c^3, c^2d, cd^2, d^3, a^2, b^2, c^2, d^2, wa, wb, wc, wd, 1]^T$.

By checking the monomials in both systems, we can see that the variables in the subset $\mathbf{x}_s = \{a, b, c, d\}$, only appear with 3rd and 1st degree in the first 4 equations ($p = 2, q = 1$), and only 2nd and 0th degree in the last equation ($p = 2, q = 0$). Thus, these two polynomial systems are both partially symmetric of type 2 to $\{a, b, c, d\}$ according to Theorem 4.2.1.

By using tools in algebraic geometry [33], we verify that there are in general 80 solutions to these two polynomial systems or equivalently 40 pairs of partially symmetric solutions. Using automatic generator in [64] which does not utilize partial symmetry, we obtain general solvers that solve for the 80 solutions directly. The elimination templates of these solvers are of size 1523×1603 and 688×788 for the optimal Euclidean registration and the PnL, respectively. Note that the two elimination templates are obtained after the build-in optimization for template size [64]. There is little possibility to reduce the sizes of the templates further.

Now we discuss in details the construction of our partially symmetric solver. It turns out that the same elimination template and solution extraction scheme work for both problems due to their similarity in structures. To start with, we can choose any quadratic monomials in $\{a, b, c, d\}$ as the action monomial to utilize type-2 partial symmetry. There is no significant effect for the choice of action monomial on the numerical stability for this problem. Here we use a^2. The second step is to choose the set of multiplication monomials for generating the elimination template. The idea is to ensure the expanded set of monomials after multiplication coincide between different equations. This will facilitate the elimination step so that the numerical stability is improved. To start with, for the first 4 equations, we choose $\{\mathcal{H}_1, \mathcal{H}_3, a^2\mathcal{H}_3, w\mathcal{H}_1, w\mathcal{H}_1,\ wa^2\mathcal{H}_3, w^2\mathcal{H}_1, w^2\mathcal{H}_1, w^2a^2\mathcal{H}_3\}$ as the set of multiplication monomials. Here $\mathcal{H}_k$ denotes the set of monomials in $\{a, b, c, d\}$ where the sum of exponents is k. In the resulting expanded set of monomials, the sum of exponents for variables $\{a, b, c, d\}$ in the monomials are all even. Correspondingly, the multiplication monomials for the last equation are chosen as $\{\mathcal{H}_2, \mathcal{H}_4, a^2\mathcal{H}_4, w\mathcal{H}_2, w\mathcal{H}_4, wa^2\mathcal{H}_4, w^2\mathcal{H}_2, w^2\mathcal{H}_4, w^2a^2\mathcal{H}_4, w, w^3\}$. This also yields a set of expanded monomials where the sum of exponents for variables $\{a, b, c, d\}$ are even. After these two expansion, we have generated a stable elimination template of size 770×854, which is already much smaller than the 1523×1603 elimination template for the optimal Euclidean registration problem. With further tuning with similar equation removal technique in [64], we obtain a more compact elimination template of size 433×487. This is used for all our experiments later. With this elimination template, to further enhance the numerical stability, we follow the column-pivoting basis selection technique (the size of the permissible set is set to 60) and construct a 40×40 action matrix.

The last remaining step is to extract solutions from eigenvectors of the transpose of the action matrix. To enhance the numerical stability, we have derived the following extraction scheme. The first observation is that, for these two problem, one can enforce the constant term i.e. 1 to be in $\mathcal{B}, \mathcal{R}$ or $\mathcal{P}'$ without breaking the type-2 partially symmetry. Extracting the values for the unknown constant λ's is simply reading off the corresponding values from the vectors. We note also that $\{a^2, ac, ab, ad\}$ are expressible by linear combination of the basis, which means that one can obtain their values up to an common unknown constant for each solution i.e. $\{\lambda a^2, \lambda ac, \lambda ab, \lambda ad\}$. Thereafter, the solutions of a^2 can be calculated by division of the corresponding values of λa^2 by λ. We retrieve the two sets of 40 solutions for a by taking square root. Given that λ and a are known, the values for $\{b, c, d\}$ can be extracted using division of $\{\lambda ab, \lambda ac, \lambda ad\}$ by λa. Again, the 2-fold ambiguity of the solutions are handled naturally in the extraction step which is much simpler than solving it directly as in the general methods. In case of Cayley degeneracy i.e. where $a = 0$ ($a \approx 0$) , the division when we extract $\{b, c, d\}$ is ill-conditioned. In those cases, we can extract the solutions with $\{b^2, ba, bc, bd\}$, $\{c^2, ca, cb, cd\}$ or $\{d^2, da, db, dc\}$ in a similar way to avoid such degeneracy. Note that these extraction steps are fast given that the bottleneck is



(a) Random (b) Near-Cayley-degenerate (c) Cayley-Degenerate

Figure 4.1: Numerical stability of the polynomial system solvers. The investigated solvers include the blind GB solver without utilizing symmetry (Blind GB), the GB solver using two-fold symmetry (Symmetric GB), the Symmetric GB followed by one damped Newton polishing step (Symmetric GB + Polish) and the resultant based solver used in DLS [45]. The horizontal axis shows the log10 value of the absolute error between the ground truth of unit-norm quaternion and the estimated quaternion after normalization, while the vertical axis shows the counts over 5,000 independent runs. This figure is best viewed in color.

generally in the elimination step. Therefore, we can extract all possible solutions in an very efficient way to avoid degeneracy. This is much superior to the schemes in [45] which requires solve several different polynomial problems with specific solvers.

4.5 Experiments

In this section, we investigate the numerical stability, speed as well as noise sensitivity of symmetric polynomial solvers.

PnP Problem To generate synthetic examples for the PnP problem, we assume a virtual perspective camera with image resolution of 640×480 pixels and focal length 800 pixels. The principle point lies in the image center and n 3D reference points are randomly generated in the camera framework. The 3D reference points are randomly distributed in the x, y and z with range of $[-2, 2] \times [-2, 2] \times [4, 8]$. Then, we choose the ground-truth translation $\mathbf{t}_{true}$ such that the origin of the object framework coincides with the centroid of these 3D points, and rotate these 3D points by using a randomly generated ground-truth rotation matrix $\mathbf{R}_{true}$.

To evaluate the numerical stability of the solvers, we randomly generate 50 ordinary 3D points and simulate their noise-free projections for fully-random, near-Cayley-degenerate ($a \approx 0$) and Cayley-degenerate rotations ($a = 0$), respectively.

We can see that the symmetric solver is superior to the general (blind) GB solver that does not utilize the symmetry in all different configurations. The stability of the DLS solver [45] is superior to the OPnP formulation in general random configuration. On the other hand, the accuracy of the OPnP solver can be improve drastically with even one step of damped Newton refinement (blue line). As degenerated configurations for Cayley's representation, the OPnP solver consistently better than the DLS solver which suffers from such degeneracy. As for computation efficiency, given that the degenerated Cayley parameterization in DLS solver yields a simpler polynomial system (3 variables and 3rd degree equations), it can be solved very efficiently with resultant method (5.8 ms). Besides the numerical stability, we can see that the solver utilizing symmetric is much faster than the general GB solver (18.5 ms compared to 37.2 ms on average).

Param.	Solution	Technique	Template	Time (ms)
OPnP	81	GB (general)	576×656	37.2
OPnP	81	GB (Symmetry)	348×376	18.5
DLS [45]	27	Resultant	120×120	5.8

Table 4.1: Average time performance of different solvers for Perspective-n-Point problem. The timing are measured on a MacBook Air with 1.8 Ghz i5 CPU.

Optimal Euclidean Registration For this experiment, we simulate point-to-plane correspondence in 3D randomly. We first study the numerical stability of the general solver generated by [64] and our solver that utilizes partial symmetry. In Figure 4.2, we can see that our method is superior to the general solver with respect to the stability. On the other hand, our solver is around three times faster than the general solver (Table 4.2). The formulation in [80] is a quadratic programming problem and a branch-and-bound scheme (bnb) for the global optima was derived. It is much more difficult problem to solve and is in general very slow with increasing number of points. Our formulation along with our solver is much fast and guarantee to find the global optimal of the same geometric error.

bnb* [80]	General [64]	Symmetric
> 1s	75ms	23ms

Table 4.2: Average time performance of different solvers for the optimal Euclidean registration problem. ($*$) based on the time reported in [80]. The other timing are measured on a MacBook Air with 1.8 Ghz i5 CPU.

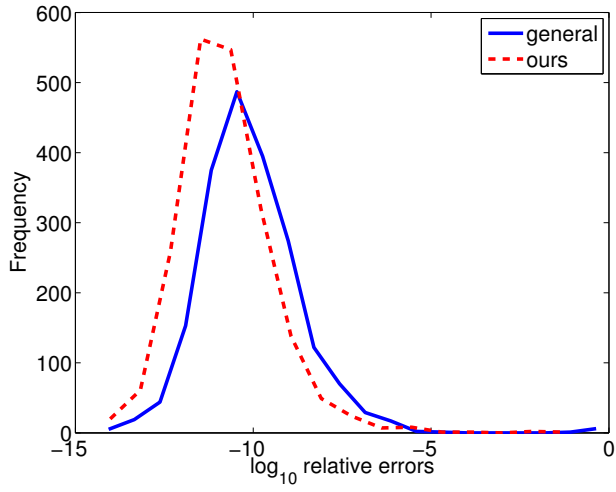
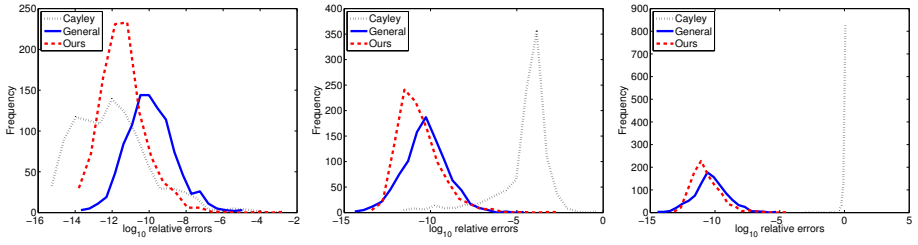


Figure 4.2: Numerical stability of the general polynomial solver [64] and our symmetric solver for the optimal Euclidean registration problem. The histogram of log_{10} relative errors of unit quaternion for 1000 noise-free random problems is shown.

PnL Problem For the synthetic experiments in this section, we first randomly generate 3D lines at around the origin, with cameras pointing towards the origin approximately. Then we calculate 2D projections of the lines onto the image plane. We perturb endpoints of lines to simulate noise. We will first look at the numerically stability of the proposed solvers under different configurations. In Figure 4.3, we first observe that our partially symmetric solver is better than the general solvers across different experiments. Our solver is also faster (23ms) compared to the general solver (55ms) which has a larger elimination template. On the other hand, for random configurations, we can see that the solver based on Cayley parameterization of the quaternion [76] performs better than both the general solver and the partially symmetric solver with unit-norm constraints. However, when the configuration is close to Cayley-degeneracy, the performance of such solve deteriorate drastically (Figure 4.3, mid). And for degenerated cases (e.g. $a = 0$), the solver in [76] fails completely.

It is also of interest to study the performance of the solvers under noise (Figure 4.4). In this experiment, we generate camera pose in fully-random manner. With formulation presented here, we can see that the solvers are much prone to degeneracy which causes the large variance in mean errors for Cayley-based method. The RPnL method described in [103] yields better results than Cayley-based method, but still inferior to the formulation here. With the same formulation, our solver performs sim-



(a) Random (b) Near degenerate (c) Degenerate

Figure 4.3: Numerical stability of the Cayley PnL solver [76], general polynomial solver [64] and our symmetric solver under varying configurations. The histogram of log_{10} relative errors of unit quaternion parameterization for 2000 noise-free random problems is shown.

ilarly to the general solver under varying noise level and number of lines, while being much faster.

4.6 Conclusions

We present a general framework for utilizing symmetry in solving polynomial systems. We study and prove the correspondence between symmetric polynomial systems and symmetric solution sets. We have also identified several examples in computer vision that are fully symmetric or partially symmetric. We verify the advantages of utilizing symmetry in improving both speed and numerical stability of polynomial solvers.

As future work, it is of practical importance to achieve automatic detection and reformation of symmetric polynomial systems. While the techniques presented in this chapter can be combined with previous optimization schemes for polynomial solvers, it is of interest to see whether specific optimization scheme can be derived for symmetric systems. Moreover, it is important to derive schemes for automatically selecting mapping for solution extraction in a numerically stable manner. Another direction is to explore other types of symmetries in polynomial systems.

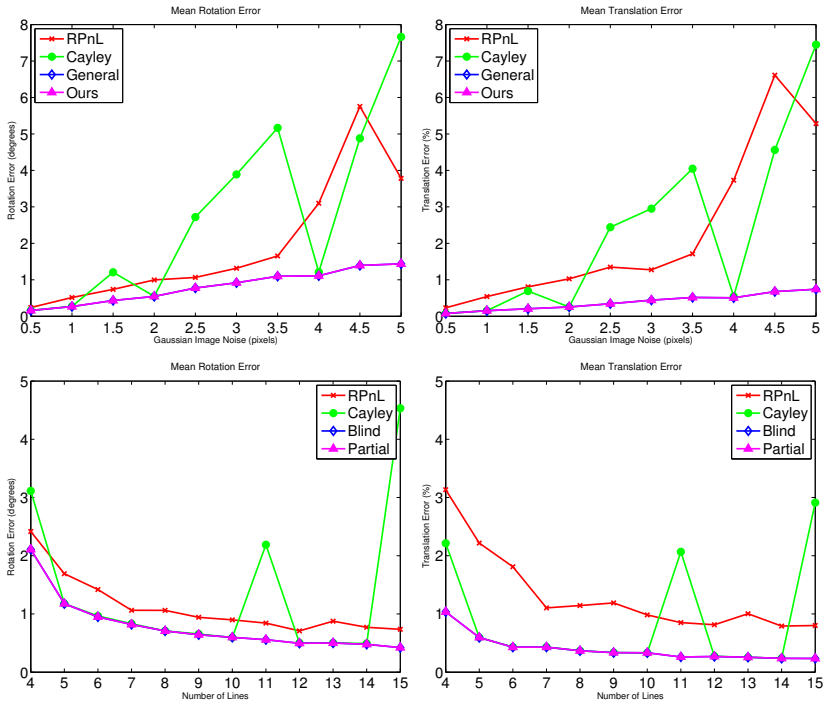


Figure 4.4: Top: noise sensitivity of different PnL solvers with varying noise levels for fixed n = 10 lines. Bottom: noise sensitivity of different PnL solvers with varying number of lines with fixed noise level (2 pixels).

Part II

Geometric Problems in Computer Vision

Chapter 5
Pose Estimation

In this chapter, we study the camera pose estimation problem. Specifically, we study the geometric problems of estimating camera pose with unknown focal length using combination of geometric primitives. We consider points, lines and quivers, i.e. points with one or more directions. We identify several minimal cases and formulate minimal cases as polynomial systems where the constraints for different primitives are handled in a unified way. We develop efficient polynomial solvers for each of the derived cases. The availability of these solvers enables robust pose estimation with unknown focal length for wider classes of features. Such rich features also allow for fewer feature correspondences and generate larger inlier sets with higher probability.

5.1 Pose Estimation with Unknown Focal

To be able to estimate camera pose given 2D-3D correspondences of geometric primitives e.g. points, lines etc. is of great interest to applications like vision-based localization [46]. In typical scenarios, focal length of the camera is the only unknown that is most difficult to determine accurately (exif-tag could provide erroneous estimate) and can render large errors in the pose estimation. All previous methods for pose estimation with unknown focal length use point correspondences. The contribution of this chapter is to enable a wider class of geometric features (combinations of points, lines and n-quivers, Figure 5.1) for simultaneous pose estimation and focal length calibration.

Related Works The problem of camera pose estimation has been studied extensively in the computer vision community. The minimal case of pose estimation using 3 points was studied in [35] and several other formulations are compared and reviewed in [40]. For line-to-line correspondences, solutions are derived for minimal of 3 lines in [25, 32]. Recently, the minimal cases using combination of points and lines are solved in [89]. In [34] a solver is derived for a minimal problem of 2 points and their corresponding tangent directions (equivalently any direction vector through each of the points). The required correspondence is reduce to a single local patch corre-

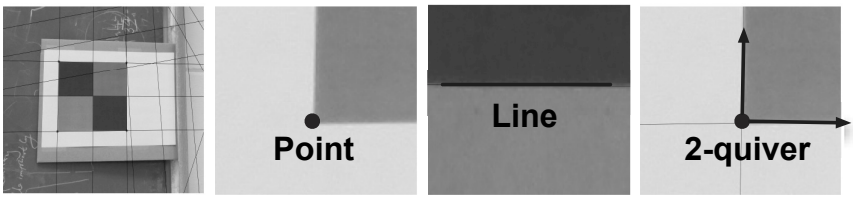


Figure 5.1: The figure illustrates three examples of image features in an image, a point with 2 degrees of freedom, a line with 2 degrees of freedom and a 2-quiver with 4 degrees of freedom. The 2-quiver consists of a point and two directions out from the point.

spondence in [52] . However, this specific setting is unfortunately very sensitive to measurement noise of the patches.

For camera pose estimation with unknown focal length, the planar case was studied and solved in [1]. For general non-planar cases, the close to minimal case using 4 2D-3D correspondences was first studied in [96]. Efficient and numerically stable solvers are developed in [12]. By combining 2D-2D and 2D-3D correspondences, [51] investigated several minimal cases for pose estimation with unknown focal length. Additionally, for camera with unknown radial distortion and unknown focal length, the 4-point minimal case was solved in [14, 50]. By using 5 point correspondences, a fast but overdetermined solver with real-time RANSAC performance was proposed in [67]

Many other works focus on solving the overdetermined problem of estimating camera pose with more than three points [45, 70, 76] or lines [76]. Very recently, the approach in [70] was extended to handle unknown focal length [82]. All of these methods are based on a formulation that minimizes certain algebraic errors and generally assume that there exist no outliers in the data. Minimal solvers are the key components of the preprocessing steps for such overconstrained solvers to robustly remove outliers.

5.2 Problem Formulation

We start with the pinhole camera model. The projection equations for a 3D point $\mathbf{X}$ and its corresponding 2D image projections $\mathbf{x}$ is,

$$\eta \mathbf{x} = \mathbf{P}\mathbf{X}, \tag{5.1}$$

where η is the depth of $\mathbf{X}$ and the projection matrix can be decomposed as $\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}$. In practical camera setups, it is generally assumed that the cameras have centered principle points, square pixels with zero skew. We thereafter assume that the

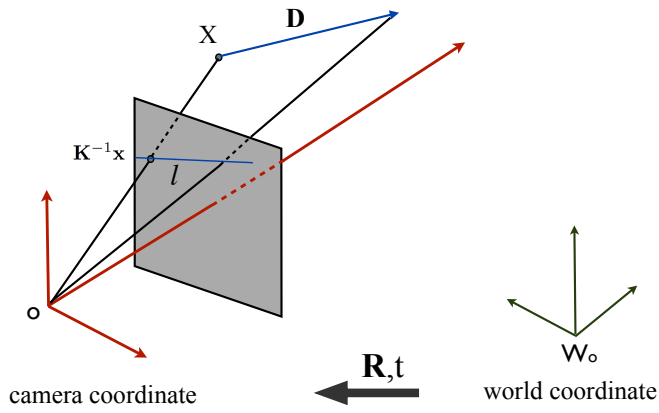


Figure 5.2: Camera coordinate, world coordinate, and the geometric relations between point, line and direction correspondences.

calibration matrix $\mathbf{K}$ only involves the unknown focal length f. Based on (2.3), the matrix $\mathbf{K}$ can be written equivalently as

$$\mathbf{K} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & w \end{bmatrix}, \tag{5.2}$$

where $w = 1/f$. Therefore, the problem of determining camera pose with unknown focal length has in total 7 degrees of freedom (3 in rotation $\mathbf{R}$, 3 in translation $\mathbf{t}$ and 1 in f).

5.2.1 Number of Constraints

In this section, we discuss in details the constraints given by different geometric primitives.

Point Constraints. Given a known 3D point $\mathbf{X}$ and its corresponding image point $\mathbf{x}$, it is well known that there are two constraints on $\mathbf{P}$ [41]. The two constraints can be chosen from the three linearly dependent equations based on (5.1) :

$$[\mathbf{x}]_\times \mathbf{P}\mathbf{X} = 0, \tag{5.3}$$

where

$$[\mathbf{x}]_\times = \begin{bmatrix} 0 & -1 & v \\ u & 0 & -1 \\ -v & u & 0 \end{bmatrix}.$$

Line Constraints. Given a known 3D line L and its corresponding image line $\mathbf{l}$, there are also two constraints on $\mathbf{P}$. If the 3D line L is represented as a 3D point $\mathbf{X}$ and the direction of the line $\mathbf{D}$, one can obtain two equations for the two points in the following form based on (5.1):

$$\begin{aligned} &\mathbf{l}^T\mathbf{P}\mathbf{X} = 0, \\ &\mathbf{l}^T\mathbf{P}(\mathbf{X} + k\mathbf{D}) = 0, \end{aligned} \tag{5.4}$$

where k is an arbitrary constant.

Quiver Constraints. For a known 3D point $\mathbf{X}$ and a directional measurement $\mathbf{D}$ through $\mathbf{X}$, given the corresponding image projection $\mathbf{x}$ and $\mathbf{d}$, there are 3 constraints on $\mathbf{P}$. We hereafter call the geometric primitive with a point and n directions passing through the point as an *n-quiver*. For a 1-quiver, we first obtain two constraints from the point correspondence according to (5.3). The other constraint comes from the directional measurement. To see this, we first convert the measurement $\mathbf{d}$ along with $\mathbf{x}$ to a line measurement $\mathbf{l}$. Then we utilize the equations in the form of (5.4) and take the difference between them. Equivalently, we have

$$\mathbf{l}^T\mathbf{P}\mathbf{D} \quad = 0. \tag{5.5}$$

For a 2-quiver, we have in total four constraints including two point constraints and two constraints in the form of (5.5). In general, there are $n + 2$ constraints for an n-quiver (two constraints from the point and n constraints from the n directional correspondences).

The number of constraints for points, lines and quivers are summarized in Table 5.1.

Point	Line	n-Quive
2	2	n+2

Table 5.1: Number of constraints enforced by 2D-3D correspondences of different geometric primitive for camera pose estimation.

5.2.2 Useful Cases

With 2D-3D correspondences of points, lines and n-quivers, one can form several novel minimal cases by searching for combination such that $2m_p+2m_l+(n+2)m_q = 7$, where m_p, m_l, m_q are the number of point, line, n-quiver correspondences, respectively. We present and solve two of such minimal cases and also study a slightly over-determined cases using 4 lines.

Two Points and One 1-Quiver ($\mathbf{P2Q1}$) : Given three points and one direction passing through one of the points, we can form 6 equations based on (5.3) and 1 equation based on (5.5). Thus this problem is minimal.

One 1-Quiver and One 2-Quiver ($\mathbf{Q1Q2}$) : For two points, where one line passing through one point, and two lines passing through the other point are known, we can form 4 point equations (5.3) and 3 equations with respect to the directions (5.5). This yields also a minimal problem.

Four Lines ($\mathbf{P4L}$) : Given 4 3D-2D line correspondences, there are in general 8 independent constraints. Thus, the problem of camera pose with unknown focal length is over-determined with 4 lines. We can choose 7 from the 8 equations, and use the eighth equation to verify a unique solution.

In a similar manner, other minimal cases include the setups: (i) one point, one line and one 1-quiver (ii) two lines and one 1-quiver which can be solved in similar manner as the presented solvers.

5.2.3 Parameterization

There are many ways to parameterize the problems related to camera pose estimation. In [96], Triggs first parameterizes the camera as an arbitrary matrix with 12 unknowns, the solutions then lie in the null space of the linear constraints given by the point constraints. Then the quadratic constraints (orthogonality and equal norm) on the rotational part of the camera matrix are enforced afterwards. The benefits of this formulation is that one needs to solve only quadratic polynomial systems. Once the rotational part is recovered, the focal length can easily be calculated using the ratios between the norms of the third and the first two rows of $\mathbf{R}$. The drawback of this formulation is that non-planar and planar scenes need to be handled separately and explicitly as shown also in [14].

On the other hand, Bujnak et al. [12] formulate the P4P problem with unknown focal length using the invariance of the ratios of distances between the 3D points under rigid transformation. For directional correspondences, one can similarly make use of the invariance of the angles between the directions [34]. Here, we discuss briefly the application of such geometric invariance to the $\mathbf{P2Q1}$ problem i.e. two points ($\mathbf{X}_1, \mathbf{X}_2$) and one point ($\mathbf{X}_3$) with a known direction ($\mathbf{D}$). To start with, we can use the three points to form 2 independent distance ratio equations involving three unknowns (two relative stretch ratios α_1,α_2 and f) as in [12]. Then for the known direction, one can

form equations using the invariance of angles for $(\mathbf{D}, \mathbf{X}_3 - \mathbf{X}_1)$ and $(\mathbf{D}, \mathbf{X}_3 - \mathbf{X}_2)$. This again produces two independent equations involving all 4 unknowns (α_1, α_2, α_3, f). Thus, we obtain 4 equations with 4 unknowns. However, the resulting equations consists at least one equation of degree 6 (after substitution and simplification), which makes the resulting polynomial system very difficult to solve. While the use of geometric invariance might yield polynomial system with fewer solutions for the $\mathbf{P}4\mathbf{P}$ problem with unknown focal length, it is not straightforward to see that such property is preserved for other primitives like directions with unknown focal length.

Here, we choose to parameterize the rotation matrix $\mathbf{R}$ with quaternion and construct equations directly based on (5.3), (5.4) and (5.5). It turns out that this straightforward parameterization, produces polynomial systems that are relatively easy to solve and general for both planar and non-planar scenes. In the rest of discussions, the rotation matrix $\mathbf{R}$ is parameterized with quaternion as

$$\begin{bmatrix} a^2+b^2-c^2-d^2 & 2bc-2ad & 2ac+2bd \\ 2ad+2bc & a^2-b^2+c^2-d^2 & 2cd-2ab \\ 2bd-2ac & 2ab+2cd & a^2-b^2-c^2+d^2 \end{bmatrix}. \tag{5.6}$$

To fix the scale of the quaternion, we can choose Cayley's representation by setting $a = 1$. By setting $a = 1$, we reduce the number of unknowns which eases the polynomial system solving. As discussed in Chapter 4, this will in general introduces degenerated rotations ($a = 0$) or potential numerical instability ($a \approx 0$). To avoid this degeneracy, one can alternatively fix the scale with the unit-norm constraint $a^2 + b^2 + c^2 + d^2 = 1$. Then the problem is partially symmetric to $\{a, b, c, d\}$ of type-2 and the techniques in Chapter 4 can be applied. Here, we present the solvers with Cayley's parameterization. Due to the rare occurrences of degenerate configurations, we will demonstrate in the experimental section such degeneracy does not affect the practical usage of the solvers.

From the factorization in (2.2), we know that $\mathbf{P} = \mathbf{K}[\mathbf{R}\ \ \mathbf{t}]$ can be rewritten as

$$\begin{bmatrix} a^2+b^2-c^2-d^2 & 2bc-2ad & 2ac+2bd & t_x \\ 2ad+2bc & a^2-b^2+c^2-d^2 & 2cd-2ab & t_y \\ w(2bd-2ac) & w(2ab+2cd) & w(a^2-b^2-c^2+d^2) & wt_z \end{bmatrix}.$$

where $\mathbf{t} = [t_x\ \ t_y\ \ t_z]^T$. If we additionally set $t'_z = wt_z$, we have in total 7 unknowns $\{b, c, d, t_x, t_y, t'_z, w\}$. Given different geometric primitives, the constraints (5.3), (5.4) and (5.5) are linear to $\{t_x, t_y, t'_z\}$. Thus, we can conveniently eliminate all three of them and rewrite the equations with respect to the 4 unknowns $\{b, c, d, w\}$ only. Specifically, for all the useful cases presented in Section 2.2, we can choose 3 of the equations to eliminate $\{t_x, t_y, t'_z\}$ and obtain 4 cubic equations with 4 unknowns (for $\mathbf{P}4\mathbf{L}$, there are 5 such cubic equations). In the next section, we will discuss the solutions to these polynomial systems.

5.3 Polynomial Solvers

We utilize the techniques based on Gröbner basis to solve the resulting polynomial systems. Instead of using the automatic solver generator [64], we choose to use the techniques in [20] with column pivoting basis selection for better numerical stability. For polynomial systems with small number of unknowns, Gröbner basis methods are generally fast and numerically stable.

We start by verifying the number of solutions. For instance, for minimal problem of two points and one 1-quiver ($\mathbf{P}2\mathbf{Q}1$), we verify using algebraic geometry tools Macaulay2 [38] in $\mathbb{Z}_p$ that there are in general 20 solutions. Recall from Section 2.3 that, after linear elimination, we are left with 4 equations with 4 unknowns $\{b, c, d, w\}$. To solve the polynomial system, we first multiply the 4 equations with all the monomials of total degree up to 6 and maximum degree of each variable as $[2, 2, 2, 3]$, respectively. In this way, we obtain an elimination template of 372 equations and 386 monomials. To enhance numerical stability, we employ the basis selection technique by choosing the permissible set to be the last 35 monomials in *grevlex* ordering. After the QR factorization with column pivoting, we can construct the action matrix of size 20×20 from which the solutions can be obtained by eigenvalue decomposition. After we solve for $\{b, c, d, w\}$, we can calculate the values of other unknowns using linear substitution.

For the other two minimal cases $\mathbf{Q1Q2}$ and $\mathbf{P}4\mathbf{L}$, we find that the number of solutions to the corresponding polynomial systems is also 20 and the same elimination template and choice of permissible monomials give very similar numerical stability. This could be due to the similar structures in the constraints of these problems.

5.4 Experiments

In this section, we study the performance of polynomial solvers on both synthetic and real data.

5.4.1 Synthetic Data

For the synthetic experiments, we choose the size of image to be 1024×800. Random scenes were generated by drawing points uniformly from a cube with side length 800 centered at the origin. Then the directions through points were chosen randomly (either in planar or in non-planar fashion). A camera was placed at a distance of around 1000 from the origin, pointing approximately at the center. The camera was calibrated except for the focal length.

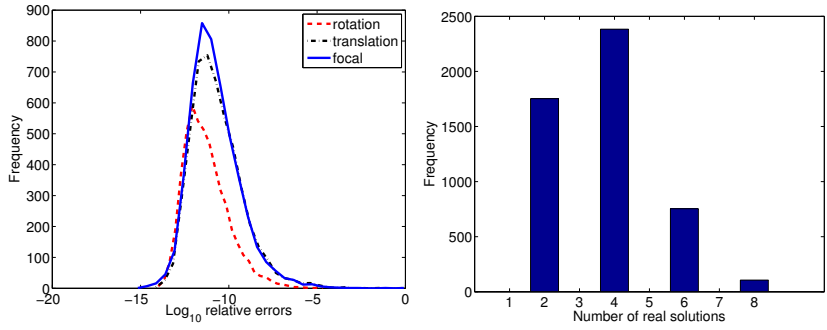


Figure 5.3: Two point and one 1-quiver ($\mathbf{P}2\mathbf{Q}1$), synthetic experiments for 5000 random problems on noise-free data with focal length set to approximately 1000. **Left**: Histogram of relative errors for rotation, translation, focal length; **Right**: Histogram of number solutions with real and positive focal lengths.

Stability and Number of Solutions

We evaluate first the solvers on noise-free data to check the numerical stability of the solvers and distribution of number of valid solutions. For $\mathbf{P}2\mathbf{Q}1$, in Figure 6.2 (left), we can see that the numerical errors are fairly low for most of the random cases. We note that the solutions to w are coupled i.e. if w is solution, so is $-f$, which also corresponds to equivalent pairs of camera matrices $\mathbf{P}$ and $-\mathbf{P}$[1]. Since only the real and positive f are geometrically valid, one can safely remove the other solutions. In the simulation, it is shown that there are up to 8 solutions with real and positive f while in most of the cases only 2 or 4 solutions (Figure 6.2, right).

The boxplot in Figure 5.4 shows the medians, 25 percentiles and 75 percentiles of the distribution of the relative errors. We can see that for noise-free data, the Gröbner basis solver for ($\mathbf{P}2\mathbf{Q}1$) is consistently stable for different focal lengths for both planar and non-planar scenes (Figure 5.4). Similar numerical behaviors are observed for the solver using lines ($\mathbf{P}4\mathbf{L}$, Figure 5.5). Given that the performance of other solvers are similar, related figures are not shown individually here.

The solvers implemented in MATLAB take approximately $15ms$. The computation is dominated by the first elimination using QR factorization. For comparison, the optimized $\mathbf{P}4\mathbf{P}$ solver in [12] runs at around $2ms$. Our solvers can also be further optimized for speed using strategies in [54, 64]. The time performance is measured on a Macbook Air with 1.8 GHz Intel Core i5 and 8 GB memory.

[1]This seems to indicate the partial symmetry of type-2 discussed in Chapter 4. In fact, there exists no symmetry for the variable w in the monomials. We conjecture that such symmetry is either induced by the coefficients or another kind of symmetry in the underlying system.

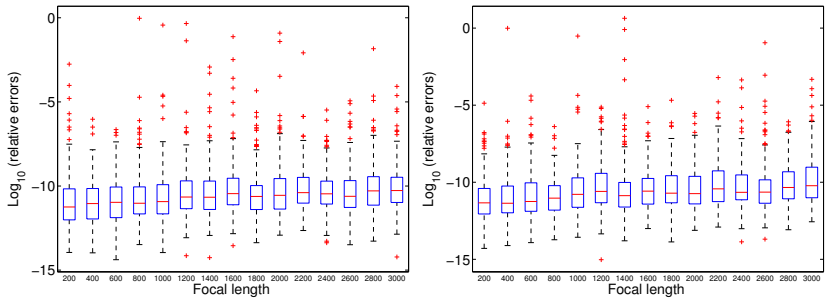


Figure 5.4: Synthetic experiments of **P**2**Q**1 on noise-free data with varying focal lengths. **Left**: Boxplot of relative errors of focal lengths for non-planar points and directions; **Right**: planar cases.

Noise Sensitivity

To study the behaviors of the solvers with noisy measurements, we add noise of different levels both to the image point positions and the angles of the directions. In Figure 5.6, it is shown that the **P**2**Q**1 solver gives fairly good estimates for focal lengths with small noise, and is still able to provide (though not as frequently) reasonably good initial solutions when the noise is around 5 pixels. We have also noticed that the solvers can be sensitive to errors in the direction measurements. We also test the **P**4**L** solvers for noisy line measurements by perturbing the intersections between the lines and the x, y axis. From Figure 5.7, we can see that the **P**4**L** solver is capable of recovering the focal length accurately for small perturbation and can become unreliable for large perturbation. To further understand the noise sensitivity, we demonstrate the performance of the solvers on real image measurements in Section 5.4.2.

RANSAC Experiments

To test the advantage of the proposed solvers for different geometric primitives, we simulate data with outliers and RANSAC is used to obtain robust initial solution. For a fixed camera with focal length 1000, we generate randomly 1000 scene points as in the previous section, directions through points are also generated randomly. Then both the image point positions and projected directions are perturbed with random noise. A subset of the points (30%) are chosen as outliers with large perturbations on both the positions and angles of the directions. We compare the solvers for two points and one 1-quiver (**P**2**Q**1) and one 1-quiver and one 2-quiver (**Q**1**Q**2) with the **P**4**P** solver in [12]. For each of the solvers, we choose the minimal set of data required for RANSAC, the distribution of the ratio of inliers of each RANSAC loop in shown in Figure 5.8. Here we define the inliers as the image points with reprojection errors less

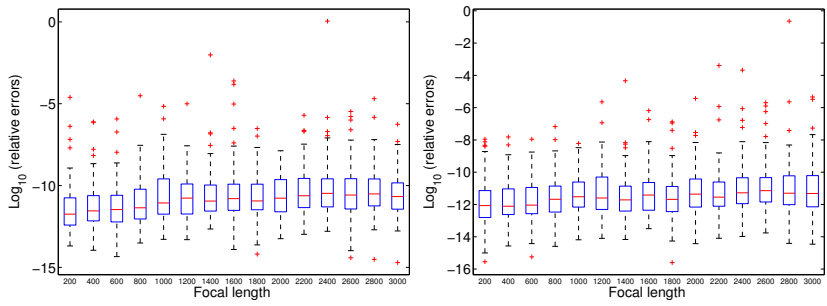


Figure 5.5: Synthetic experiment of **P**4**L** on noise-free data with varying focal lengths. **Left**: Boxplot of the relative errors of focal lengths for non-planar line configurations **Right**: planar cases.

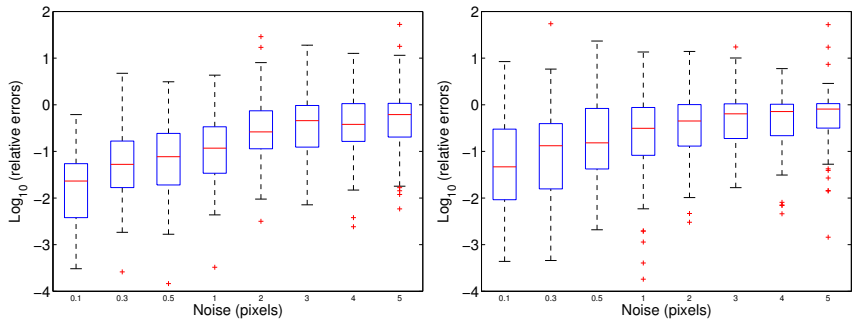


Figure 5.6: Synthetic experiments for **P**2**Q**1 on noisy data with varying noise levels on image point positions with fixed $f = 1000$ and angle perturbation of degree $[-0.1, 0.1]$. **Left**: Relative errors of focal lengths for non-planar points and directions; **Right**: planar cases.

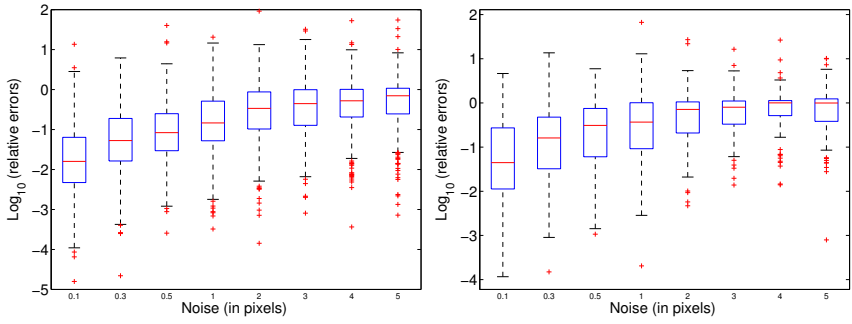


Figure 5.7: Synthetic experiments for **P**4**L** with varying noise on the intersection points of between the lines and x, y axes with fixed $f = 1000$. **Left**: Relative errors of focal lengths for non-planar lines; **Right**: planar cases.

than a predefined threshold. It is not surprising to see that the **Q**2**Q**1 solver performs the best with respect to recovering inliers since it only requires two points. While (**P**2**Q**1) performs slightly worse, it still gives better results than the **P**4**P** solver which needs at least 4 point correspondences.

5.4.2 Real Data

We took 16 images of seven cardboards placed in a non-planar configuration with varying focal lengths (Figure 5.9), using a standard Canon EOS 50D camera. Each cardboard is attached with a pattern with dark and light squares for the ease of line detection. The automatic line detection algorithm detected 6 lines for each of the card board, and 9 points as the intersections of those lines. Thus, we have in total 63 points, 42 lines and 63 2-quivers.

We used these images to verify the applicability of the proposed solvers on real images with point, line and quiver features. The lines were estimated by sub-pixel edge-detection, cf. [4]. This makes it possible to both estimate edge positions and edge position uncertainty. Lines as well as the uncertainty in their parameters were then obtained by fitting to these data. Finally points and their uncertainty were estimated by intersection of two or more such lines. For 16 images, there are in total 621 visible measurements of the points (2-quivers) and 456 measurements of lines. The output is thus a number of image points, image lines, and image quivers as illustrated in Figure 5.1. Given correspondences of these image features, we reconstructed both scene structure, camera motion as well as the focal length. The resulting reconstruction of the 3D points and the camera poses as well as the focal lengths after bundle adjustment are fairly accurate and thus serves as ground truth. In the bundle adjustment we have also used the estimated uncertainties in the image features.

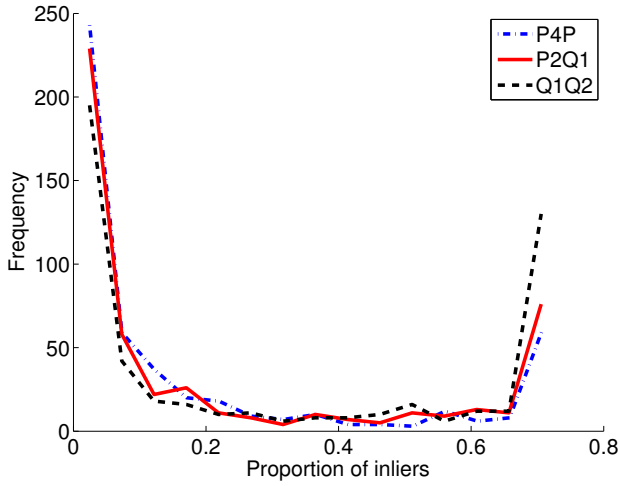


Figure 5.8: Distribution of inlier proportions for 1000 RANSAC runs for different solvers **P**4**P**, **P**2**Q**1 and **Q**1**Q**2.

Given the correspondences between the image features and 3D reconstruction, we use the proposed solvers to estimate both the camera pose and the focal length for each of the image. The estimations are then compared with the results given by the reconstruction. Due to the high quality of the reconstruction, the data can be seen as outlier-free. We first look at the reprojection errors for the camera poses and focal lengths estimated using different solvers and investigate whether the solvers adapt to real image noisy measurements. To measure the reprojection errors, we run different solvers in a RANSAC manner by choosing random minimal measurements. The average reprojection errors of image points for different solvers are reported in Table 5.2. We can see from Table 5.2 that the errors of all our proposed solvers are similar to the **P**4**P** solver.

	P4**P**	**P**2**Q**1	**Q**1**Q**2	**P**4**L**
Errors	2.463	2.531	3.123	3.141

Table 5.2: Average reprojection errors (in pixels) of image points with camera poses and focal lengths of the 16 images estimated with different solvers.

To further test the performance of the solvers, we also generate outliers by adding large synthetic perturbations to a random subset (30%) of image point positions,

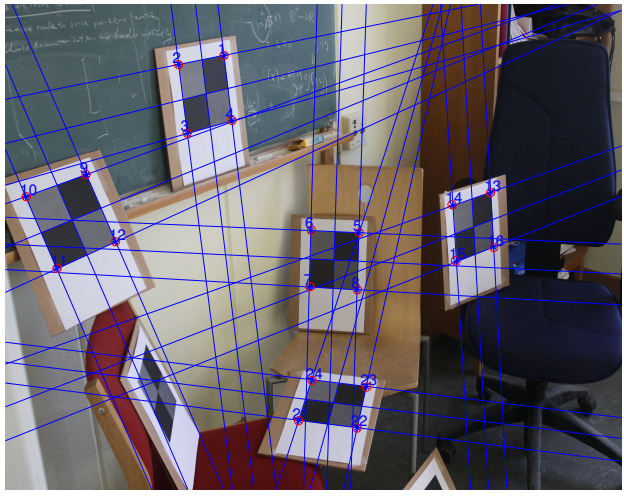


Figure 5.9: One of the Images of cardboards with detected lines and points.

quiver directions and lines. We then run RANSAC (1000 runs for each image) on the perturbed data. For the inlier threshold of 3 pixels, the number of inliers (among in total 621 measurements) and the average reprojection errors for inliers are reported in Table 5.3. For this specific example, **P**4**P** and **P**2**Q**1 output higher count of inliers and in the meantime give lower average reprojection errors. The slightly inferior performance of **Q**1**Q**2 and **P**4**L** solvers might be due to the sensitivity of these solvers to measurement errors in the directions of quivers or lines.

	P4**P**	**P**2**Q**1	**Q**1**Q**2	**P**4**L**
Inliers	309	298	253	223
Errors	1.502	1.330	1.402	1.633

Table 5.3: Number of inliers and average reprojection errors (in pixels) of inliers with 30% synthetic outliers for the cardboard dataset.

To evaluate the accuracy of the solvers, we compare the best focal length estimated (the one with maximum number of inliers) for each solver against the output from bundle adjustment as well as those extracted from EXIF-tag (conversion from 35mm film equivalent). We set the inlier threshold to be 2 pixels and run RANSAC on the original data without synthetic perturbation. The statistics of the estimated focal lengths are shown in Figure 5.10. It is noted that the focal lengths given by the exif

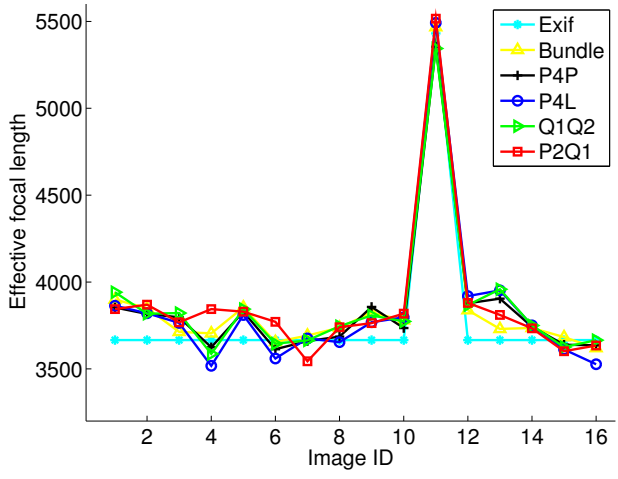


Figure 5.10: Statistics of focal length estimation of different solvers, bundle adjustment and exif-tag for the Cardboard dataset.

information seems to be very coarse compared to those estimates from image data directly. We can also see that all solvers give fairly similar estimates to the results from bundle adjustment.

5.5 Discussions

For the calibrated pose estimation problem where focal length is known, we see the potential of combining the simplicity the quaternion parameterization and the stability of Gröbner basis solvers. In [34], the minimal case of equivalently two 1-quivers (the direction is detected as the tangent to curves instead of arbitrary direction) for pose estimation was studied. A closed form solution for a polynomial equation of degree 16 was derived through rather involved calculation. With the quaternion formulation, we directly arrive at 3 quadratic equations on 3 unknowns $\{b, c, d\}$ which is extremely fast to solve using Gröbner basis solver (approximately 1ms) compared to a few milliseconds of the released implementation for [34]. Though it is not fair to compare the time performance for unoptimized codes (both of them), it could still suggest superiority of the easy formulation and implementation of the Gröbner basis based solvers.

5.6 Conclusions

In this chapter, we present several novel cases for pose estimation with unknown focal length with combinations of points, lines and quivers. Pose estimation from mixtures of features allow for fewer feature correspondences and generate larger inlier sets with higher probability.

One of the key directions in research is to evaluate the application of new solvers to discriminative feature like SIFT to ease the correspondence problem for edges (direction of a quiver and line). One potential way is to make use of the dominant gradient directions given by SIFT and treat them as quiver directions. In this way, the correspondence problem is made relatively easier. In this case, one need to verify whether the solvers are robust against noisy estimation of the gradient directions. To improve the speed and numerical stability of the solvers, it is of interest to resolve the intrinsic symmetry in the quaternion parameterization either by algebraic manipulation or by deriving alternative set of constraints using geometric invariances.

Chapter 6
Radial Distortion

In this chapter, we study the problems of estimating relative pose between two cameras in the presence of radial distortion. Specifically, we consider minimal problems where one of the cameras has no or known radial distortion. There are three useful and previously unsolved cases for this setup with a single unknown radial distortion (i) fundamental matrix estimation where the two cameras are uncalibrated, (ii) essential matrix estimation for a partially calibrated camera pair, (iii) essential matrix estimation for one calibrated camera and one camera with unknown focal length. We study the parameterization of these three problems and derive fast and stable polynomial solvers.

6.1 Single Unknown Radial

Taking radial distortion into account is important in pose estimation and structure from motion problems. Problems for relative poses with unknown radial distortion have been studied extensively in the past [5, 36, 42, 65, 66, 68]. For all these cases, either constant or varying radial distortion is assumed on the cameras. Such settings are useful for self-calibrating cameras with no prior knowledge on the cameras' intrinsic parameters. Estimating relative pose and the radial distortion simultaneously is achieved by solving either a linear system or a polynomial system. Due to the difficulty of these problems, they either require many point correspondences or dedicated polynomial solvers. In fact, some of the important minimal problems have still not been solved due to their difficulty, like estimating the relative pose with unknown focal length and radial distortion (both constant). In a general structure from motion pipeline, however, it is common that certain *seed* images have pre-calibrated or already known radial distortion. For these cases, the prior on the calibration of one of the cameras can reduce the complexity of the problem. In [13], a similar strategy has been applied in focal length estimation where the focal length of one camera is assumed to be known. This motivates us to study the problem of relative pose estimation with a single unknown radial distortion (sometimes referred to as *one-sided* radial distortion problem). We are particularly interested in solving the related mini-

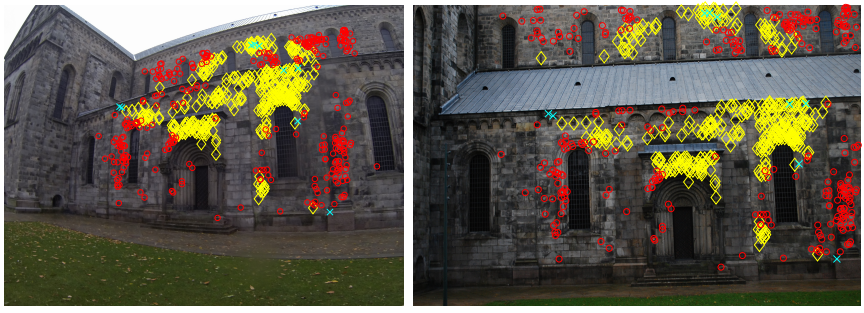


Figure 6.1: Example of modeling one-sided radial distortion for estimating relative pose. Left: an image taken with an GoPro-Hero3 with unknown radial distortion. Right: an image taken with a Nikon D60 camera which has known radial distortion. Yellow diamonds are the common inliers for fundamental matrix estimation obtained by both the standard 7-point method [41] and our minimal one-sided 8-point method. By considering radial distortion, our method obtains many extra inliers (red circles) while misses only a few inliers (cyan crosses) compared to the 7-point method.

mal problems for robust estimation purposes.

Related Works To model radial distortion, we have followed the one-parameter division model proposed by Fitzgibbon [36] and assumed that the radial distortion center is known. For fundamental matrix and constant radial distortion on both cameras, Fitzgibbon proposed an overdetermined 9-point solver which involves solving a quadratic eigenvalue problem that has two solutions. The minimal case of this problem was solved in [65], and has in general 12 solutions. For a fundamental matrix with varying radial distortion, a linear solution using 13 point correspondences was proposed in [5]. An efficient and stable 9-point minimal solver was later derived in [68] with 24 solutions. For an essential matrix with constant radial distortion, which requires minimally 6 point correspondences, there are in general 56 solutions. It was also solved in [68]. The most related work is [11], which considered the one-sided fundamental matrix estimation problem. Instead of solving the minimal case with 8 point correspondences, 9 point correspondences were used to simplify the problem into solving a cubic equation. While obtaining a simple overdetermined solver is useful, the study of the minimal cases is of both of theoretical interest and of great practical importance in robust estimation, for example using RANSAC [35]. Non-iterative solvers for estimating the radial distortion center (both one-sided and two-sided cases) were also studied in [9, 10, 71]. Even though the additional modeling of radial distortion center is useful, it has been shown that the estimation can be very sensitive to

noise [10].

Contributions In this chapter, we study the three unsolved minimal cases for relative pose estimation with a single unknown radial distortion: (i) 8-point fundamental matrix and radial distortion, (ii) 7-point essential matrix, focal length and radial distortion, and (iii) 6 point essential matrix and radial distortion. For each of these cases, we derive a parameterization and a linear elimination scheme to simplify the polynomial systems. We study the polynomial systems and verify the number of solutions. We then develop fast and stable polynomial solvers for all these minimal cases. These solvers are minimal, fast and more accurate than previous non-minimal solvers, for example [11]. The availability of these fast solvers enables the possibility of initializing a large-scale structure from motion pipeline with radial distortion taken into account. We demonstrate the benefits of modeling radial distortion on real images by using our proposed solvers.

6.2 Problem Formulation

We use the pinhole camera model and assume a one-parameter division model for radial distortion as in [36]. Under radial distortion, the relation between undistorted point coordinates $\mathbf{p}_u = \begin{bmatrix} x_u & y_u & 1 \end{bmatrix}^T$ and radially distorted point coordinates $\mathbf{p}_d = \begin{bmatrix} x_d & y_d & 1 \end{bmatrix}^T$ can be written as

$$\begin{bmatrix} x_u \\ y_u \\ 1 \end{bmatrix} \sim \begin{bmatrix} x_d \\ y_d \\ 1 + \lambda r_d^2 \end{bmatrix}, \tag{6.1}$$

where λ is the distortion coefficient and r_d is the distance of $\mathbf{p}_d$ to the distortion center. Here we assume that the distortion center is known and at the center of the image. We further assume that the cameras have centered principal points and square pixels. Therefore, the two unknown calibration parameters we consider here are focal length f and radial distortion λ. The calibration matrix $\mathbf{K}$ can be expressed as

$$\mathbf{K} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & w \end{bmatrix}, \tag{6.2}$$

where $w = 1/f$. Then we have the undistorted image point coordinates

$$\mathbf{p}_n \sim \mathbf{K}\mathbf{p}_u. \tag{6.3}$$

In two-view geometry, it is well known that for uncalibrated cameras, the fundamental matrix $\mathbf{F}$ has 7 degrees of freedom, and for calibrated cameras, the essential

matrix $\mathbf{E}$ has 5 degrees of freedom [41]. On the other hand, we know that each image point correspondence gives one constraint. We consider here the *one-sided* case where calibration matrix $\mathbf{K}'$ and radial distortion λ' is fully known for the second camera. Note that the prior knowledge on the calibration of the second camera does not reduce the degrees of freedom of the epipolar geometry. In Table 6.1, we summarize the related minimal problems on relative pose with radial distortion for both two-sided as well as the one-sided cases. We also show the number of solutions to these minimal problems in general.

Points	Case	Two-sided	One-sided
9	$\mathbf{F} + \lambda + \lambda'$	24 [68]	-
8	$\mathbf{F} + \lambda$	16 [65]	8 (this chapter)
7	$\mathbf{E} + \lambda + f$	68^*	19 (this chapter)
6	$\mathbf{E} + \lambda$	52 [68]	26 (this chapter)

Table 6.1: Minimal problems and corresponding number of solutions to relative pose with unknown radial distortion on both cameras (two-sided) and a single camera (one-sided). $(*)$ To the best of our knowledge, the minimal problem for two-sided $\mathbf{E} + \lambda + f$ has not been solved before. The analysis on number of solutions is based on Macaulay2 [33].

In the following sections, we describe the geometric constraints for these problems including the epipolar constraints from the point correspondences and the intrinsic constraints on the fundamental matrix and the essential matrix.

6.2.1 Fundamental Matrix and Radial Distortion

Given n point correspondences, the epipolar constraints are

$$\mathbf{p}_{n_i}^T(\lambda)\mathbf{F}\mathbf{p}'_{n_i} = 0, \quad i = 1, \ldots, n. \tag{6.4}$$

where $\mathbf{p}_n$ and $\mathbf{p}'_n$ are corresponding undistorted image points in the first and the second image, respectively.

The singularity of the fundamental matrix $\mathbf{F}$ is enforced as

$$\det(\mathbf{F}) = 0. \tag{6.5}$$

6.2.2 Essential Matrix and Radial Distortion

Depending on whether the focal length f is known for the first camera, we have two minimal cases as follows.

Radial Distortion Only. If two cameras are calibrated up to an unknown radial distortion on the first camera, given n point correspondences, the epipolar constraints on the essential matrix $\mathbf{E}$ are

$$\mathbf{p}_{u_i}^T(\lambda)\mathbf{E}\mathbf{p}'_{u_i} = 0, \quad i = 1, \ldots, n. \tag{6.6}$$

Radial Distortion and Focal Length. Here we study the case where the first camera is calibrated up to an unknown focal length and an unknown radial distortion. Based on (6.1) and the calibration matrix $\mathbf{K}$, we can parameterize the undistorted case and normalize points as a function of λ and w. Similarly, the constraints are

$$\mathbf{p}_{u_i}^T(\lambda, w)\mathbf{E}\mathbf{p}'_{u_i} = 0, \quad i = 1, \ldots, n. \tag{6.7}$$

We then use the intrinsic constraints on essential matrices. The singularity of the essential matrix is enforced as

$$\det(\mathbf{E}) = 0. \tag{6.8}$$

The trace constraints, which indicate that the two singular values are equal, are expressed as

$$2\mathbf{E}\mathbf{E}^T\mathbf{E} - \text{trace}(\mathbf{E}\mathbf{E}^T)\mathbf{E} = 0. \tag{6.9}$$

6.3 Polynomial Solvers

In this section, we describe the numerical schemes for solving the polynomial systemsḞor all of these problems, the solution involves the following steps: (i) a linear elimination step [65] to reduce the number of unknowns in the polynomial system, (ii) solving the reduced polynomial system using Gröbner basis methods [20].

6.3.1 8 Point Case: F + λ

We parameterize the fundamental matrix $\mathbf{F}$ as

$$\mathbf{F} = \begin{bmatrix} f_1 & f_4 & f_7 \\ f_2 & f_5 & f_8 \\ f_3 & f_6 & f_9 \end{bmatrix}. \tag{6.10}$$

To fix the scale of the fundamental matrix, we assume that $f_9 \neq 0$ and choose f_9 = 1.

Linear Elimination. We first look at the 8 equations in (6.4) given by 8 point correspondences. This gives a polynomial system in matrix form as follows

$$\mathbf{G}\mathbf{x} = 0, \tag{6.11}$$

where $\mathbf{G}$ is a 8×12 coefficient matrix, and $\mathbf{x}$ is a vector of monomials $[\lambda f_3, \lambda f_6, f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, \lambda, 1]^T$. Given that $\{f_1, f_2, f_4, f_5, f_7, f_8\}$ are linear in (6.11), we can eliminate these 6 unknowns using 6 of the 8 equations. We can further choose to eliminate one more unknown f_3 or f_6 using the remaining 2 equations. Here we choose to eliminate f_3. To achieve this, we use the two last equations to eliminate f_3 and λf_3. To this end, we can rewrite the two equations as $f_3 = g_1(f_6, \lambda)$ and $\lambda f_3 = g_2(f_6, \lambda)$. Thus, we obtain a new polynomial system with only 2 unknowns i.e. f_6 and λ. With the singularity constraint on $\mathbf{F}$ in (6.5), we can then obtain a bivariate polynomial system

$$\lambda g_1(f_6, \lambda) - g_2(f_6, \lambda) = 0, \tag{6.12}$$

$$\det(\mathbf{F}(f_6, \lambda)) = 0. \tag{6.13}$$

The two equations are of degree 3 and 6, respectively.

Polynomial System. By expressing the resulting polynomial system with coefficients in $\mathbb{Z}_p$, and using algebraic geometry tools [33], we find that there are in general 8 solutions for this problem. This size of the solution set is much smaller than two-sided cases with varying (24 solutions) and constant (16 solutions) radial distortion. Then we generate the single elimination template by multiplying the two equations with a set of multiplication monomials. The multiplication monomials are chosen in such a way that (i) the maximum monomial degree in the resulting polynomial equations is 7, and (ii) the highest degree of any of the two unknowns in the multiplication monomials is 4. We arrive at an elimination template of 12 equations and 24 monomials. Then we use the column-pivoting scheme to select the monomial basis to form the 8×8 action matrix. For this problem, we find that choosing the last 9 monomials (in *grevlex* order) as the permissible set gives good stability. After we construct the action matrix, solutions to the polynomial system can be extracted from the eigenvectors of the transpose of the action matrix. Once we have found solutions for λ and f_6, the solutions of the other unknowns can be calculated linearly.

6.3.2 7 Point Case: $\mathbf{E} + \lambda + f$

For this problem, instead of parameterizing the essential matrix directly, we choose to parameterize $\mathbf{F}$ and solve for $\mathbf{E}$ implicitly. The reason is that, with this implicit parameterization, we achieve a simpler elimination step.

Linear Elimination. Similar to Section 6.3.1, we first use point equations in (6.4) given by 7 point correspondences. We use 6 of the equations to linearly eliminate $\{f_1, f_2, f_4, f_5, f_7, f_8\}$. With the last remaining equation, we can rewrite λf_6 as a quadratic function $h(f_3, f_6, \lambda)$ of f_3, f_6 and λ such that

$$\lambda f_6 - h(f_3, f_6, \lambda) \quad = 0. \tag{6.14}$$

Now we can write the fundamental matrix $\mathbf{F}$ as a function of f_3, f_6 and λ.

With the parameterization of $\mathbf{K}$ as in (6.2), we can express the essential matrix $\mathbf{E}$ with respect to the fundamental matrix:

$$\mathbf{E} = \mathbf{K}(w)^T \mathbf{F}(f_3, f_6, \lambda)\mathbf{K}'. \tag{6.15}$$

By substituting (6.15) into (6.8) and (6.9), along with (6.14), we obtain in total 11 equations in 4 unknowns f_3, f_6, w and λ.

Polynomial System. Before verifying the number of solutions, we need to simplify the equation system further. The first observation is that there are 4 equations where w is the common multiplier. This will potentially introduce a set of false solutions corresponding to $w = 0$. To cope with this, we simply divide all these 4 equations by w (this is done implicitly by changing the monomials in the equations). Secondly, we find that after this division step, monomials involving w only appear in the 9 equations from (6.9) in quadratic form as w^2. We therefore substitute w^2 with a new variable $z = w^2$. To this end, we have one quadratic equation (6.14), one 5th degree equation (6.8) and 9 equations from (6.9) (3 equations of 5th degree and 6 equations of 6th degree).

With these simplification and using [33], we verify that in general there are 19 solutions for this system. We generate the elimination template by multiplying a set of multiplication monomials such that (i) the maximum degree of the monomials in the resulting equations is 8, and (ii) the highest degrees of the unknowns in the multiplication monomials are $[3, 4, 2, 4]$, respectively. The resulting elimination template consists of 200 equations and 231 monomials. We observe that further reducing the size of the template by limiting the degrees in the multiplication monomials generally deteriorate the numerical stability of the solver. We then choose the last 40 monomials in *grevlex* order as the permissible set for basis selection. Again, solutions to the polynomial system can be extracted from the eigenvectors of the transpose of the 19×19 action matrix.

6.3.3 6 Point Case: $\mathbf{E} + \lambda$

Similar to the fundamental matrix, we parameterize the essential matrix $\mathbf{E}$ as

$$\mathbf{E} = \begin{bmatrix} e_1 & e_4 & e_7 \\ e_2 & e_5 & e_8 \\ e_3 & e_6 & e_9 \end{bmatrix}. \tag{6.16}$$

and set $e_9 = 1$ to fix the scale.

Linear Elimination. From point equations in (6.6) given by 6 point correspondences, we have a polynomial system in matrix form as follows

$$\mathbf{H}\mathbf{x} = 0, \tag{6.17}$$

where $\mathbf{H}$ is a 6×12 coefficient matrix, and $\mathbf{x}$ is a vector of monomials $[\lambda e_3, \lambda e_6, e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, \lambda, 1]^T$. Again, since $\{e_1, e_2, e_4, e_5, e_7, e_8\}$ are linear in (6.17), we can eliminate these 6 unknowns using the 6 point equations.

With the two intrinsic constraints (6.8) and (6.9), we arrive at a well-defined polynomial system of 10 equations (one from (6.8) and 9 from (6.9)) in 3 unknowns e_3, e_6 and λ. These equations are of degree 5 and 6.

Polynomial System. We find that there are in general 26 solutions for this problem using [33]. In contrast, the two-sided case with constant unknown radial distortion [68] has in general 52 solutions and is much more difficult to solve. To generate the elimination template, we choose the set of multiplication monomials such that (i) the maximum monomial degree in the resulting equations are up to degree 8, and (ii) the highest degrees of the unknowns in the multiplication monomial set are $[3, 4, 4]$ respectively. We then obtain an elimination template of 48 equations and 70 monomials. For the basis selection step, we have chosen the permissible set to be the last 32 monomials in *grevlex* order. Solutions of e_3, e_6 and λ can be extracted from the eigenvectors of the transpose of the 26×26 action matrix.

6.3.4 Alternative Parameterization

In [5, 11], another way to form polynomial systems for these minimal problems is described. Let us consider the example of estimating fundamental matrix and a single radial distortion. Instead of parameterizing λ and $\mathbf{F}$ directly, in [11], image points are lifted to 4D space with a corresponding 4 by 3 *radial distortion matrix* which has 11 parameters (after fixing the scale). In this way, the point equations in (6.4) can be written as linear equations. By parameterizing the solutions as the null space of the n linear equations, we can eliminate n unknowns. Thus, for the one-sided 8-point case,

we will have a polynomial system of 3 unknowns. Recall that in Section 6.3.1, using the linear elimination scheme, we obtain a polynomial system of only 2 unknowns which facilitates the subsequent polynomial solving step. And for the other two cases with even fewer correspondences, straightforward adaption of such lifting technique yields much more involved polynomial systems than the linear elimination scheme used here.

6.3.5 Degenerated Cases

It is well-known that for planar scenes, the problem of determining the fundamental matrix from two views degenerates [43]. Therefore, both one-sided 8-point and one-sided 9-point for estimating fundamental matrix and radial distortion degenerate for planar scenes. On the other hand, there are two degeneracies related to our formulation. The first one is the degeneracy for planar scenes for one-sided 7 point and one-side 6 point formulation. This is caused by the rank-deficiency in the linear elimination step of our formulation. The other degeneracy relates to fixing f_{33} or e_{33} to be 1, which leads to failures of the algorithms for forward motions where $f_{33} = 0$ or $e_{33} = 0$. Using different parameterization or constructing specialized solvers for planar scenes can resolve these problems. Further works are needed to derive general and fast solvers that avoid these degeneracies.

6.4 Experiments

In this section, we first study numerical stability and noise sensitivity of the proposed solvers with synthetic data. Then, we demonstrate the usefulness of the solvers on real image data.

6.4.1 Synthetic Experiments

To simulate synthetic scenes, we randomly generate 3D points within a cube of width 1000 units centered at the origin. We place the two cameras to be around 1000 units away from the origin with a random rotation pointing approximately to the origin. The length of translation between the cameras is chosen to be around 300 units. The focal lengths of the cameras are uniformly generated in the range of $[1000, 2000]$. Then we generate image point correspondences by projecting the 3D points onto the image planes (1000×1000 pixels) of both cameras. The image point coordinates are then distorted based on (6.1) with radial distortion coefficients chosen randomly from the interval $[-0.2, 0.2]$. Note that we specify the radial distortion with respect

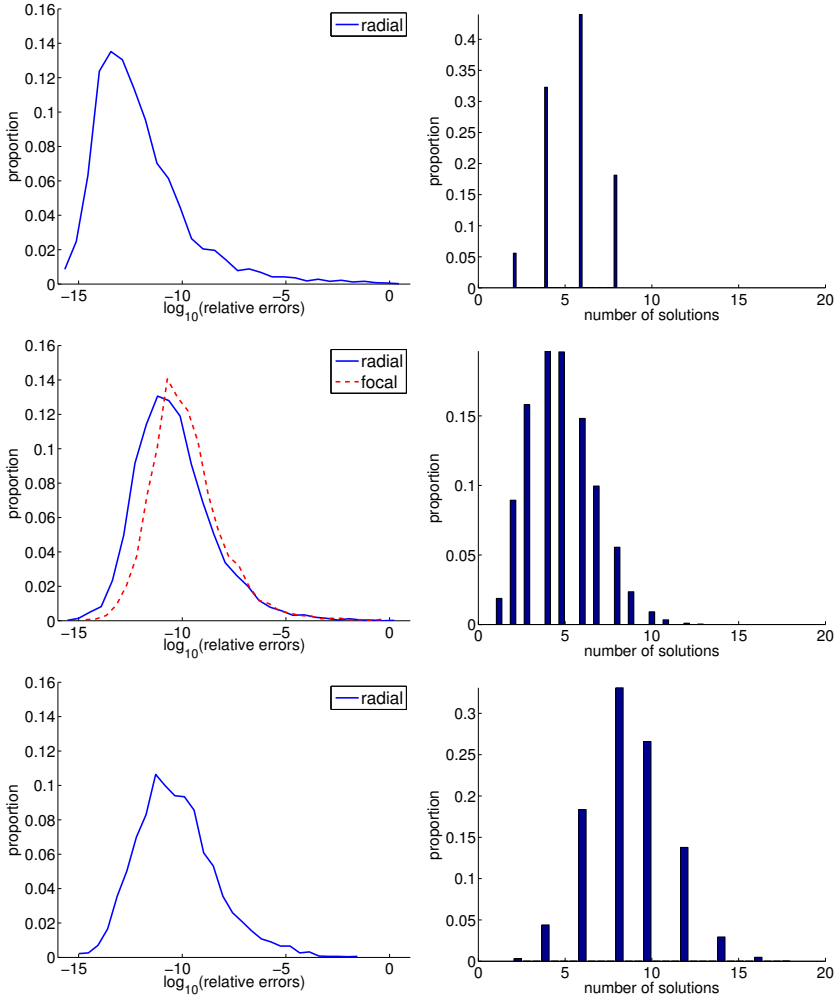


Figure 6.2: Left : Numerical stability of the different solvers (from top to bottom: 8 point, 7 point and 6 point minimal cases for one-sided radial distortion). Error distribution compared with ground-truth radial distortion and focal length on synthetic experiments with 5000 randomly generated problems. Right: Distribution of the number of valid solutions.

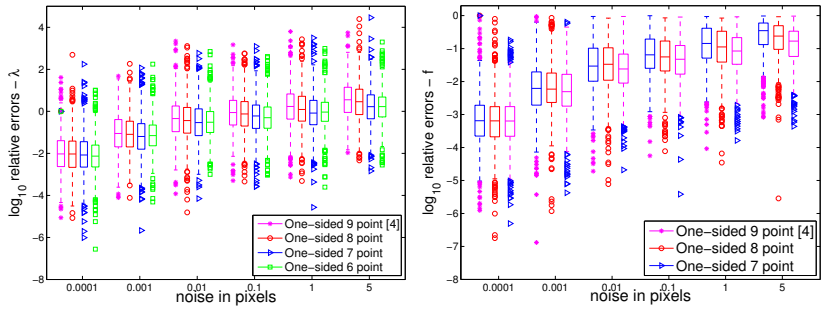


Figure 6.3: Noise sensitivity of different solvers in estimating radial distortion (Left) and focal length(right) with varying noise levels. Blue box specifies the 25 and 75 percentiles for the log_{10} relative errors.

to normalized image point coordinates in the range [-1, 1]. Outliers are simply image points randomly generated with a uniform distribution.

Numerical Stability. We first look at the numerical stability of the solvers on noise-free data. We evaluate the solvers with relative errors between the estimated f, λ and the ground truth. The distributions of log_{10} relative errors for different solvers are shown in Figure 6.2. We can see that both λ and f are estimated with high accuracy. For 5000 random problems, the medians of the log_{10} relative errors for λ are -11.58, -9.54 and -10.01 for 8-point, 7-point and 6-point cases, respectively. The median of the log_{10} relative errors for f estimated in the 7-point algorithm is -10.65.

Number of Solutions. It is of interest to investigate the number of *valid solutions* for these minimal cases under general settings. By valid solutions, here we mean real solutions for the radial distortion λ and positive solutions for f. For the 8-point problem, we see that (Figure 6.2, right) that there can be up to 8 real solutions in general. On the other hand, for the 7-point problem where there are in general 19 solutions, only 3-8 of those solutions are valid in most of the cases. As for the 6-point problem which has 26 solutions, we observe that very often, only fewer or equal to 16 of those solutions are real. The small number of valid solutions compared to the double-sided solvers further facilitates the solution verification step in RANSAC.

Noise Sensitivity. To understand the performance of the solvers in the presence of noise, we perturb the image points with Gaussian noise of varying standard deviations σ. Here we are interested in the noise sensitivity of estimating the radial distortion.

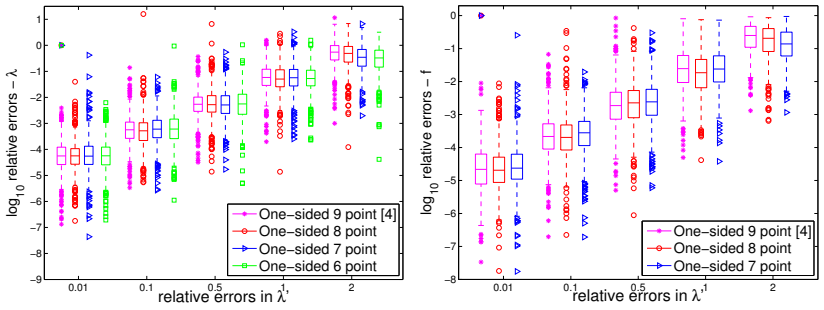


Figure 6.4: Sensitivity of the different solvers on calibration errors in radial distortion (λ') of the second camera. Blue box specifies the 25 and 75 percentiles for the log_{10} relative errors.

For small perturbations ($\sigma = 0.01, 0.1, 0.5$) on the image point coordinates, we see that all solvers give fairly good initial solutions to the radial distortion (Figure 6.3, left). For larger perturbations, e.g. $\sigma = 1, 2$, the relative errors become higher. This is similar to previous minimal solvers with radial distortion [11, 68]. These solutions from minimal configurations can be utilized in the kernel voting scheme or serve as initial solutions (after RANSAC) for bundle adjustment. While one is not aiming for better noise tolerance, the three minimal solvers proposed here perform marginally better than the non-minimal 9-point solver in [11][1].

Estimating Focal Length. We also test and compare different solvers in estimating focal lengths with noisy measurements. For the cases where only the fundamental matrix is estimated i.e. the 9-point in [11] and our 8-point solver, we have extracted the focal length from the fundamental matrix using the method in [11]. We can see from Figure 6.3 (right) that all the solvers perform similarly under settings with low noise levels. When the measurements are perturbed by noise of higher or equal to 1 pixel, our proposed solvers perform better, while requiring fewer number of correspondences. This further justifies the usefulness of the new minimal solvers in robust estimation settings. When comparing to the estimation of radial distortion, we observe that the estimating focal length is less sensitive to the noise i.e. the errors are lower for focal length than radial distortion under the same noise level.

Effects of Calibration Errors. In this experiment, we investigate how the errors in the calibration of the second camera affect the estimation of λ and f. We focus

[1]For all comparisons in this chapter, we have used the publicly available solvers at http://www.cvg.ethz.ch/research/distortion-in-multiple-view-geometry/

on the effects of inaccurate calibration on the radial distortion of the second camera. To study this, we add uniform random noise to the true radial distortion λ' of the second camera. Here, we have chosen fixed values for λ and λ' such that $\lambda = -0.2$ and $\lambda' = -0.1$. We observe (Figure 6.4) that for all solvers, the errors of estimating λ increase approximately linearly with respect to the calibration errors in λ'. On the other hand, there is no significant difference between the sensitivity of different solvers to the calibration errors.

Speed. All minimal solvers have been implemented in MATLAB. On a Macbook Air with 8GB memory and 1.8 GHz i5 CPU, the average runtime for the different solvers are: 1.0 ms (milliseconds) for the one-sided 8-point, 8.6 ms for the one-sided 7-point and 3.2 ms for the one-sided 6-point. The solvers can be further optimized in C or C++, e.g. with a more efficient construction of the coefficient matrix, which is one of the current bottleneck in MATLAB. For comparison, the one-sided 9-point method in [11] solves a cubic equation system, which takes on average 0.5 ms.

6.4.2 Real Experiments

In this section, we evaluate and validate the proposed solvers on real images. We have used an Nikon D60 camera as the fully calibrated camera in our setup. It has been verified that the Nikon D60 has very small radial distortion thus we assume that there is no radial distortion in it. We have also captured images (3000×2292) with a GoPro Hero3 camera, which shows significant radial distortion. To evaluate the estimation of the essential matrix as well as focal length, we calibrated the fixed focal length of the GoPro camera using the OpenCV calibration toolbox, which then serves as ground truth. For the experiments, we collected in total 25 pairs of images of different scenes with the two cameras. For each image in the dataset, we detect interest point and extract SIFT [72] features. Then for each pair of images, we obtain tentative descriptor matches based on Lowe's criterion. Given these preliminary matches, we estimate the fundamental matrix or the essential matrix using different solvers in the RANSAC loop.

In Figure 6.5, we show several example image pairs where we compare inliers from our minimal solvers to those obtained by the standard 7-point fundamental estimation. It can be seen that, by modeling radial distortion, we obtain more geometrically correct inliers for the image pairs. The comparison with the standard 8-point method [44] for fundamental matrix is very similar and thus is not shown here.

We then look at the gain of inliers for different minimal problems with the same number of RANSAC iterations. In Table 6.2, we can see that all the one-sided radial methods obtain around 60-70% more inliers than the baseline 7-point method. Our proposed minimal solvers obtain slightly more (3-5%) inliers than the non-minimal solver in [11]. We can also see that the inlier set for different one-sided solvers are

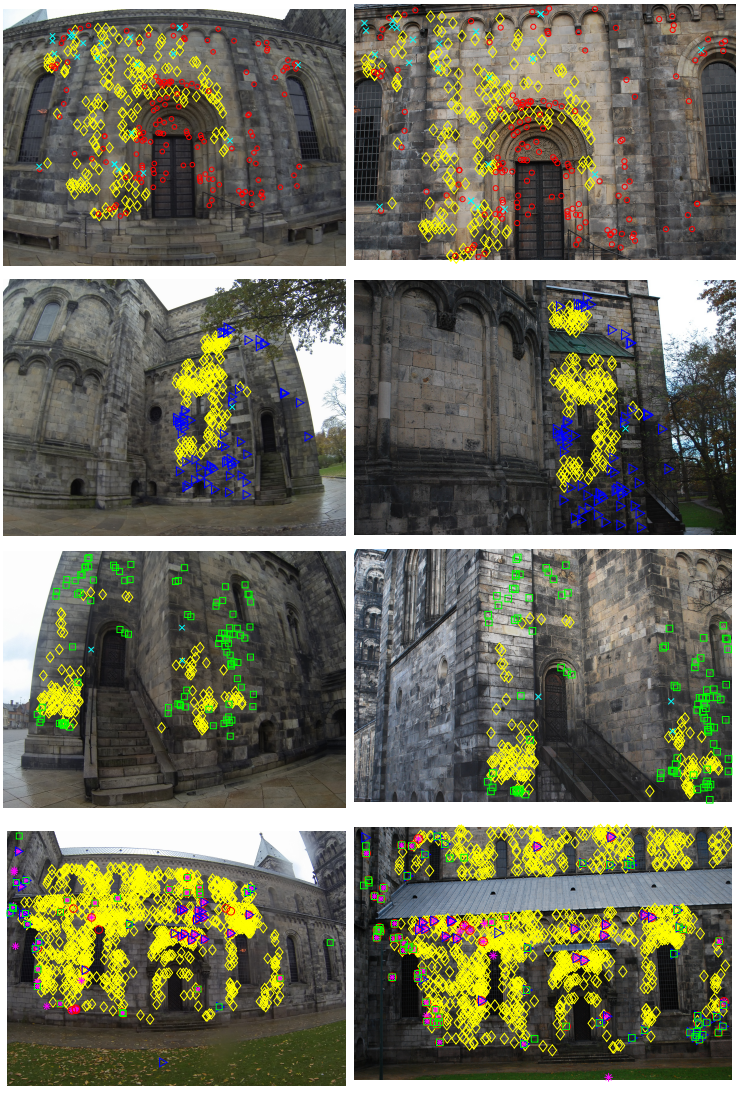


Figure 6.5: Examples on real image pairs. Left: images taken with an GoPro-Hero3; Right: images taken with a Nikon D60 camera. Yellow diamonds are inliers obtained by the standard 7-point method. Cyan crosses are inliers only obtained by the 7-point method. The others are extra inliers obtained by the specific one-sided minimal solver from the 1st to the 3rd row: one-sided 8 point, one-sided 7 point, one-sided 6 point. The last row is the comparison of inliers obtained by different one-side solvers where yellow diamonds are the common inliers by the four one-sided method.

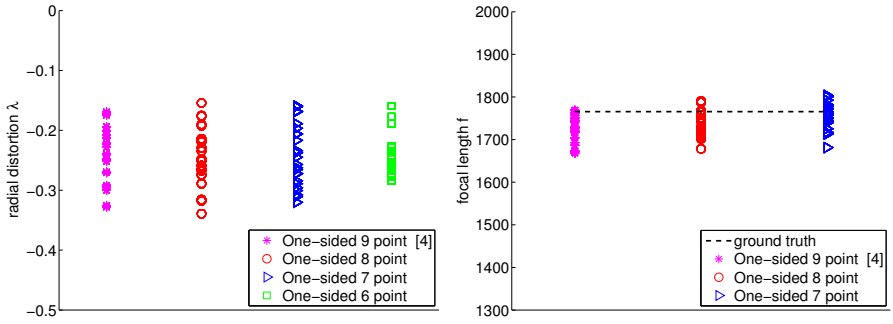


Figure 6.6: Estimation of radial distortion and focal length for the GoPro Hero3 camera from 25 image pairs.

consistent with each other (Figure 6.5, last row).

9 point [4]	8 point	7 point	6 point
1.67	1.71	1.72	1.70

Table 6.2: Average ratio of inliers obtained by different one-sided radial solvers to inliers obtained by the standard 7-point algorithm on the dataset.

We then study the noise sensitivity and consistency of the one-sided solvers on real images. In Figure 6.6 (Left), the estimates of the radial distortion for 25 GoPro-Nikon pairs are shown. We can see that the estimates from all solvers are centered at approximately -0.25. However, the variance of the estimates is fairly large due to the presence of noise. For the focal length (6.6, right), we again observe that the estimates for solvers are close to the ground truth from the calibration. To mitigate the effects of noise, we can also apply the kernel voting scheme of [71] to obtain better estimates for focal length and radial distortion. In Figure 6.7, we show an example of corrected images according to radial distortion estimated from different one-sided methods. We can see that most of the radial distortion are correctly reduced.

6.5 Conclusions

These minimal solvers introduced in this chapter enable calibration of a camera with unknown radial distortion using another camera with known or no radial distortion. More importantly, these simpler one-sided solvers enable robust estimation of radial

Figure 6.7: Example of corrected images based on the estimated λ by different solvers. Top left: Original image. Top right, bottom left and bottom right are images corrected according to 8-point, 7-point and 6-point one-sided method respectively.

distortion in large-scale unordered structure from motion. In large-scale unordered structure from motion, the proposed solvers can estimate a fundamental or essential matrix robustly with respect to seed images with known or no radial distortion. Such estimation can be readily integrated with both sequential and non-sequential structure from motion methods. The study of these one-sided cases could pave the way to a deeper understanding of the previous unsolved two-sided cases as well. In particular, our solution to the one-sided essential matrix with unknown focal length and radial distortion shed light to the unsolved two-sided minimal problem of essential matrix with constant unknown focal length and radial distortion. In our initial study, with the similar elimination and parameterization for the one-sided problem, we have implemented a fairly stable polynomial solver for this problem with 68 solutions.

There are several interesting avenues for one-side radial distortion calibration. Of particular theoretical interest is the critical configuration of these minimal problems. Solving minimal problems for essential matrix for one-sided radial with unknown distortion center is another direction to pursue.

Part III

Geometric Problems in Sensor Networks

Chapter 7
TOA Self-Calibration

In this chapter, we study the sensor network self-calibration problem with time-of-arrival (TOA) measurements. Traditionally, the calibration of sensor network relies on with prior knowledge on receiver positions and local optimization. An efficient solution to the TOA self-calibration problem opens up new technological possibilities e.g. calibration of a sensor network on the fly, determining positions of receivers and transmitters while moving in an unknown terrain etc. We provide a complete characterization of TOA self-calibration problem and derive fast and stable minimal solvers for different spatial configurations of receivers and transmitters.

7.1 Background

Sensor network calibration with time-of-arrival measurements arises in application such as calibration of radio antenna networks, audio or ultra-sound arrays and WiFi transmitter arrays. Extensive works has been done in the literature previously. For completely connected graph i.e. given all pair-wise distances between sensor nodes, the problem can be solved using multi-dimensional scaling [101]. For general graph, the well-studied semi-definite programming [6, 105] is very efficient for large scale network. However, it requires at least $k + 1$ anchor sensors with known positions, where k is the dimension of the affine span of the sensor positions. In anchor-free setup, [88] explored graph rigidity to find a fold-free graph embedding as the initial solution and refined the solution using local gradient method based on heuristics. The main contribution of [88] is that it proposed a way to generate a good initial solution based on graph rigidity. It was shown that, the method as well as other methods based on local iterative optimization will fail drastically without a good initial solution. Another line of works have focused on anchor-free localization for sensor network with bipartite structures. In [7], Bolker and Roth discussed and proved the minimal cases in 2D and 3D. However, no numerical solutions were given. The minimal cases are also studied by [92], where solutions to the minimal case of 3 transmitters and 3 receivers in the plane is given. However, no practical methods for general 3D positions are given. In [87], a non-minimal linear solution to 3D TOA self-calibration problem

is derived for 10 (4) receivers and 4 (10) transmitters. In [30, 31], a solution is given to the bipartite TOA self-calibration problem, if one may additionally assume that the position of one of the receivers coincides with the position of one of the transmitters.

7.2 The TOA Self-Calibration Problem

Let $\mathbf{r}_i$, $i = 1, \ldots, m$ and $\mathbf{s}_j$, $j = 1, \ldots, n$ be the spatial coordinates of m receivers (e.g. microphones) and n transmitters (e.g. sound events), respectively. For measured time of arrival t_{ij} from transmitter $\mathbf{r}_i$ and receiver $\mathbf{s}_j$, we have $vt_{ij} = \|\mathbf{r}_i - \mathbf{s}_j\|_2$ where v is the speed of measured signals. We assume that v is known and constant, and that we at each receiver can distinguish which transmitter j each event is originating from. This can be done e.g. if the signals are temporally separated or using different frequencies. We will in the sequel work with the distance measurements $d_{ij} = vt_{ij}$. The TOA calibration problem can then be defined as follows.

Problem 7.2.1 (TOA Self-Calibration) Given absolute distance measurements d_{ij} determine receiver positions $\mathbf{r}_i$, $i = 1, \ldots, m$ and transmitter positions $\mathbf{s}_j$, $j = 1, \ldots, n$ such that $d_{ij} = \|\mathbf{r}_i - \mathbf{s}_j\|_2$.

In TOA self-calibration problem, the underlying graph is bipartite (the two disjoint sets of vertices are the receiver nodes and transmitter nodes, respectively). Note also that for such problems, one can only reconstruct locations of receivers and transmitters up to an Euclidean transformation and mirroring, henceforth referred to as *gauge freedom*. In the following discussions, we assume that the dimensionality of the affine space spanned by $\mathbf{r}_i$ and $\mathbf{s}_j$ is the same and it is denoted as k. Typical value in practice is $k = 3$ for transmitters and receivers in general 3D positions.

For this special type of bipartite network structure, one also aims to identify and solve the minimal problem i.e. minimal number of receivers and transmitters required for the problem to be well-defined (or solvable). Note that for this problem, the roles of receivers and transmitters are equivalent. Therefore, when discussing minimal cases, the number for receivers and transmitters are exchangeable. To identify minimal problems, a straightforward way is to calculate the number of degrees of freedom in the measurements d_m and the number of degrees of freedom in the manifold of unknown parameters d_f The necessary condition for a configuration or a case to be minimal is that $d_m = d_f$. For TOA self-calibration problem in general $\mathbb{R}^k$ space, $d_m = mn$ and $d_f = k(m + n) - g(k)$, where $g(k) = k(k + 1)/2$ is the gauge freedom in general k-dimensional space. However, to be more precise one has to study the equations using algebraic geometry and graph theory, to make certain that there are no anomalies in the set of constraints. The corresponding statement for the minimal cases of TOA self-calibration problem is to decide when a bipartite graph is rigid given the edge lengths between two independent sets of m nodes and n nodes. It is known in 19th century that $(m = 3, n = 3)$ in $\mathbb{R}^2$ is rigid and Bolker and Roth [7] also show that

rigid cases in $\mathbb{R}^3$ are $(m = 6, n = 4)$ and $(m = 5, n = 5)$. We denote in the sequel the cases with m receivers and n transmitters as mr/ns. We will in the following derive a general parameterization for these minimal cases that yield stable numerical solvers.

7.2.1 Minimal Cases

In this section, we describe a parameterization of TOA self-calibration problem and derive algorithms for minimal cases.

Since the distance measurements are assumed to be real and positive, one does not lose any information by squaring the basic measurement equations $d_{ij} = \|\mathbf{r} - \mathbf{s}_j\|$ on both sides, i.e.

$$d_{ij}^2 = (\mathbf{r}_i - \mathbf{s}_j)^T(\mathbf{r}_i - \mathbf{s}_j) = \mathbf{r}_i^T\mathbf{r}_i + \mathbf{s}_j^T\mathbf{s}_j - 2\mathbf{r}_i\mathbf{s}_j.$$

Notice that these are now polynomial equations in the unknowns. We then form new equations by the following linear combinations of d_{ij}^2:

$$\begin{pmatrix} d_{11}^2 & d_{12}^2 - d_{11}^2 & \dots & d_{1n}^2 - d_{11}^2 \\ d_{21}^2 - d_{11}^2 & & & \\ \dots & & \bar{\mathbf{D}} & \\ d_{m2}^2 - d_{11}^2 & & & \end{pmatrix} \tag{7.1}$$

where $\bar{\mathbf{D}}$ is a $(m-1)(n-1)$ matrix with entries as $\bar{d}_{ij} = d_{i,j}^2 - d_{i1}^2 - d_{1j}^2 + d_{11}^2$, with $i = 2, \dots m$ and $j = 2, \dots, n$. These mn new equations are equivalent to the mn equations formed by d_{ij}^2. The new ones are in fact an invertible linear combinations of the old ones.

These equations are of four types:

(A) One equation: $d_{11}^2 = (\mathbf{r}_1 - \mathbf{s}_1)^T(\mathbf{r}_1 - \mathbf{s}_1)$.

(B) $n-1$ equations: $d_{1j}^2 - d_{11}^2 = \mathbf{s}_j^T\mathbf{s}_j - \mathbf{s}_1^T\mathbf{s}_1 - 2\mathbf{r}_1^T(\mathbf{s}_j - \mathbf{s}_1)$, $j = 2, \dots n$.

(C) $m-1$ equations: $d_{i1}^2 - d_{11}^2 = \mathbf{r}_i^T\mathbf{r}_i - \mathbf{r}_1^T\mathbf{r}_1 - 2(\mathbf{r}_i - \mathbf{r}_1)^T\mathbf{s}_1$, $i = 2, \dots m$.

(D) $(m-1)(n-1)$ equations: $d_{ij}^2 - d_{i1}^2 - d_{1j}^2 + d_{11}^2 = -2(\mathbf{r}_i - \mathbf{r}_1)^T(\mathbf{s}_j - \mathbf{s}_1)$, $i = 2, \dots m,\ j = 2, \dots n$.

The problem is symmetric in transmitters and receivers. Without loss of generality we may assume that $m \geq n$. It turns out that the characterization of the problem depends on the dimensions of the affine span of the transmitters and the receivers.

We first describe the factorization step using equations of Type D. Let $\bar{\mathbf{r}}_i = (\mathbf{r}_{i+1} - \mathbf{r}_1)$ and $\bar{\mathbf{s}}_j = -2(\mathbf{s}_{j+1} - \mathbf{s}_1)$. The equations of Type D can be written as $\bar{\mathbf{D}} = \bar{\mathbf{R}}^T\bar{\mathbf{S}}$ with $\bar{\mathbf{r}}_i$ as columns of $\bar{\mathbf{R}}$ and $\bar{\mathbf{s}}_j$ as columns of $\bar{\mathbf{S}}$. The ranks of $\bar{\mathbf{R}}$ and $\bar{\mathbf{S}}$ depend solely on the dimension of the affine span of the receivers and the transmitters respectively.

As we assume that both of these are k, then the matrix $\bar{\mathbf{D}}$ also has rank k. This also implies that in order to solve the problem, it is required that $m \geq k+1$ and $n \geq k+1$. We have the following lemma.

Lemma 7.1. *For positions of m transmitters and n receivers in k-dimensional affine space, the matrix $\bar{\mathbf{D}}$ is at most of rank k.*

Proof. The matrix $\bar{\mathbf{R}}$ and $\bar{\mathbf{S}}$ are of size $k \times (m-1)$ and $k \times (n-1)$ respectively. It can be easily seen that $\bar{\mathbf{D}} = \bar{\mathbf{R}}^T \bar{\mathbf{S}}$ is at most of rank-k as we increase m and n. □

Given that $\bar{\mathbf{D}}$ is of rank k, we can factorize $\bar{\mathbf{D}} = \hat{\mathbf{R}}^T \hat{\mathbf{S}}$ using e.g. singular value decomposition. Note that, using factorization, we can only determine $\bar{\mathbf{R}}$ and $\bar{\mathbf{S}}$ up to an unknown full-rank $k \times k$ transformation matrix $\mathbf{L}$ such that $\bar{\mathbf{D}} = \hat{\mathbf{R}}^T \mathbf{L}^{-1} \mathbf{L} \hat{\mathbf{S}} = \bar{\mathbf{R}}^T \bar{\mathbf{S}}$. To determine the unknown transformation $\mathbf{L}$ and the unknown initial positions $\mathbf{r}_1$ and $\mathbf{s}_1$, we need to utilize the remaining constraints in equations of Type A, B and C.

First we fix the translational part of the gauge freedom by choosing the location $\mathbf{r}_1$ at the origin. Given that $\bar{\mathbf{R}} = \mathbf{L}^{-T} \hat{\mathbf{R}}$ and $\bar{\mathbf{S}} = \mathbf{L} \hat{\mathbf{S}}$, by parameterizing $\mathbf{s}_1$ as $\mathbf{L}\mathbf{b}$ where $\mathbf{b}$ is a $k \times 1$ vector, we have :

$$\mathbf{r}_1 = \mathbf{0}, \tag{7.2}$$

$$\mathbf{s}_1 = \mathbf{L}\mathbf{b}, \tag{7.3}$$

$$\mathbf{r}_{i+1} = \mathbf{L}^{-T} \hat{\mathbf{r}}_i, \qquad i = 1 \ldots m-1, \tag{7.4}$$

$$\mathbf{s}_{j+1} = \mathbf{L}(\hat{\mathbf{s}}_j^* + \mathbf{b}), \; j = 1 \ldots n-1, \tag{7.5}$$

where $\hat{\mathbf{s}}_j^* = \hat{\mathbf{s}}_j / (-2)$.

We have the following constraints from the distance measurements of Type A, B and C on $\mathbf{r}_i$ and $\mathbf{s}_j$:

$$\begin{aligned} d_{11}^2 &= (\mathbf{r}_1 - \mathbf{s}_1)^T (\mathbf{r}_1 - \mathbf{s}_1) = \mathbf{s}_1^T \mathbf{s}_1 \\ &= \mathbf{b}^T \mathbf{L}^T \mathbf{L} \mathbf{b}, \end{aligned} \tag{7.6}$$

$$\begin{aligned} d_{1,j+1}^2 - d_{11}^2 &= \mathbf{s}_j^T \mathbf{s}_j - \mathbf{s}_1^T \mathbf{s}_1 \\ &= \hat{\mathbf{s}}_j^{*T} \mathbf{L}^T \mathbf{L} \hat{\mathbf{s}}_j^* + 2\mathbf{b}^T \mathbf{L}^T \mathbf{L} \hat{\mathbf{s}}_j^*, \end{aligned} \tag{7.7}$$

$$\begin{aligned} d_{i+1,1}^2 - d_{11}^2 &= \mathbf{r}_i^T \mathbf{r}_i - 2\mathbf{r}_i^T \mathbf{s}_1 \\ &= \hat{\mathbf{r}}_i^T (\mathbf{L}^T \mathbf{L})^{-1} \hat{\mathbf{r}}_i - 2\mathbf{b}^T \hat{\mathbf{r}}_i. \end{aligned} \tag{7.8}$$

Since all the equations above involve only $\mathbf{L}^T \mathbf{L}$ (and its inverse) and $\mathbf{b}$, we can represent $(\mathbf{L}^T \mathbf{L})^{-1}$ by a symmetric matrix $\mathbf{H}$ parameterized with $k(k+1)/2$ unknowns[1].

[1] This introduces implicitly a non-equality constraint that $\mathbf{H}$ must be positive definite denoted as $\mathbf{H} \succ 0$. We leave the discussion and optimization scheme on this to Section 7.5.

The constraints in (7.6), (7.7) and (7.8) can then be simplified as

$$d_{11}^2 = \mathbf{b}^T\mathbf{H}^{-1}\mathbf{b}, \tag{7.9}$$

$$d_{1,j+1}^2 - d_{11}^2 = \hat{\mathbf{s}}_j^{*T}\mathbf{H}^{-1}\hat{\mathbf{s}}_j^* + 2\mathbf{b}^T\mathbf{H}^{-1}\hat{\mathbf{s}}_j^*, \tag{7.10}$$

$$d_{i+1,1}^2 - d_{11}^2 = \hat{\mathbf{r}}_i^T\mathbf{H}\hat{\mathbf{r}}_i - 2\mathbf{b}^T\hat{\mathbf{r}}_i. \tag{7.11}$$

By utilizing $\mathbf{H}^{-1} = \operatorname{adj}(\mathbf{H})/\det(\mathbf{H})$, where $\operatorname{adj}(\mathbf{H})$ is the adjoint of $\mathbf{H}$, we can multiply equations in (7.9) and (7.10) on both sides by $\det(\mathbf{H})$ to rewrite them as polynomials equations. With this parameterization, we have in total $k(k+1)/2 + k$ unknowns $k(k+1)/2$ and k unknowns for $\mathbf{H}$ and $\mathbf{b}$, respectively). For 3D problems ($k = 3$), we have $(n+m-1)$ equations, among which the $(m-1)$ equations in (7.11) are linear, the $(n-1)$ equations in (7.10) are polynomial equations of degree 3 and Equation (7.9) is of degree 4. Thus we need $n+m-1 \geq 9$ or $n+m \geq 10$ in order to solve for the 9 unknowns. Since both $m \geq 4$ and $n \geq 4$, there are two minimal cases $4r/6s$ ($6r/4s$) and $5r/5s$. These cases correspond to the minimal cases proved in [7].

7.2.2 Solving the Polynomial Systems

In this section, we will present the related polynomial systems with unknowns $\mathbf{H}$ and $\mathbf{b}$ for each minimal problems. We derive a necessary saturation step and study the solution space of these systems. We then develop efficient and stable solvers based on action matrix method for both problems. For simplicity we will in the following presentation below concentrate on the 2 minimal cases for 3D problems. Notice, however, the method and theory are straightforward to generalize to other dimensions. A discussion on the self-calibration problem in other dimensions is given in Section 7.3.

4 Receivers and 6 Transmitters

For the minimal case of 4 receivers and 6 transmitters (6 transmitters and 4 receivers), there are 5 linear equations of type C. By linear elimination we can express $\mathbf{H}$ and $\mathbf{b}$ in terms of $9-5=4$ unknowns $\mathbf{x} = (x_1, x_2, x_3, x_4)$. We now obtain four equations

$$\det(\mathbf{H})d_{11}^2 = \mathbf{b}^T \operatorname{adj}(\mathbf{H})\mathbf{b}, \tag{7.12}$$

$$\det(\mathbf{H})(d_{12}^2 - d_{11}^2) = \hat{\mathbf{s}}_1^{*T} \operatorname{adj}(\mathbf{H})\hat{\mathbf{s}}_1^* + 2\mathbf{b}^T \operatorname{adj}(\mathbf{H})\hat{\mathbf{s}}_1^*, \tag{7.13}$$

$$\det(\mathbf{H})(d_{13}^2 - d_{11}^2) = \hat{\mathbf{s}}_2^{*T} \operatorname{adj}(\mathbf{H})\hat{\mathbf{s}}_2^* + 2\mathbf{b}^T \operatorname{adj}(\mathbf{H})\hat{\mathbf{s}}_2^*, \tag{7.14}$$

$$\det(\mathbf{H})(d_{14}^2 - d_{11}^2) = \hat{\mathbf{s}}_3^{*T} \operatorname{adj}(\mathbf{H})\hat{\mathbf{s}}_3^* + 2\mathbf{b}^T \operatorname{adj}(\mathbf{H})\hat{\mathbf{s}}_3^*. \tag{7.15}$$

in the four unknowns. Here both $\mathbf{H}$ and $\mathbf{b}$ depend on $\mathbf{x}$. Using tools from algebraic geometry it can be shown that the solution set to equations (7.12-7.15) in general consists of 38 points and a set of dimension 1 (a curve) of false solutions that fulfill $\det(\mathbf{H}) = 0$. This is done by running the system of equations in Macaulay2 [38]. To remove

the one-dimensional curve of false solutions corresponding to $z = \det(\mathbf{H}) = 0$, we employ a saturation technique similar to [93] as follows[2]. We rewrite the equations using an additional unknown z and an additional equation $\det(\mathbf{H}) = z$, i.e.

$$zd_{11}^2 = \mathbf{b}^T \operatorname{adj}(\mathbf{H})\mathbf{b}, \tag{7.16}$$
$$z(d_{12}^2 - d_{11}^2) = \hat{\mathbf{s}}_1^{*T} \operatorname{adj}(\mathbf{H})\hat{\mathbf{s}}_1^* + 2\mathbf{b}^T \operatorname{adj}(\mathbf{H})\hat{\mathbf{s}}_1^*, \tag{7.17}$$
$$z(d_{13}^2 - d_{11}^2) = \hat{\mathbf{s}}_2^{*T} \operatorname{adj}(\mathbf{H})\hat{\mathbf{s}}_2^* + 2\mathbf{b}^T \operatorname{adj}(\mathbf{H})\hat{\mathbf{s}}_2^*, \tag{7.18}$$
$$z(d_{14}^2 - d_{11}^2) = \hat{\mathbf{s}}_3^{*T} \operatorname{adj}(\mathbf{H})\hat{\mathbf{s}}_3^* + 2\mathbf{b}^T \operatorname{adj}(\mathbf{H})\hat{\mathbf{s}}_3^*, \tag{7.19}$$
$$\det(\mathbf{H}) = z. \tag{7.20}$$

We then multiply all equations with monomials such that the highest degree of monomials in the expanded set of equations is up to degree 9. By doing this one can construct 966 equations involving 715 monomials which do not contain z and 210 monomials that do contain z. These equations can be represented by a sparse coefficient matrix $\mathbf{M} = [\mathbf{M}_0 \;\; \mathbf{M}_z]$ of size 966×925, where the coefficients corresponding to monomials without z are in $\mathbf{M}_0$ and those corresponding to monomials with z are in M_z. After multiplication with $\mathbf{Q}^T$, where $\mathbf{QT} = \mathbf{M}_0$ is the QR-factorization of $\mathbf{M}_0$, we obtain

$$\mathbf{Q}^T\mathbf{M} = [\mathbf{T} \;\; \mathbf{Q}^T\mathbf{M}_z].$$

Here the last 336 rows of $\mathbf{T}$ is zero and thus the last 336 equations can all be written in the form of $zf(\mathbf{x}) = 0$. After division with z, we obtain 336 equations of degree 6 in $\mathbf{x}$. It can be verified using algebraic geometry that the solution set to these equations consist of 38 points.

We then use these equations for solving for the 38 solutions. To solve the polynomial system, we have used the action matrix method with column-pivoting basis selection.

Conjecture 1 (Solutions to Four (Six) Receivers and Six (Four) Transmitters) Given four (six) receivers and six (four) transmitters in general 3D positions, if the Euclidean distances between each receiver and each transmitter are known, there are in general 38 solutions.

For each such solution, we then calculate $\mathbf{H}$ and $\mathbf{b}$ and then generate the solutions for $\mathbf{r}_i$ and $\mathbf{s}_j$ according to (7.2), finding $\mathbf{L}$ by e.g. Cholesky factorization of $\mathbf{H}$. $\mathbf{L}$ is thus only determined up to a matrix $\mathbf{G}$ where $\mathbf{G}^T\mathbf{G} = I$, which coincides with the gauge freedom of rotating and/or mirroring our solution. Some of the solutions obtained are complex. Some solutions, although real, give invalid matrices $\mathbf{H}$ that are not positive definite which does not have a real decomposition into $(\mathbf{L}^T\mathbf{L})^{-1}$. The

[2]Another technique to remove undesired solutions is called Rabinovich's trick which introduces one additional equation $\det \mathbf{H}.z - 1 = 0$. In our experiment, it turns out that this technique is extremely numerically unstable. Thus, we were not able to generate any working solver with this technique.

number of real and valid solutions varies. Typical values are shown in a histogram, cf. Figure 7.2 (left).

5 Receivers and 5 Transmitters

The case of 5 receivers and 5 transmitters (5 transmitters and 5 receivers) is interesting. It is an overdetermined case in the sense that there are 25 measurements and 24 degrees of freedom in the solutions set. There is thus one constraint that has to be satisfied, i.e. the constraint that the 4×4 matrix $\bar{\mathbf{D}}$ has determinant zero. However for all such data, the problem of determining $\mathbf{H}$ and $\mathbf{b}$ is minimal. There are $m + n - 1 = 9$ equations (1 of type A, 4 of type B and 4 of type C) and 9 unknowns. We follow a similar solution scheme as for the $4r/6s$ case. By linear elimination using the 4 linear constraints of type C, we can express $\mathbf{H}$ and $\mathbf{b}$ in terms of $9 - 4 = 5$ unknowns $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$. The remaining five constraints

$$\det(\mathbf{H})d_{11}^2 = \mathbf{b}^T \operatorname{adj}(\mathbf{H})\mathbf{b} \tag{7.21}$$

$$\det(\mathbf{H})(d_{12}^2 - d_{11}^2) = \hat{\mathbf{s}}_1^{*T} \operatorname{adj}(\mathbf{H})\hat{\mathbf{s}}_1^* + 2\mathbf{b}^T \operatorname{adj}(\mathbf{H})\hat{\mathbf{s}}_j^* \tag{7.22}$$

$$\det(\mathbf{H})(d_{13}^2 - d_{11}^2) = \hat{\mathbf{s}}_2^{*T} \operatorname{adj}(\mathbf{H})\hat{\mathbf{s}}_2^* + 2\mathbf{b}^T \operatorname{adj}(\mathbf{H})\hat{\mathbf{s}}_j^* \tag{7.23}$$

$$\det(\mathbf{H})(d_{14}^2 - d_{11}^2) = \hat{\mathbf{s}}_3^{*T} \operatorname{adj}(\mathbf{H})\hat{\mathbf{s}}_3^* + 2\mathbf{b}^T \operatorname{adj}(\mathbf{H})\hat{\mathbf{s}}_j^* \tag{7.24}$$

$$\det(\mathbf{H})(d_{15}^2 - d_{11}^2) = \hat{\mathbf{s}}_4^{*T} \operatorname{adj}(\mathbf{H})\hat{\mathbf{s}}_4^* + 2\mathbf{b}^T \operatorname{adj}(\mathbf{H})\hat{\mathbf{s}}_j^* \tag{7.25}$$

gives 42 solutions after a saturation procedure similar to the previous case. Specifically, we first multiply all equations in (7.21-7.25) with all possible monomials in $\mathbf{x}$ such that highest degree of monomials in the resulting equations is $\mathbf{x}$ up to degree 8. This expanded set of 1386 equations involving 1287 monomials which do not contain z and 252 monomials that do contain z. These equations can be represented by a sparse coefficient matrix $\mathbf{M}$ of size 1386×1539. After similar QR-factorization and multiplication with $\mathbf{Q}^T$, we obtain 420 equations that can all be written $zf(\mathbf{x}) = 0$. After division with z, we obtain 420 equations of degree 5 in $\mathbf{x}$. Using algebraic geometriy, we verify that this solution set to these equations consists of 42 points. With these equations, we again use the action matrix method with column-pivoting basis selection to obtain numerically stable solutions.

Conjecture 2 (Solutions to Five (Five) Receivers and Five (Five) Transmitters) Given five (five) receivers and five (five) transmitters in general 3D positions, if the Euclidean distances between each receiver and each transmitter are known, there are in general 42 solutions.

The solver for 2D minimal case ($m = 3, n = 3$) can be derived in a similar manner. To summarize, the solvers for the minimal cases for TOA self-calibration problem in general k-dimensional space consists of (i) a factorization step (ii) a polynomial solver

Algorithm 7.2.1 Minimal Solvers for TOA Self-Calibration

Given TOA measurements $\mathbf{D} = \{d_{ij}\}$ of m receivers and n transmitters (corresponding to a minimal case in k-dimensional space), outputs the receivers positions $\{\mathbf{r}_i\}$ and transmitters positions $\{\mathbf{s}_j\}$.

1. Construct $\bar{\mathbf{D}} = \{\bar{d}_{ij}\}$ where $\bar{d}_{ij} = d_{ij} - d_{1j} - d_{i1} + d_{11}$.
2. Apply SVD on $\bar{\mathbf{D}} = \mathbf{U}\mathbf{\Gamma}\mathbf{V}^T$, $\hat{\mathbf{S}}^* = -\mathbf{V}_k^T/2$ and $\hat{\mathbf{R}} = (\mathbf{U}_k\mathbf{\Gamma}_k)^T$.
3. Solve for $\mathbf{H}$ and $\mathbf{b}$ that satisfy (7.9), (7.10) and (7.11).
4. Solve for $\mathbf{L}$ by Cholesky factorization on $\mathbf{H} = (\mathbf{L}^T\mathbf{L})^{-1}$.
5. Reconstructing $\{\mathbf{r}_i\}$ and $\{\mathbf{s}_j\}$. $\mathbf{r}_1 = \mathbf{0}$, $\mathbf{s}_1 = \mathbf{L}\mathbf{b}$, $\mathbf{s}_{2,\ldots,n} = \mathbf{L}(\hat{\mathbf{s}}^* + \mathbf{L}\mathbf{b})$, and $\mathbf{r}_{2,\ldots,m} = \mathbf{L}^{-T}\hat{\mathbf{r}}$.

for the unknown transformation parameterized by $\mathbf{H}$ and $\mathbf{b}$. This is summarized in Algorithm 7.2.1[3].

7.3 Lower and Higher Dimensions

The ideas presented here are relatively easy to generalize to other dimensions. The one-dimensional case is trivial. Only one measurement is needed to solve for the problem. In 2D planar case, the same approach can be used to solve the minimal problem $3s/3r$. The problem has in general 8 solutions. It was solved with a different approach by Stewénius in [92]. In that approach, the positions of the receivers and transmitters are recovered by solving for the *tri-sonal* tensor. The solver is also based on Gröbner basis and has a elimination template of size 31×35. In our approach, the size of our elimination template for saturation is 70×76, and becomes 18×20 after the saturation.

For 4-dimensional space, similar analysis gives constraints $m \geq 5, n \geq 5, m + n \geq 15$. Hence we conjecture that there are three minimal cases $5r/10s, 6r/9s, 7r/8s$. Similar to the $5r/5s$ case in 3D, for cases $6r/9s$ and $7r/8s$, there are additional nonlinear constraints on the entries in $\bar{\mathbf{D}}$ whose determinant should be zero. In [7], it is proved formally that $6r/7s$ which satisfies $mn = 4m + 4n - 10$, is in fact an underdetermined case. This is also confirmed with our formulation such that $m + n = 13 < 15$. While in practice, all such 4D measurements might be rare, we believe the analysis here is of theoretical interest regarding hardness of the problem or limitation of our formulation. All the cases in 4D are much more difficult to solve than

[3]For $k = 3$, the two cases are $4r/6s$ ($6r/4s$) and $5r/5s$. For $k = 2$, the minimal case is $3r/3s$.

the lower dimensional problems using the same formulation. To this end, we have not constructed any numerically stable solver for these problems. Nevertheless, we have the following conjecture on the solutions to the $10r/5s$ case using algebraic geometry.

Conjecture 3 (Solutions to Five (Ten) Receivers and Ten (Five) Transmitters) Given five (ten) receivers and ten (five) transmitters in general 4-dimensional space, if the Euclidean distances between each receiver and each transmitter are known, there are in general 223 solutions.

7.4 Difference in Dimension

In this section, we study a variant of the TOA self-calibration problem where the dimensions of the affine span of the receivers and transmitters are different. This occurs e.g. in the configurations where the receivers lie on a common plane and the transmitters are located at general 3D positions. Utilizing the same parameterization, we derive a linear solution to the minimal cases for TOA self-calibration with difference in dimensions. In the following, we assume that receivers lie in affine space of dimension k and transmitters in affine space of dimension $k+1$. We parameterize the positions of receivers and transmitters as $\mathbf{r}_i \in \mathbb{R}_{k+1}$ and $\mathbf{s}_j \in \mathbb{R}_{k+1}$ respectively. To fix the gauge freedom of the problem, we choose the coordinate systems such that $\mathbf{r}_1$ is at the origin and the $(k+1)^{th}$ coordinate of $\mathbf{r}_i$ is 0. With this and the type D equations used in TOA of general dimension:

$$\bar{d}_{ij} = d_{ij}^2 - d_{i1}^2 - d_{1j}^2 + d_{11}^2 = -2(\mathbf{r}_i - \mathbf{r}_1)^T(\mathbf{s}_j - \mathbf{s}_1),$$

we can see that the matrix $\bar{\mathbf{D}} = \{\bar{d}_{ij}\}$ is at most of rank k as we increase m and n. Specifically, the zeros at $(k+1)^{th}$ coordinates $\mathbf{r}_i$'s reduce the rank to the corresponding lower dimension spanned by the receivers. Given at least $k+1$ receivers and $k+1$ transmitters, we apply the linear factorization to find the projections of $\mathbf{r}_i$ and $\mathbf{s}_j$ onto the lower k dimensional space are retrieved and we denote them as $\hat{\mathbf{r}}_i$ and $\hat{\mathbf{s}}_j$ up to unknown transformation. We can then use the set of linear equations (7.11) in a similar manner as TOA in general dimension:

$$\begin{aligned} d_{i1}^2 - d_{11}^2 &= \mathbf{r}_i^T\mathbf{r}_i - 2\mathbf{r}_i^T\mathbf{s}_1 \\ &= \hat{\mathbf{r}}_i^T\mathbf{H}\hat{\mathbf{r}}_i - 2\mathbf{b}^T\hat{\mathbf{r}}_i \end{aligned} \tag{7.26}$$

where $\mathbf{H} \in \mathbb{R}^{k\times k}$ is symmetric and positive definite. For the unknown $\mathbf{H}$ and $\mathbf{b}$, there are in total $k(k+1)/2 + k$ unknowns. Given $m = k(k+1)/2 + k + 1$, we obtain the same number of linear equations as the number of unknowns and we can solve for $\mathbf{H}$ and $\mathbf{b}$ linearly. To this end, we can reconstruct the positions of the receivers as well as the projections of the transmitters onto the k-dimension affine space spanned by the receivers (up to an Euclidean transformation and mirroring). The remaining

nonlinear equations in (7.9) and (7.10) can then be utilized to recover the distance of the transmitters to the projected lower-dimensional space spanned by the receivers. To summarize, the linear solver requires $m = k(k+1)/2+k+1$ receivers and $n = k+1$ transmitters.

7.4.1 Minimal Problems

To identify the minimal problems of TOA self-calibration with difference in dimension, we start by counting the degrees of freedom of the problem. For m receivers in $\mathbb{R}_k$ and n transmitters in $\mathbb{R}_{k+1}$, the degrees of freedom is $d_f = km + (k+1)n - g(k)$ where $g(k) = (k+1)k/2$ is the gauge freedom. The necessary condition for a case to be minimal is that the number of measurement $d_m = mn$ is equal to the degrees of freedom of the problem. We can see that if $m = (k+1)k/2 + 1$ and $n = k+1$, we have $d_m = d_f$ for TOA self-calibration with difference in dimension. This shows that the linear solver in the previous section actually solves the minimal case for TOA self-calibration with difference in dimension. For instance, if receivers lie on a plane and transmitters located in general 3D positions ($k = 2$), the minimal case is $m = 6, n = 3$. Note that, in [16], the same minimal case is solved with rank-$(k+1)$ factorization in a slightly more complicated manner than the algorithm presented in this section.

7.4.2 Discussion

We have focused on the cases where the difference in dimensions of the receivers and the transmitters is 1. For larger difference in dimension, there exist infinite number of solution to the underlying problems. For instance, if receivers lie on a line and the transmitters lie on general 3D space, then one can only determine the projection of the transmitters positions onto the line, and their distances to the line. Therefore, the solutions to the positions of the transmitters are represented as one-parameter circles.

7.5 Failure Modes

In this section, we discuss the cases where our proposed method will fail in providing solutions. There are basically two failure modes for noise-free data. The first failure mode relates to the rank-k constraint on the transformed measurement matrix $\bar{\mathbf{D}}$ in (7.1). If $\mathrm{rank}(\bar{\mathbf{D}}) < k$, then the assumption for factorization is not satisfied. Hence, our method fails. This corresponds to cases where either the positions of receivers or the transmitters span a affine space of dimension lower than k. Specifically, for $k = 3$, the failures mode corresponds to cases when either the receivers or transmitters lie on a plane or a line. The method for solving these degenerate cases are discussed in Section 7.4. The second failure case relates to special configurations of positions of

the receivers and the transmitters, which could lead to graphs that is not infinitesimal rigid [7]. Hence, there exist infinitely many solutions to the problems. Specifically, the bipartite graph is not infinitesimal rigid if transmitters and receivers lie on a quadratic surface. For instance, in 2D, this corresponds to the 3 transmitters and 3 receivers lie on a conic.

For noisy data, there exist a failure case that is related to the implicit non-equality constraint that $\mathbf{H}$ must be positive definite. For noise-free data, it is always possible for our method to recover an $\mathbf{H}$ that satisfies the positive definite condition. However, for noisy data, there can be cases where no solutions for $\mathbf{H}$ from the minimal solvers that are positive definite. Thus, in these cases, our minimal solvers also fail in providing valid initial solutions. One scheme to resolve this is to formulate an intermediate semi-definite programming (SDP) step. Specifically, we have the following minimization problem based on (9)-(12) :

$$\begin{aligned} \min_{\mathbf{H},\mathbf{b}} \quad & \big(\det(\mathbf{H}) * d_{11} - \mathbf{b}^T\mathbf{H}^{-1}\mathbf{b}\big)^2 + \\ & \sum_{j=2}^{n} \Big(\det(\mathbf{H})(d_{1j}^2 - d_{11}^2) \\ & -\hat{\mathbf{s}}_j^{*T} \operatorname{adj}(\mathbf{H})\hat{\mathbf{s}}_j^* + 2\mathbf{b}^T \operatorname{adj}(\mathbf{H})\hat{\mathbf{s}}_j^* \Big)^2, \\ s.t. \quad & \mathbf{H} \succ 0. \end{aligned} \tag{7.27}$$

For this nonlinear optimization, we can use the solutions from our minimal solvers as initial solutions. This step may be applied when there exists relatively high magnitude of noise in the measurements.

7.6 Overdetermined Cases

In this section, we discuss the construction of solvers for overdetermined cases.

7.6.1 General Overdetermined Solvers

The general strategy to solve overdetermined cases is a two-step procedure (i) use minimal solvers to reconstruct robustly the positions of a subset of receivers and transmitters (ii) calculate the positions of the remaining receivers and transmitters with trilateration. Assuming that all the measurements are inliers, we use nonlinear optimization (Section 7.6.3) to obtain maximum likelihood estimates. One drawback of this approach is the accumulation of errors in the trilateration step. In the following, we study non-iterative solvers for a subset of overdetermined cases that can be very useful for initialization with outlier-free measurements.

7.6.2 Specialized Overdetermined Solvers

We study a subset of overdetermined cases that are solvable in a non-iterative manner. In 3D, these cases are $(m = 4, n > 6)$, $(m = 5, n > 5)$ or $(m = 6, n > 4)$. Here, we assume that the factorization step has been done as in Algorithm 7.2.1 and present two methods for solving $\mathbf{H}$ and $\mathbf{b}$ for these overdetermined cases.

To solve $\mathbf{H}$ and $\mathbf{b}$ for overdetermined cases, the first scheme is to use all the linear equations of Type C and choose only a subset of the nonlinear equations of Type A, B. For instance, for $(m = 7, n = 4)$ case, we can first utilize the 6 linear equations of Type C. With linear substitution, we are left with only 3 unknowns. There are still 1 Type A equation and 3 Type B equations. By choosing 3 equations from the 4 equations, we can solve the resulting system of polynomials. The remaining equation can be used to choose the solution based on the residues. Note that when $(m > 9, n = 4)$ or $(m = 4, n > 9)$, the solution can be extracted in one step using only the Type C linear equations, which corresponds to the solver in [87]. This scheme can still involve solving polynomial systems and computationally it can be expensive[4].

The second method uses all available constraints and solve for $\mathbf{H}$ and $\mathbf{b}$ linearly. The general idea is that for noise-free data, there generally exists one unique solution given the additional constraints in overdetermined cases. For noise-free cases, the unique solution can be extracted by finding the null space of an expanded coefficient matrix based on the equations. For noisy measurements in overdetermined systems, the solution to can be found directly using SVD.

To illustrate the formulation, we start by a simple overdetemined system of equations with two unknowns $\mathbf{x} = \{x_1, x_2\}$ and coefficient vector $\mathbf{a}, \mathbf{b}, \mathbf{c}$

$$\begin{cases} a_1 x_1^3 + a_2 x_1 + a_3 x_2 + a_4 & = 0 \\ b_1 x_1^2 + b_2 x_1 x_2 + b_3 x_2 + b_4 & = 0 \\ c_1 x_2^2 + c_2 x_1 + c_3 x_2 + c_4 & = 0 \end{cases}$$

It can be written as matrix multiplication form between a coefficient matrix $\mathbf{C}$ and a monomial vector $\mathbf{v}$ as follows:

$$\underbrace{\begin{pmatrix} a_1 & 0 & 0 & 0 & a_2 & a_3 & a_4 & \\ 0 & b_1 & & 0 & b_2 & 0 & b_3 & b_4 \\ 0 & 0 & c_1 & 0 & c_1 & c_2 & c_4 & \end{pmatrix}}_{\mathbf{C}} \underbrace{\begin{pmatrix} x_1^3 \\ x_1^2 \\ x_2^2 \\ x_1 x_2 \\ x_1 \\ x_2 \\ 1 \end{pmatrix}}_{\mathbf{v}} = \mathbf{0}. \tag{7.28}$$

[4]The saturation step as in the minimal cases is needed for cases $(m = 6, n = 7)$ and $(m = 5, n = 4)$.

One can expand the polynomial system by multiplying the original equations with monomials formed by the unknowns. We denote the coefficient matrix and the monomial vector after the expansion as $\mathbf{C}_{\text{exp}}$ and $\mathbf{v}_{\text{exp}}$, respectively. Assuming there are l_e equations and l_m monomials after expansion i.e. $\mathbf{C}_{\text{exp}}$ is a $l_e \times l_m$ matrix and $\mathbf{v}$ is a $l_m \times 1$ vector. By ignoring the constraints between the monomials in $\mathbf{v}_{\text{exp}}$ (e.g. for $\mathbf{v}$ in (7.28), $\mathbf{v}_5\mathbf{v}_6 = x_1x_2 = \mathbf{v}_4$), we have a linear system $\mathbf{C}_{\text{exp}}\mathbf{v}_{\text{exp}} = \mathbf{0}$. For noise-free overdetermined systems, it can be shown that $\text{rank}(\mathbf{C}_{\text{exp}}) < l_m$. Since there exists a unique solution to the system, if one choose a proper set of monomials to multiply with, one will get a $\mathbf{C}_{\text{exp}}$ that is of rank-$(l_m - 1)$. Therefore, the unique solution can be found by first computing the row null space of $\mathbf{C}_{\text{exp}}$. Then after normalizing the solution such that the last entry is 1, one can read off the solutions to $\mathbf{x}$ from the normalized solution. For noisy data, the rank-deficiency generally does not preserve and the solution can be obtained with SVD. In this case, the constraints between monomials in the solutions are also not satisfied. The effects of noise on this approach are studied in details in the experiment section. With this formulation, the overdetermined systems is solved linearly. For the TOA self-calibration problem studied here, we first use $\max\{m, n\} - 1$ Type C linear equations to reduce the number of unknowns as much as possible. By using all nonlinear equations of Type A and B for the remaining unknowns, we can then derive linear solvers for $\mathbf{H}$ and $\mathbf{b}$.

7.6.3 Nonlinear Optimization

In the presence of noise, the best parameter estimate is obtained by maximum likelihood estimates based on measurement noise model. The solutions obtained from the minimal solvers or the over-determined solvers can be used as initial solutions to the following nonlinear optimization problem:

$$\min_{\mathbf{r}_i, \mathbf{s}_j} \sum_{ij} \left(d_{ij} - \|\mathbf{r}_i - \mathbf{s}_j\|_2\right)^2,$$

which minimizes the sum of square deviation between the TOA measurements and the reconstruction. This minimization problem is solved using standard local optimization algorithm (Levenberg-Marquardt) and can be easily extended to measurements with missing data.

7.7 Experiments

In this section, we present extensive experimental results for our proposed solvers on both synthetic and real data. In particular, we investigate the numerical stability of the solvers, efficiency of the solvers as well as the sensitivity of the solvers to noise.

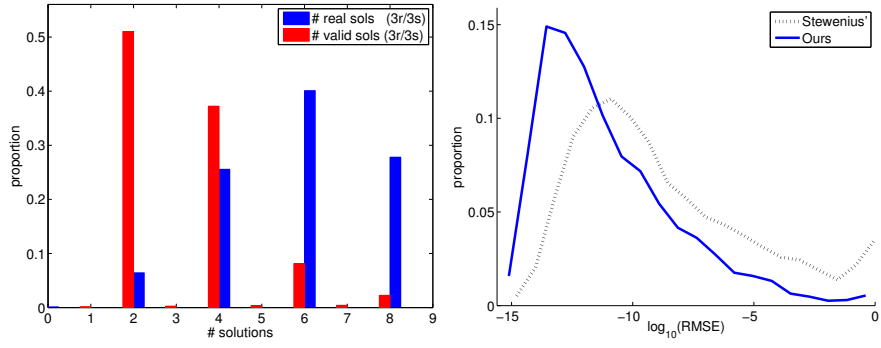


Figure 7.1: Synthetic experiment on 5000 noise-free random 2D TOA problems. Left: Distribution of the number of real and valid solutions each run produces by our solver, showing the relative frequency of number of real and valid solutions among the 8 solutions. Right: Performance of [92] (black, dotted line) and our solver (blue, solid line)

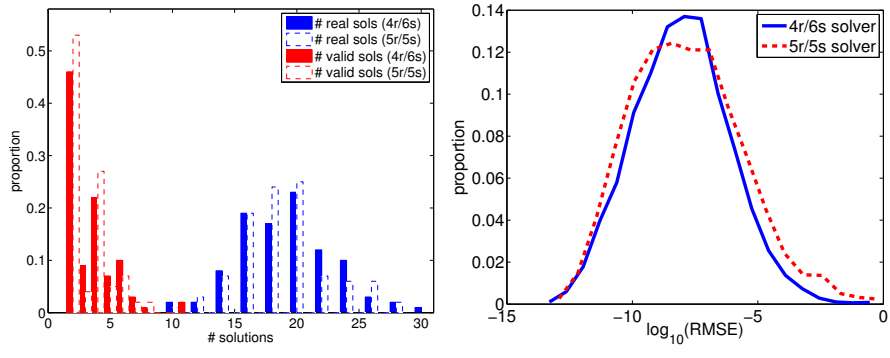


Figure 7.2: Synthetic experiment on 5000 noise-free random 3D TOA problems. Left: Distribution of the number of real and valid solutions among the 38 (or 42) solutions. Right: the error distribution (RMSE) of reconstructed positions of microphone and sound sources.

7.7.1 Synthetic Experiments

For synthetic data, we simulate the positions of receivers and transmitters as 2D or 3D points with independent Gaussian distribution of zero mean and identity covariance matrix. For noise-free synthetic data, we can see in Figure 7.2 (Right) that both the $4r/6s$ solver and $5r/5s$ solver are numerically stable. For 2D solver with $3r/3s$, we also compare our solver with the publicly available solver by Stewénius [92]. We can see in Figure 7.1 that, our solver is superior in numerical stability both in terms of average and worst case performance. To investigate sensitivity of the minimal solvers to noise, we also test $4r/6s$ and $5r/5s$ solvers on data under different relative magnitudes of noise and we observe that the solvers gives fairly good solutions under reasonable level of noise (Figure 7.3, left, solid lines). Using the solutions from the minimal solvers as initial solutions, we also apply nonlinear optimization step (Figure 7.3, dash lines). With respect to the failure cases, the two 3D solvers encounter relatively high count of failures when the noise level reaches 10^{-3}. Specifically, for different noise magnitudes at $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$, the failure percentages are approximately $\{0.4\%, 0.8\%, 2\%, 5\%, 20\%, 60\%\}$. We also test the solvers on over-determined cases with fixed noise level, $m = 10$ and varying n (Figure 7.3, right). Both minimal solvers give fairly good initial solutions. We can see that as n increases, the reconstruction errors after nonlinear optimization also become smaller.

As for the linear solvers based on SVD for slightly overdetermined cases, we also run these solvers on both noise-free and noisy data. We can see that (Figure 7.4) the linear solvers for $4r/ns$ (n>6) are fairly stable numerically. In our experiments, the solvers for $5r/ns$ (n>5) perform similarly. With respect to varying noise levels, the linear solvers show similar behaviors to the minimal cases with $5r/ns$ solvers perform slightly better than $4r/ns$ solvers as showed in Figure 7.5.

Regarding the speed of the minimal solvers, the current implementation in Matlab takes on average $800ms$, $300ms$ and $8ms$ for the $4r/6s$, $5r/5s$ and $3r/3s$ solvers, respectively, on a Macbook Air with 1.8 GHz Intel Core i5 and 8 GB memory. The most expensive part of the computation is the saturation step which involves a QR-factorization for a relative large matrix. One may replace the QR-factorization with other elimination methods e.g. LU-factorization. In our experiments, QR-factorization gives the best numerical stability with a tradeoff of slightly slower performance. The computation time can be improved by using efficient C implementation or utilizing schemes of optimizing polynomial solvers (e.g. [55]). For overdetermined solvers, the linear solvers run approximately $5ms$ since only a SVD on small coefficient matrices are performed.

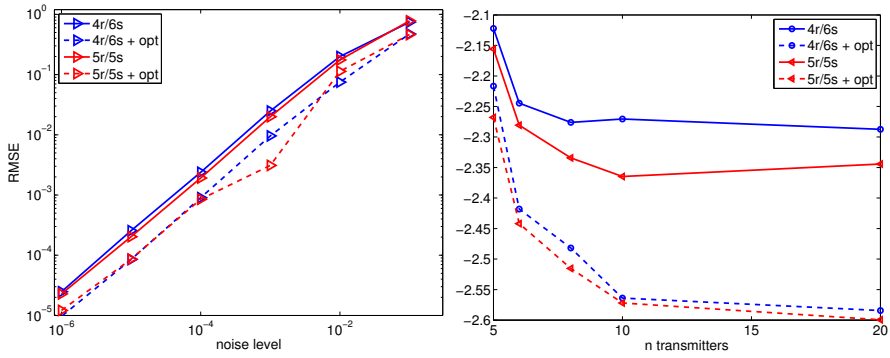


Figure 7.3: Performance of minimal solvers on noisy 3D synthetic TOA data - average errors (RMSE) of reconstructed positions of receivers and transmitters. Left: minimal solvers (4r/6s and 5r/5s) on corresponding minimal cases under varying noise magnitudes. Right: 10 receivers and varying number of transmitters under Gaussian noise with standard deviation 2×10^{-3}.

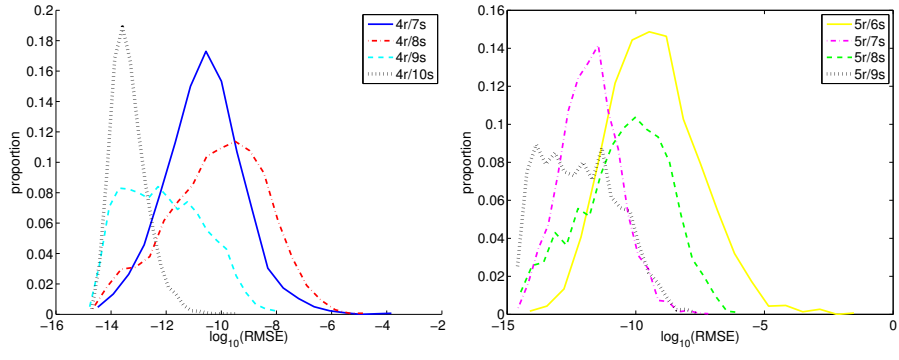


Figure 7.4: Performance of overdetermined linear solvers on 3D synthetic TOA data. Left: linear solvers $4r/ns$ (n>6) with corresponding setups on noise-free data Right: $5r/ns$ (n>5).

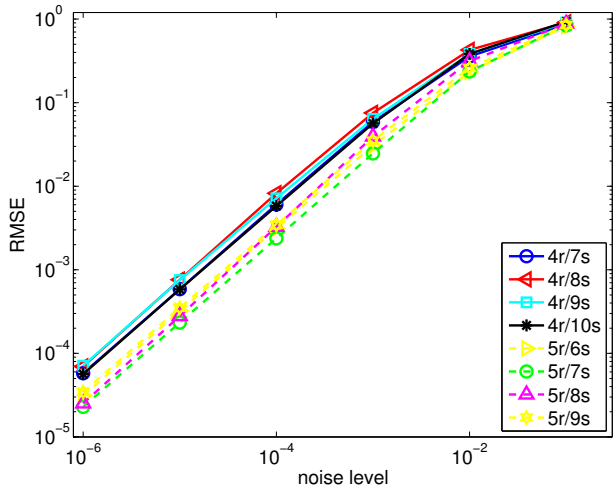


Figure 7.5: Performance of overdetermined linear solvers on 3D synthetic TOA data under varying noise magnitudes.

7.7.2 Real Experiments

For real experiments, we have first verify the solvers in a publicly available dataset [30]. In the dataset [30], the distances between the 8 microphones and 21 sounds are estimated based on the time-of-arrival measurements. The first microphone is assumed to be at the same location as first sound. For our formulation, no such assumption is needed. To verify this, we simply remove the distance measurements corresponding to the first microphone and first sound, which gives us a 7×20 matrix. For this reduced set of measurements, the root mean square errors (RMSE) of our reconstructed positions of microphones and sound sources after nonlinear iterative optimization are $0.0083m$ and $0.0108m$, respectively. This is similar to the accuracy [30] achieves ($0.0091m$ for microphones and $0.0111m$ for sound sources) with the additional assumption. For the full set of data (8 microphones and 21 sounds), our solvers also gives similar errors as in [30].

Another set of real data was obtained using seven T-bone MM-1 microphones and five Roxcore portable speakers, connected to a Fast Track Ultra 8R sound card in an indoor environment, with speakers and microphones placed in an approximate $1.5 \times 1.5 \times 1.5\ m^3$ volume (Figure 7.6, Left). TOA measurements were obtained by heuristically matching sounds from different speakers to sound flanks recorded from different microphones. To convert TOA measurements to distances, we assume the speed of sound during our measurement condition is $343\ m/s$. For this set of

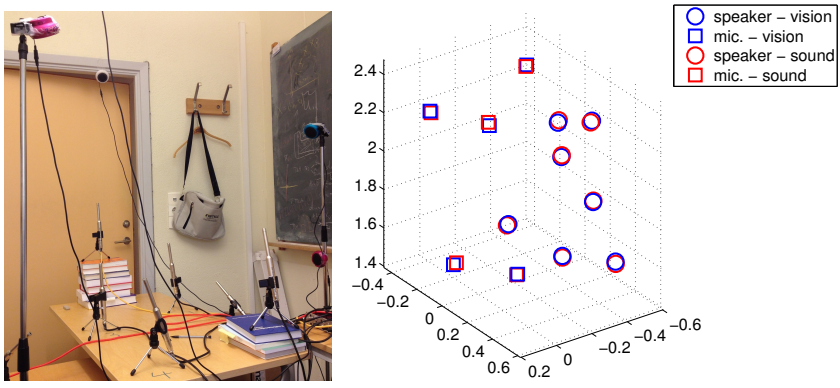


Figure 7.6: Real microphone and speaker calibration setup. Left: the setup of microphones and speakers in an office, Right: the reconstructed microphone positions using TOA measurements (red) aligned with the positions estimated based on computer vision (blue).

measurements, we also reconstructed the positions of the microphones using computer vision based algorithms as ground truth. The reconstruction based on sound (Figure 7.6, Right) when compared to the vision-based reconstruction has RMSE $0.0088m$ and $0.0131m$ for microphones and speakers respectively.

7.8 Conclusions

In this chapter, we completely characterize the TOA self-calibration problem. We first solve the minimal problems and identify the failure modes for receivers and transmitters with k-dimensional affine span. We have also studied a linear technique for solving over-determined cases. For the degenerate cases where there exists difference in the dimensions of receivers and transmitters, we derive a linear solver that reconstruct the projections of the sensor positions onto the lower-dimensional affine space. The theoretical results as well as practical algorithms described in this chapter should serve as a building block for solving robustly large-scale TOA sensor network calibration problem. On the other hand, as we will see in the next Chapter that the solution to the TOA self-calibration problem is also essential to understanding and solving the TDOA self-calibration problem.

Chapter 8

TDOA Self-Calibration

This chapter studies the problem of sensor network self-calibration from time-difference-of-arrival (TDOA) measurements. To solve this problem, a stratified approach is to first recover the unknown offsets and then solve the remaining TOA calibration problem. In this chapter, we focus on solving the unknown offsets and the methods discussed in Chapter 7 are used to solve the TOA self-calibration problem. Based on the rank constraints of the measurement matrix, we present two methods to estimate the unknown time offsets (i) a non-iterative algorithm that uses a set of rank constraints (ii) an iterative method suitable for overdetermined cases that seeks for minimal rank of the compacted matrix over the unknown offsets.

8.1 Background

Self-calibration of bipartite sensor networks using TDOA measurements is a nonlinear optimization problem, for which proper initialization is essential. The minimal problems for the complete 2D TDOA self-calibration have been studied in [92]. Given the difficulty of TDOA self-calibration problem, there is no existing non-iterative method to find all the solutions to the minimal problems in general affine space even for 2D cases. Several efforts to find all solution in a non-iterative manner has been made. A non-iterative linear solver based on matrix factorization was studied in [87], where solutions were given to non-minimal cases (e.g. 10 receivers, 5 transmitters in 3D). For 2D TDOA calibration, a recursive search algorithm was proposed in [98]. In [2], the minimal problems where all receivers lie on a line are solved for both TOA and TDOA. On the other hand, extensive works have been done on iterative methods. For instance, in [37], the rank constraint was explored and the unknown offsets are solved iteratively as a nonlinear alternating optimization with rank constraint enforced. In [81], an iterative method based on auxiliary functions is proposed to solve the full TDOA self-calibration problem.

8.2 Minimal Cases for Unknown Offsets

To recall, the TDOA self-calibration problem for bipartite networks is defined as

Problem 8.2.1 (TDOA Bipartite Network Calibration) Given relative distance measurements f_{ij}, determine receiver positions $\mathbf{r}_i$, $i = 1, \ldots, m$, transmitter positions $\mathbf{s}_j$, $j = 1, \ldots, n$ and offsets $o_j, j = 1, \ldots, n$ such that $f_{ij} = \|\mathbf{r}_i - \mathbf{s}_j\|_2 + o_j$.

In this section, we study two schemes to estimate the offsets $\{o_j\}$ based on the intrinsic *rank constraints* on the TDOA measurement matrix. With a slightly looser rank constraint, we first derive a linear factorization scheme to estimate the unknown offsets. By tightening the rank constraints, we arrive at several minimal problems for determine unknown offsets. In the following, we assume that both the m receivers $\mathbf{r}_i$ and n transmitters $\mathbf{s}_j$ lie in general affine space of dimension k. Note that if the offsets $\{o_j\}$ are known or recovered, the TDOA self-calibration problem can be converted to a TOA self-calibration problem i.e. $d_{ij} = f_{ij} - o_j = \|\mathbf{r}_i - \mathbf{s}_j\|_2$.

8.2.1 Rank-$(k+1)$ Constraint

In this section, we present a linear solver that utilizes rank-$(k+1)$ constraints of the TDOA measurement matrix. To illustrate this, we first rewrite the constraints into matrix form. From the basic measurement equation, we know that

$$(f_{ij} - o_j)^2 = (\mathbf{r}_i - \mathbf{s}_j)^T(\mathbf{r}_i - \mathbf{s}_j) = \mathbf{r}_i^T\mathbf{r}_i - 2\mathbf{r}_i^T\mathbf{s}_j + \mathbf{s}_j^T\mathbf{s}_j.$$

And we have

$$f_{ij} - 2o_{ij}^2 = \mathbf{r}_i^T\mathbf{r}_i - 2\mathbf{r}_i^T\mathbf{s}_j + (\mathbf{s}_j^T\mathbf{s}_j - o_j^2).$$

By constructing

$$\hat{\mathbf{r}}_i = \begin{bmatrix} 1 & \mathbf{r}_i^T & \mathbf{r}_i^T\mathbf{r}_i \end{bmatrix}^T$$

and

$$\hat{\mathbf{s}}_j = \begin{bmatrix} \mathbf{s}_j^T\mathbf{s}_j - o_j^2 & -2\mathbf{s}_j^T & 1 \end{bmatrix}^T,$$

we obtain

$$f_{ij}^2 - 2f_{ij}o_j = \hat{\mathbf{r}}_i^T\hat{\mathbf{s}}_j.$$

By collecting $\hat{\mathbf{r}}_i$ and $\hat{\mathbf{s}}_j$ into matrix $\mathbf{R} \in \hat{\mathbf{R}}^{(k+2)\times m}$ and $\hat{\mathbf{S}} \in \mathbb{R}^{(k+2)\times n}$, we have $\hat{\mathbf{F}} = \hat{\mathbf{R}}^T\hat{\mathbf{S}}$, where $\hat{\mathbf{F}} \in \mathbb{R}^{m\times n}$ with entries $\{f_{ij}^2 - 2f_{ij}o_j\}$. This suggests that matrix $\hat{\mathbf{F}}$ is at most of rank $k+2$ as we increase m and n. This is the rank-$(k+2)$ constraint used in [87]. Based on this constraint, in 3D, a linear solver requiring at least 10 receivers and 5 transmitters was derived.

We show that one can further reduce the rank-constraint to $k+1$ and reduce the required number of receivers by 1. The idea is to exploit the structure of $\hat{\mathbf{S}}$ - ones

in the last row. By multiplying $\hat{\mathbf{F}}$ from the right by a matrix $\mathbf{C}_n \in \mathbb{R}^{n\times(n-1)}$ of the form $[-\mathbb{1}_{n-1}\ \mathbf{I}_{n-1}]^T$ where $\mathbb{1}_{n-1} \in \mathbb{R}^{(n-1)\times 1}$ is a vector with all 1's and $\mathbf{I}_{n-1}$ is the $(n-1)\times(n-1)$ identity matrix. This operation subtracts the first column from each column of $\hat{\mathbf{S}}$ and results in a matrix with all zeros at the last row. Equivalently, this gives $\tilde{\mathbf{F}} = \hat{\mathbf{F}}\mathbf{C}_n = \tilde{\mathbf{R}}^T\tilde{\mathbf{S}}$, where $\tilde{\mathbf{F}} \in \mathbb{R}^{m\times(n-1)}$ with

$$\tilde{f}_{ij} = f_{i,j+1}^2 - f_{i1}^2 - 2f_{i,j+1}o_{j+1} + 2f_{i1}o_1$$

and

$$\tilde{\mathbf{r}}_i = \begin{bmatrix}1 & \mathbf{r}_i^T\end{bmatrix}^T,$$

$$\tilde{\mathbf{s}}_j = \begin{bmatrix}\mathbf{s}_{j+1}^T\mathbf{s}_{j+1} - o_{j+1}^2 - (\mathbf{s}_1^T\mathbf{s}_1 - o_1^2) & -2(\mathbf{s}_{j+1} - \mathbf{s}_1)^T\end{bmatrix}^T.$$

We can see that the columns of both $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{S}}$ are in $\mathbb{R}^{k+1}$. This effectively gives a constraint that the matrix $\tilde{\mathbf{F}}$ is at most of rank $k+1$. Let $\mathbf{A} \in \mathbb{R}^{m\times(n-1)}$ with $a_{ij} = f_{i,j+1}^2 - f_{i1}^2$, $\mathbf{B} \in \mathbb{R}^{m\times(n-1)}$ with $b_{ij} = -2f_{i,j+1}$, $\mathbf{c} \in \mathbb{R}^{m\times 1}$ with $c_i = 2f_{i1}$ and $\mathbf{e} = [o_1, \ldots, o_1]^T \in \mathbb{R}^{(n-1)\times 1}$ and $\mathbf{T} \in \mathbb{R}^{(n-1)\times(n-1)}$ is a diagonal matrix with $\{o_j\}_{j\geq 2}$ as diagonal entries. We have

$$\tilde{\mathbf{F}} = \tilde{\mathbf{R}}^T\tilde{\mathbf{S}} = \begin{bmatrix}\mathbf{A} & \mathbf{B} & \mathbf{c}\end{bmatrix}\begin{bmatrix}\mathbf{I}_{n-1}\\ \mathbf{T}\\ \mathbf{e}^T\end{bmatrix}. \tag{8.1}$$

Given that the first column of $\tilde{\mathbf{R}}^T$ are all ones, there exist $(k+1)$ columns of $\tilde{\mathbf{F}}$ whose linear combination forms a column of ones. If we choose $m = 2k+3$ ($\begin{bmatrix}\mathbf{A} & \mathbf{B} & \mathbf{c}\end{bmatrix}$ is of full rank) and $n = k+2$ ($\tilde{\mathbf{F}}$ is of rank $k+1$), we can find a unique solution for $\mathbf{u}$ to the following linear system

$$\tilde{\mathbf{F}}\mathbf{w} = \begin{bmatrix}\mathbf{A} & \mathbf{B} & \mathbf{c}\end{bmatrix}\underbrace{\begin{bmatrix}\mathbf{I}_{n-1}\\ \mathbf{T}\\ \mathbf{e}^T\end{bmatrix}\mathbf{w}}_{\mathbf{u}} = \begin{bmatrix}\mathbf{A} & \mathbf{B} & \mathbf{c}\end{bmatrix}\mathbf{u} = \mathbb{1}_{2k+3}. \tag{8.2}$$

Then we can recover the offsets $\{o_j\}$ from $\mathbf{u}$ as $o_1 = \frac{u_{2k+3}}{\sum_{j=1}^{k+1} u_j}$ and $o_j = \frac{u_{j+k}}{u_{j-1}}$ for $j = 2, ..., k+2$. Specifically, we need at least 9 receivers and 5 transmitter in 3D ($k = 3$) and at least 7 receivers and 4 transmitters in 2D ($k = 2$) to solve for the unknown offsets.

8.2.2 Rank-k Constraint

In this section, we further exploit the tighter rank-k constraints. This is done by utilizing the similar structure in $\tilde{\mathbf{R}}$. By multiplying $\tilde{\mathbf{F}}$ from the left with $\mathbf{C}_m =$

$m \setminus n$	4	5	6	7	8	9
5	-	-	-	-	-	I
6	-	-	I	-	III	-
7	-	I	III	-	-	-
8	-	-	-	-	-	-
9	-	III	-	-	-	-
10	I	II	-	-	-	-

$m \setminus n$	3	4	5	6
4	-	-	I	-
5	-	-	-	III
6	I	-	-	-
7	-	III	-	-
8	-	II	-	-
9	-	-	-	-

Table 8.1: Cases for TDOA problem for 3D (left) and 2D (right). (I) minimal cases, (II) solvable cases for [87] and (III) solvable cases based on methods in this chapter.

$[-\mathbb{1}_{m-1}\ \mathbf{I}_{m-1}]^T$, we have correspondingly $\bar{\mathbf{F}} = \mathbf{C}_m^T \hat{\mathbf{F}} \mathbf{C}_n = (\hat{\mathbf{R}}\mathbf{C}_m)^T (\hat{\mathbf{S}}\mathbf{C}_n)$. Here $\bar{\mathbf{F}} \in \mathbb{R}^{(m-1)\times(n-1)}$ and the entries $\bar{f}_{ij}$ of $\bar{\mathbf{F}}$ can be expressed with respect to the unknown $\{o_j\}$ as

$$\bar{f}_{ij} = g_{ij} - g_{0j} - g_{i0} + g_{00} \tag{8.3}$$

where $g_{ij} = f_{i+1,j+1}^2 - 2f_{i+1,j+1}o_{j+1}$. Given that the first row of $\hat{\mathbf{R}}\mathbf{C}_m$ and the last row of $\hat{\mathbf{S}}\mathbf{C}_n$ are all zeros, we have equivalently $\bar{\mathbf{F}} = \bar{\mathbf{R}}^T\bar{\mathbf{S}}$, where the columns of $\bar{\mathbf{R}} \in \mathbb{R}^{k\times(m-1)}$ and $\bar{\mathbf{S}} \in \mathbb{R}^{k\times(n-1)}$ are $\bar{\mathbf{r}}_i = \mathbf{r}_{i+1} - \mathbf{r}_1$ and $\bar{\mathbf{s}}_j = \mathbf{s}_{j+1} - \mathbf{s}_1$, respectively. It is clear that the matrix $\bar{\mathbf{F}}$ is at most of rank k. Therefore, given that each entry of $\bar{\mathbf{F}}$ is a function of the unknown offsets $\{o_1, \ldots, o_n\}$, we can solve for the offsets by enforcing these rank constraints on the sub-matrices of $\bar{\mathbf{F}}$. Specifically, all $(k+1)\times(k+1)$ sub-matrices of $\bar{\mathbf{F}}$ (if there exist any) will be rank-deficient and have rank k. This gives equivalently constraints on the determinants of the set of $(k+1)\times(k+1)$ sub-matrices Λ_{k+1} :

$$\det \mathbf{Q} = 0,\ \forall \mathbf{Q} \in \Lambda_{k+1}. \tag{8.4}$$

For a $(m-1)\times(n-1)$ matrix $\bar{\mathbf{F}}$ of rank k, there exist $N_c = |\Lambda_{k+1}| = \binom{m-1}{k+1}\binom{n-1}{k+1}$ constraints among which $(m-1-k)(n-1-k)$ constraints are linearly independent. Each constraint is a polynomial equation of degree $k+1$ in $\{o_1, \ldots, o_n\}$. For different choices of m and n, this system of polynomial equations can either be well-defined, overdetermined or underdetermined. To resolve this, we rely on algebraic geometry tools and make use of *Macaulay2* [39]. It turns out that there are several choices for m and n that produce well-defined and solvable polynomial systems. We summarize those cases and the number of solutions of the related polynomial systems for $k = 3$ and $k = 2$ in Table 8.1. In the following discussions, we denote the case with m receivers and n transmitters as mr/ns. Given these solvable

k	Rank	m	n	N_{sol}
3	5	10	5	1
3	4	9	5	1
3	3	7	6	5
3	3	6	8	14
2	4	8	4	1
2	3	7	4	1
2	2	5	6	5

Table 8.2: Number of solutions to the polynomial systems given by constraints on unknown offsets for different cases (3D and 2D).

cases, we have developed numerically stable polynomial solvers for the unknown offsets. Note that we can verify the number of solutions and sufficiency for rank-$(k+2)$ and rank-$(k+1)$ constraints in a similar manner. For those cases, the algebraic tools indicate that all those cases have only 1 solution. For example, the two cases with only 1 solution: $9\text{r}/5\text{s}$ in 3D and $7\text{r}/4\text{s}$ in 2D correspond to the linearly solvable cases discussed in Section 8.2.1.

8.2.3 Discussions

One could say that we are using necessary rank constraint on the calibrated matrix $\mathbf{D}$ with entries $d_{ij}^2 = (f_{ij} - o_j)^2$ to determine the offsets. Notice, however, the rank constraint is necessary, but not sufficient condition on $\mathbf{D}$ coming from TOA measurements. For instance, although $7\text{r}/6\text{s}$ is a minimal case for determining the offsets from the $\text{rank}(\mathbf{C}_m^T \hat{\mathbf{F}} \mathbf{C}_n) = 3$, the remaining TOA problem after we compensate for the offsets is actually overdetermined, where the minimal problems for TOA is $4\text{r}/6\text{s}$, $6\text{r}/4\text{s}$ and $5\text{r}/5\text{s}$ (Chapter 7). From Table 8.1, it can be seen that the solvers based on tighter constraints in this chapter yield cases that are closer to the minimal cases for the TDOA self-calibration problem.

8.3 Rank Optimization

In this section, we study an iterative method for determining the unknown offsets. This method is suitable for overdetermined cases as well as measurements with missing data. The idea is to utilize the rank-k constraint on the measurement matrix and to formulate a rank-constrained optimization problem with respect to unknown offsets.

To start with, we know from Section 8.2 that the modified measurement matrix $\bar{\mathbf{F}}$

is of rank k. When we have an overdetermined case, we can formulate the following optimization problem for finding the unknown offsets $\mathbf{o} = \{o_j\}$

$$\begin{aligned} \min_{\mathbf{o},\mathbf{X}} \quad & \|\mathbf{X}\|_*, \\ \text{s.t.} \quad & \mathbf{B}_0 + \sum_{j=1}^{n} o_j \mathbf{B}_j = \mathbf{X}. \end{aligned} \tag{8.5}$$

Here $\|.\|_*$ is the nuclear norm of a matrix $\mathbf{X}$ defined as $\|\mathbf{X}\|_* = \sum_{i=1}^{\min(m,n)} \sigma_i$, where σ_i is the ith singular value of a matrix X in $\mathbb{R}^{m\times n}$. $\mathbf{B}_0, \ldots, \mathbf{B}_n$ are constant matrices in $\mathbb{R}^{(m-1)\times(n-1)}$ derived from (8.3). Specifically, $\mathbf{B}_0 = \mathbf{C}_m^T \hat{\mathbf{F}} \mathbf{C}_n$, $\mathbf{B}_1 = [\mathbf{b}_1, \ldots, \mathbf{b}_1]$ and $\mathbf{B}_j$ $(j \geq 2)$ is a matrix with $(n-2)$ columns of zeros and the $(j-1)^{th}$ column is –$\mathbf{b}_j$ where $\mathbf{b}_j = [2(f_{2j} - f_{1j}) \ \ldots \ 2(f_{m,1} - f_{1j})]^T$.

Due to the existence of noise in real measurements, we relax the strict equality constraints on the measurements as

$$\min_{\mathbf{o},\mathbf{X}} \ \|\mathbf{X}\|_* + \frac{\mu}{2}\|\mathbf{B}_0 + \sum_{j=1}^{n} o_j \mathbf{B}_j - \mathbf{X}\|_{\mathcal{F}}^2, \tag{8.6}$$

where μ is some positive constant scalar parameter and $\|.\|_{\mathcal{F}}$ is the Frobenius norm. When minimizing the nuclear norm of $\mathbf{X}$, instead of getting a low rank approximation of $\mathbf{X}$, one might minimize the singular values evenly[1]. To avoid this, we apply the same strategy in [102] called *truncated nuclear norm*, where the sum of the $(\min\{m,n\} - k)$ smallest singular values is minimized. To achieve this, the minimization problem in (8.6) is modified as

$$\begin{aligned} \min_{\mathbf{o},\mathbf{X},\mathbf{U},\mathbf{V}} \ & \|\mathbf{X}\|_* - \mathrm{Tr}(\mathbf{U}\mathbf{X}\mathbf{V}^T) \\ & + \frac{\mu}{2}\|\mathbf{B}_0 + \sum_{j=1}^{n} o_j \mathbf{B}_j - \mathbf{X}\|_{\mathcal{F}}^2. \end{aligned} \tag{8.7}$$

where $\mathbf{U} \in \mathbb{R}^{k\times m}$, $\mathbf{V} \in \mathbb{R}^{k\times n}$, and $\mathbf{U}\mathbf{U}^T = \mathbf{I}_k$ and $\mathbf{V}\mathbf{V}^T = \mathbf{I}_k$. It was shown in [102] that adding the term $-\mathrm{Tr}(\mathbf{U}\mathbf{X}\mathbf{V}^T)$ in the minimization is equivalent to minimizing the truncated nuclear norm. Now (8.7) is a non-convex optimization problem and an optimization scheme for obtaining local minima is presented in the next section.

8.3.1 Optimization Scheme

In the section, we discuss the optimization scheme for the proposed problems. We use a two-step iterative scheme as in [102]. In Step 1, we fix $\mathbf{o}$, $\mathbf{X}$ and solve for $\mathbf{U}$ and

[1]This is indeed what happens for this problem in our initial implementation without the term $-\mathrm{Tr}(\mathbf{U}\mathbf{X}\mathbf{V}^T)$)

$\mathbf{V}$ as an outer loop and in Step 2, with $\mathbf{U}, \mathbf{V}$ fixed, we optimize over $\mathbf{o}, \mathbf{X}$ iteratively in an inner loop until $\mathbf{X}$ converges. The first step is relatively simple (as shown in Algorithm 8.3.1) and we will discuss in details the optimization in Step 2.

Algorithm 8.3.1 Nuclear Norm Regularization for TDOA Unknown Offsets

Input: TDOA measurements $\{f_{ij}\}$ of m receivers and n transmitters, threshold ϵ :
Output: The offsets $\mathbf{o} = \{o_1, \ldots, o_n\}$.
Initialize: Construct $\{\mathbf{B}_j\}$, $j = 0, \ldots, n$ based on (8.3), set $\mathbf{o} = \mathbf{0}$, $\mathbf{X}_1 = \hat{\mathbf{X}}_1 = \mathbf{Y} = \hat{\mathbf{F}}$
Repeat at iteration k:

1. Solve for $\mathbf{U}_k$ and $\mathbf{V}_k$ given $\mathbf{X}_k$.
   (a) $(\mathbf{A}_k, \mathbf{\Sigma}_k, \mathbf{C}_k) = \mathbf{svd}(\mathbf{X}_k)$
   where $\mathbf{A}_k = (\mathbf{a}_1, \ldots, \mathbf{a}_{m-1}) \in \mathbb{R}^{(m-1)\times(m-1)}$ and $\mathbf{C}_k = (\mathbf{c}_1, \ldots, \mathbf{c}_{n-1}) \in \mathbb{R}^{(n-1)\times(n-1)}$
   (b) $\mathbf{U}_k = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3)^T$, $\mathbf{V}_k = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)^T$
2. Solve $\{\mathbf{X}_{k+1}, \mathbf{o}_{k+1}\} = \arg\min_{\mathbf{o},\mathbf{X}} \|\mathbf{X}\|_* - \mathrm{Tr}(\mathbf{U}_k\mathbf{X}\mathbf{V}_k^T) + \frac{\mu}{2}\|\mathbf{B}_0 + \sum_{j=1}^n o_j\mathbf{B}_j - \mathbf{X}\|_{\mathcal{F}}^2$

Until: $\sqrt{\|\mathbf{X}_{k+1} - \mathbf{X}_k\|_{\mathcal{F}}^2 + \|\mathbf{o}_{k+1} - \mathbf{o}_k\|_{\mathcal{F}}^2} < \epsilon$

Optimization using ADMM

To optimize w.r.t $\mathbf{X}, \mathbf{o}$ with $\mathbf{U}, \mathbf{V}$ fixed in Step 2, we use the *alternating direction methods of multipliers* (ADMM) [8]. First, by introducing a new variable $\hat{\mathbf{X}}$, we rewrite (8.6) as

$$\begin{aligned} \min_{\mathbf{o},\mathbf{X},\hat{\mathbf{X}}} \quad & \|\mathbf{X}\|_* - \mathrm{Tr}(\mathbf{U}\hat{\mathbf{X}}\mathbf{V}^T) \\ & + \frac{\mu}{2}\|\mathbf{B}_0 + \sum_{j=1}^n o_j\mathbf{B}_j - \hat{\mathbf{X}}\|_{\mathcal{F}}^2, \\ \text{s.t.} \quad & \mathbf{X} = \hat{\mathbf{X}}. \end{aligned} \tag{8.8}$$

We can see that the augmented Lagrange function of (8.8) is as follows

$$L(\mathbf{o}, \mathbf{X}, \hat{\mathbf{X}}, \mathbf{Y}) = \|\mathbf{X}\|_* - \mathrm{Tr}(\mathbf{U}\hat{\mathbf{X}}\mathbf{V}^T) + \frac{\mu}{2}\|\mathbf{B}_0 + \sum_{j=1}^{n} o_j\mathbf{B}_j - \hat{\mathbf{X}}\|_{\mathcal{F}}^2 + \frac{\lambda}{2}\|\mathbf{X} - \hat{\mathbf{X}}\|_{\mathcal{F}}^2 + \mathrm{Tr}\left((\mathbf{Y}^T(\mathbf{X} - \hat{\mathbf{X}})\right), \tag{8.9}$$

where λ is a positive scalar. With the schemes in ADMM, we alternate the optimization on a subset set of variables by fixing the rest of the variables. Specifically, starting with initial values that $\mathbf{o}_1 = \mathbf{0}$, $\mathbf{X}_1 = \hat{\mathbf{X}}_1 = \mathbf{Y} = \hat{\mathbf{F}}$, we have the following three iterative steps for iteration $k+1$:

Computing $\mathbf{X}_{k+1}$: Given $\mathbf{o}_k$, $\hat{\mathbf{X}}_k$, and $\mathbf{Y}_k$, we minimize $L(\mathbf{o}_k, \mathbf{X}, \hat{\mathbf{X}}_k, \mathbf{Y}_k, \lambda)$ over $\mathbf{X}$,

$$\mathbf{X}_{k+1} = \mathrm{argmin}_{\mathbf{X}}\Big(\|\mathbf{X}\|_* - \mathrm{Tr}(\mathbf{U}\hat{\mathbf{X}}_k\mathbf{V}^T) + \frac{\mu}{2}\|\mathbf{B}_0 + \sum_{j=1}^{n} o_{j\,k}\mathbf{B}_j - \hat{\mathbf{X}}_k\|_{\mathcal{F}}^2 + \frac{\lambda}{2}\|\mathbf{X} - \hat{\mathbf{X}}_k\|_{\mathcal{F}}^2 + \mathrm{Tr}\left(\mathbf{Y}_k^T(\mathbf{X} - \hat{\mathbf{X}}_k)\right)\Big), \tag{8.10}$$

which is equivalent to the following by ignoring the constants

$$\mathbf{X}_{k+1} = \arg\min_{\mathbf{X}}\left(\|\mathbf{X}\|_* + \frac{\lambda}{2}\|\mathbf{X} - (\hat{\mathbf{X}}_k - \frac{1}{\lambda}\mathbf{Y}_k)\|_{\mathcal{F}}^2\right). \tag{8.11}$$

This can be solved via Singular Value Thresholding theorem [23].

Computing $\mathbf{o}_{k+1}, \hat{\mathbf{X}}_{k+1}$: Fix $\mathbf{X}_{k+1}$ and $\mathbf{Y}_k$, we can calculate $\mathbf{o}_{k+1}$ and $\hat{\mathbf{X}}_{k+1}$ as follows:

$$\{\mathbf{o}_{k+1}, \hat{\mathbf{X}}_{k+1}\} = \arg\min_{\mathbf{o},\hat{\mathbf{X}}}\big(\frac{\mu}{2}\|\mathbf{B}_0 + \sum_{j=1}^{n} o_j\mathbf{B}_j - \hat{\mathbf{X}}\|_{\mathcal{F}}^2 + \frac{\lambda}{2}\|\mathbf{X}_{k+1} - (\hat{\mathbf{X}} - \frac{1}{\lambda}\mathbf{Y}_k)\|_{\mathcal{F}}^2\big), \tag{8.12}$$

which is sum of two quadratic functions and can be solved by finding $\{\mathbf{o}, \hat{\mathbf{X}}\}$ such that $\partial L(\mathbf{o}, \mathbf{X}_{k+1}, \hat{\mathbf{X}}, \mathbf{Y}_k) = 0$.

Computing $\mathbf{Y}_{k+1}$: $\mathbf{Y}$ can be updated as

$$\mathbf{Y}_{k+1} = \mathbf{Y}_k + \lambda(\mathbf{X}_{k+1} - \hat{\mathbf{X}}_{k+1}). \tag{8.13}$$

We iterate the steps described above in an inner loop until $\mathbf{X}$ converges. Then we feed the updated $\mathbf{X}$ to the outer loop as in Step 1 of the algorithm to update $\mathbf{U}$, $\mathbf{V}$. The outer loop runs until $\sqrt{\|\mathbf{X}_{l+1} - \mathbf{X}_l\|_{\mathcal{F}}^2 + \|\mathbf{o}_{l+1} - \mathbf{o}_l\|_{\mathcal{F}}^2} < \epsilon$ or a maximum number of iterations is reached.

Missing Data The previous iterative steps can be easily modified for TDOA measurements with missing data. In this case, we only need to replace the term in the above $\|\mathbf{B}_0 + \sum_{j=1}^{n} o_j \mathbf{B}_j - \hat{\mathbf{X}}\|_{\mathcal{F}}^2$ by $\|\mathbf{B}_0 + \sum_{j=1}^{n} o_j \mathbf{B}_j - \hat{\mathbf{X}}\|_{\mathcal{F},\Omega}^2$, where Ω is set of measurements that have been observed[2]. This modification will only change slightly the iterative updating step for $\mathbf{o}_{k+1}, \hat{\mathbf{X}}_{k+1}$ and the other steps remain the same.

8.4 Solving TDOA Self-Calibration

We can combine the steps for unknown offsets and the TOA problem to solve the full TDOA problem. To this end, we have devised a set of close-to-minimal solvers as well as an iterative method to solve for unknown offsets TDOA self-calibration problem. For overdetermined measurements, we can apply similar strategy as incremental structure from motion in computer vision. One starts by choosing m^* receivers that have largest number of correspondences from n^* transmitters and solves for the offsets. Then the offsets of remaining transmitters can then be solved incrementally with least square followed by also a nonlinear optimization to refine the solution. We can then recover the positions of the chosen receivers and transmitters. The positions of remaining receivers and transmitters are calculated by trilateration e.g. [24]. In the presence of outliers, our proposed solvers can be utilized for robust fitting technique e.g. RANSAC. The parameters obtained can then be used as initial estimates to local optimization of the nonlinear least squares

$$\min_{\mathbf{r}_i, \mathbf{s}_j, o_j} \sum_{ij} \left(f_{ij} - (\|\mathbf{r}_i - \mathbf{s}_j\|_2 + o_j) \right)^2$$

using standard techniques (Levenberg-Marquart) in order to obtain the maximal likelihood estimate of the parameters.

8.5 Experiments

8.5.1 Synthetic Data

In this section, we study the numerical behaviors of the TDOA solvers on synthetic data. We simulate the positions of receivers and transmitters as 3D points with independent Gaussian distribution of zero mean and identity covariance matrix. We generate the offsets randomly with independent Gaussian distribution with zero mean and standard deviation 10. We study the effects of zero-mean Gaussian noise on the solvers, where we vary the standard deviation of the Gaussian noise added to the

[2]Here we can assume that there is one receiver that has complement measurements of the n transmitters, and one transmitter that is measured by all m receivers.

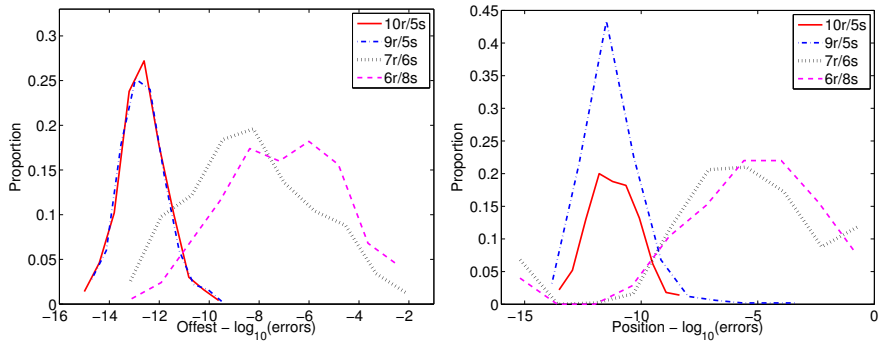


Figure 8.1: Numerical stability of different solvers on 500 noise-free synthetic examples. Errors compared to ground truth are shown (left) Offsets (right) Positions of the sensors.

TDOA measurements. When solving TOA problem, we have used the methods presented in the Chapter 7. To compare the reconstructed positions of receivers and transmitters with the true positions, we rotate and translate the coordinate system and register the solutions with the ground truth positions, accordingly.

Minimal Solvers

We first study the numerical stability of the solvers on noise-free synthetic data. From Figure 8.1, we can see that both linear solvers $10r/5s$ and $9r/5s$ perform very well in terms stability. The solvers $7r/6s$ and $6r/8s$ are stable enough but are not as good as the linear ones. This is mainly due to the high degree and large number of unknowns in the corresponding polynomial systems.

We then study the effects of zero-mean Gaussian noise on the solvers. In this experiment, we vary the standard deviation of the Gaussian noise added to the TDOA measurements. We can see from Figure 8.2 that, our proposed solvers for 3D space i.e. $9r/5s$, $7r/6s$ and $6r/8s$ give numerically similar results as the $10r/5s$ case in [87] for both minimal settings and overdetermined cases ($m = 20, n = 20$). Specifically, we also observe that for noise level lower then 10^{-3}, the errors of the estimated offsets and positions are approximately the same scale as the noise level. As the noise level increases, the errors of the initial estimates from the all the solvers are slightly larger.

Rank Optimization

First of all, we study the convergence of the algorithm for noise free cases. For well-constrained over-determined cases, we can see that in Figure 8.3 (left) that the ADMM

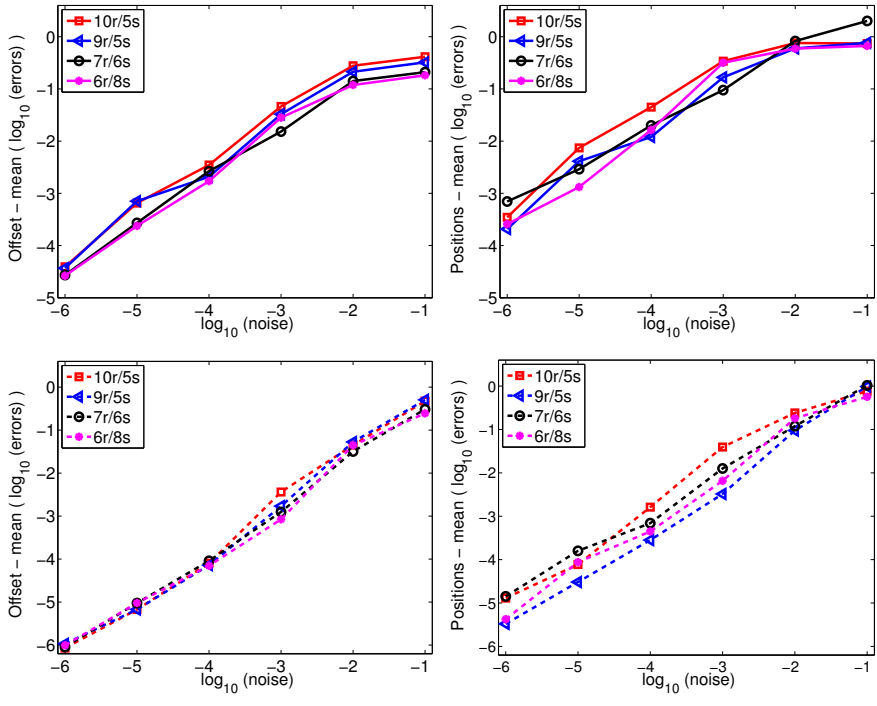


Figure 8.2: Synthetic experiments for TDOA solvers on 3D under Gaussian noise. The errors of estimated time offsets (left) and reconstructed positions of microphones and sounds (right) are shown. (Top) Performance of different solvers (10r/5s [87], 9r/5s, 7r/6s and 6r/8s) with their corresponding minimal settings for solving offsets; (Bottom) with 20 receivers and 20 microphones.

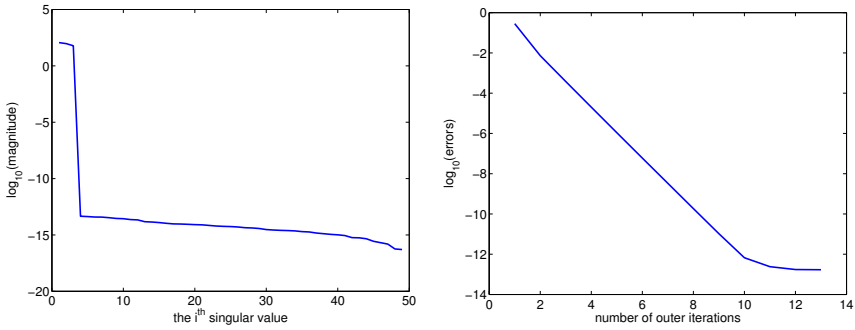


Figure 8.3: Synthetic TDOA measurements with no noise ($m = n = 50$). Left: Singular values of the matrix $\mathbf{X}$ after optimization; Right : Speed of convergence ($\|\mathbf{o} - \mathbf{o}_{gt}\|_{\mathcal{F}}$) for the ADMM algorithm ($\mu = 10,\ \lambda = 1, \epsilon = 10^{-12}$).

algorithm converges to a rank-3 matrix $\mathbf{X}$ and the ground truth offsets $\mathbf{o}_{gt}$ in a few outer iterations.

To further understand the effects of increasing number of measurements, we run the algorithm on noise-free data for different fixed number of receivers m and varying number of transmitters n. For this experiment, we set $\epsilon = 10^{-12}$ and the maximum number of iterations to 5000. We observe that in Figure 8.4 (left) for small m e.g. $m = 5$, the method does not converge to a reliable initial solution for the offsets. This can be related to results for minimal cases for offsets in 3D (see Table 8.1) where at least 6 receivers are needed to recover the unknown offsets. For larger m's ($m \geq 6$), we can see that the relative errors of recovered offsets decrease as we have more transmitters, suggesting the benefits of increasing number of measurements. Due to the fact that the function we are optimizing are non-convex, over-constrained case with more measurements might reduce the number of local minima, thus gives better convergence behavior for random initialization. It is also noted that for a fixed m, when n is larger than 20, the convergence of the method does not change much.

We also test the method under different levels of noise with increasing number of measurements. We see that in Figure 8.4 (right) the errors of the recovered offsets decrease as the number of measurement increases. Up to certain number of measurements, we see that the method does not give better performance e.g. $m = n = 30$ gives similar errors as $m = n = 100$. It is also observed that when the noise level reaches standard deviation of 10^{-1}, the method performs poorly no matter how much the number of measurements increases.

While the previous experiments assume that measurements are complete, it would be of interest to see how the algorithm performs with missing data. In this experiment, we run the algorithm on noise-free synthetic TDOA examples where we randomly

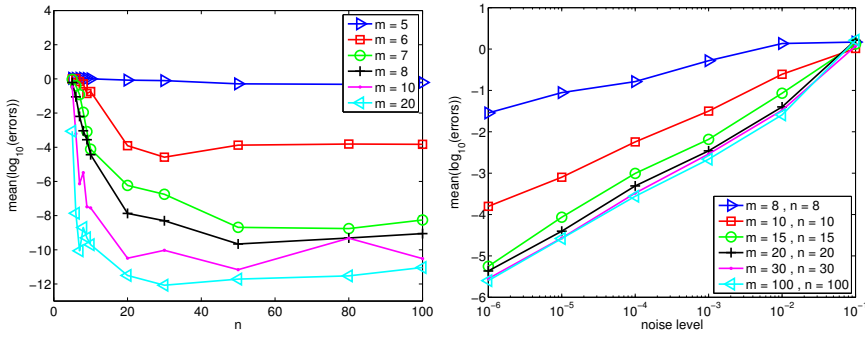


Figure 8.4: Synthetic experiments - average errors $\|\mathbf{o} - \mathbf{o}_{gt}\|_{\mathcal{F}} / \|\mathbf{o}_{gt}\|_{\mathcal{F}}$ on 100 random synthetic TDOA measurements. Left: Noise-free data - varying n for different fixed m; Right: Noisy data - varying m and n for different levels of noise

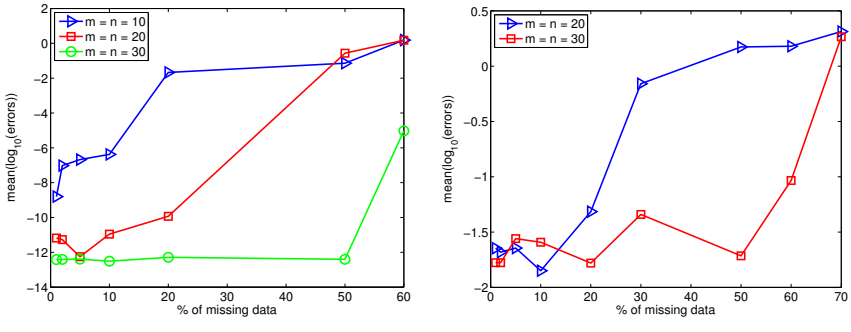


Figure 8.5: Synthetic TDOA measurements with varying percentage of missing data. Left : noisy-free data for different m and n; Right : noisy data with Gaussian noise of standard deviation 10^{-2}.

remove certain percentage of entries in the measurements. We vary the percentage of the missing entries in the measurement matrix, we can see that in Figure 8.5 for relatively large m and n, the algorithm only breaks down when more than 50% of the measurements are missing ($m = n = 30$). For smaller m and n ($m = n = 10$), the method fails to provide good estimate when there are more than 20% missing entries. For noisy measurements with missing data, with noise level as 10^{-2}, which is typical in practice, for $m = n = 20$, the algorithm works well until there are more than 30% missing data (Figure8.5, right). For cases that have more constraints i.e. $m = n = 30$, the algorithm is more robust to increasing number of missing entries (up to 50%) and it gives fairly good initial guess of the offsets even under noise.

8.5.2 Real Data

To collect real TDOA data, we work with sound signals and microphones. We placed 8 synchronized microphones (Shure SV100) recorded at 44.1kHz in an office. They are approximately 0.3-1.5 meters away from each other and placed in a non-planar fashion. We connected them to an audio interface (M-Audio Fast Track Ultra 8R), which was connected to a computer. We generated sounds by moving around in the room and clapping approximately 1-2 meters from the microphones. We collected 5 independent recordings of approximately 20s. Each recording contains roughly 30 claps (transmitters).

To obtain TDOA measurements, we coarsely matched sounds of the claps to sound flanks (edges between periods with low energy and periods with high energy) recorded from different microphones. For the experiment, we used only those claps that were detected in all 8 channels. We run both the $7\text{r}/6\text{s}$ and $6\text{r}/8\text{s}$ solvers to determine the offsets followed by an alternating optimization as in [87] that refines the offset estimation. After calibrating the TDOA measurements with the unknown offsets, we recover an initial euclidean reconstruction for the locations of microphones and claps using the TOA solvers described in the last Chapter. Finally we refine the reconstruction with nonlinear optimization. The result of one of these 5 reconstructions are shown in Figure 8.6 (middle). The reconstructed microphone positions from these 5 independent multi-channel recordings were put in a common coordinate system and compared to each other. The average distance from each microphone to its corresponding mean position (estimated from corresponding reconstruction of the 5 recordings) is 2.60 cm. It is important to point out that without proper initialization using our methods, the solutions we get converge poorly (with large reconstruction errors). Previous solvers do not work here due to either insufficient number of receivers (10 receivers needed in [87]) or violating the assumption that one of the microphones collocates with one of the claps [31] .

As an additional evaluation, we have also reconstructed the locations of the microphones based on computer vision techniques. We took 11 images of the experimental

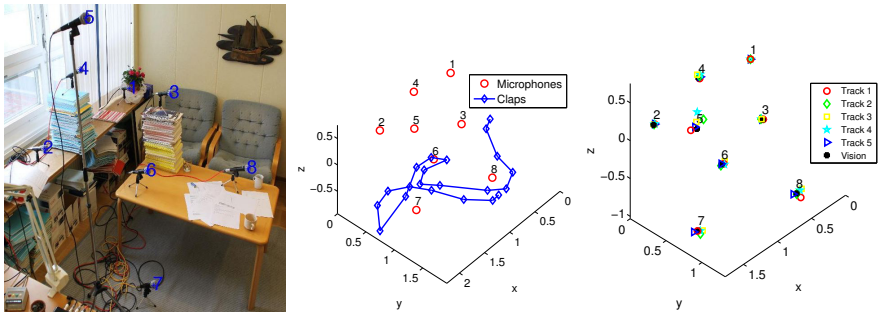


Figure 8.6: Results on TDOA with microphones and sounds. Left : Reconstruction of microphone (8, red - 'o') and sound (21, blue - '◇') positions for one the 5 recordings; Right : Reconstructed microphone positions estimated from 5 different tracks of TDOA measurements and the corresponding reconstruction from computer vision (black - '+')

setup. Figure 8.6 (left) shows one of the 11 images used. We manually detected the 8 microphone center positions in these 11 images and used standard structure from motion algorithms to estimate the positions of the 8 microphones. The resulting reconstruction is also compared to that of the five structure-from-sound reconstructions. The comparison is shown in Figure 8.6 (right). We can see the TDOA-based reconstructions are consistent with the vision-based reconstruction.

8.6 Conclusions

In this chapter we have studied the sensor network self-calibration problem in the time-difference-of-arrival (TDOA) setting, where only relative distances between the transmitters and receivers are given. To solve for the unknown time offsets, we utilize the rank constraints on the measurements. We have derived several non-iterative solvers that require minimal information on the measurements. We also have studied an iterative scheme with nuclear norm regularizer.

There are several interesting avenues of future research. Although, the presented non-iterative solvers improve on the state-of-the-art, the minimal cases for bipartite TDOA self-calibration have not been solved. It would be interesting to solve these cases, to study the failure modes, both generic failure modes (or critical configurations) for the generic problem, and also if there are additional failure modes of the presented algorithms. For the iterative method, it is of interest to improve the robustness by using $\mathbf{L}_1$-norm $\|\mathbf{B}_0 + \sum_{j=1}^{n} o_{j\,k}\mathbf{B}_j - \hat{\mathbf{X}}_k\|_1$ in the cost function. An extension to optimization for UTOA problems will also enable wider application of the scheme.

Chapter 9

Far-Field TDOA

This chapter presents a study of the far-field approximation to sensor network self-calibration from TDOA measurements. The problem can be present in many applications e.g. calibration of microphone, WiFi-transmitter arrays and cellular base stations. In the far-field approximation, we assume that the relative motion of the receiver or the inter-distances between different receivers are small in comparison to their distances to the transmitters. In far-field setup, we aim to reconstruct the positions or motions of the receivers and the directions from the transmitters to the receivers. We derive a linear method to solve the minimal problems in general dimensions and study the failure modes for the algorithms. For overdetermined cases, the linear method is also applicable to provide an initial solution to the refinement step using nonlinear optimization.

9.1 Far-Field TDOA Self-Calibration

We consider the far-field setup that the receivers are moving or placed in a small area, and the transmitters are far away from the receivers. A moving receiver can be seen as virtual receivers at varying positions. In the following discussion, we make no difference between real and virtual receivers or transmitters. To model this formally, we assume that two receivers at positions $\mathbf{r}_0$ and $\mathbf{r}$, and a transmitter at location $\mathbf{s}$. We assume that both the receivers and transmitters have affine span of dimension k i.e. $\mathbf{r} \in \mathbb{R}^k$ and $\mathbf{s} \in \mathbb{R}^k$. For far-field setup, we have the distances $\|\mathbf{r}_0 - \mathbf{s}\|_2 \gg \|\mathbf{r} - \mathbf{r}_0\|_2$ and $\|\mathbf{r} - \mathbf{s}\|_2 \gg \|\mathbf{r} - \mathbf{r}_0\|_2$. If we assume that the direction $\mathbf{n}$ from $\mathbf{s}$ to the two receivers is parallel (see also Section 2.2.1), we have the following approximation for the distance between $\mathbf{r}$ and $\mathbf{s}$

$$\begin{aligned} d &= \|\mathbf{r} - \mathbf{s}\|_2 \\ &\approx \|\mathbf{r}_0 - \mathbf{s}\|_2 + (\mathbf{r} - \mathbf{r}_0)^T \mathbf{n}_j \\ &= \mathbf{r}^T \mathbf{n} + \underbrace{(\|\mathbf{r}_0 - \mathbf{s}\|_2 - \mathbf{r}_0^T \mathbf{n})}_{\hat{o}} \end{aligned} \tag{9.1}$$

Here $\mathbf{n}$ is the direction vector with unit length. For TDOA measurements, we have

$$f = \|\mathbf{r} - \mathbf{s}\|_2 + \tilde{o},$$

where $\tilde{o}$ is the unknown offset related to unknown clock of the transmitter. By setting $o = \tilde{o} + \hat{o}$, one obtains the far-field approximation for TDOA measurements

$$f \approx \mathbf{r}^T \mathbf{n} + o.$$

Note that the offset o is only related to the transmitter and the reference position $\mathbf{r}_0$. To put it more formally, the far-field TDOA (FFTDOA) problem that arises from this approximate relative distance measurement is the following

Problem 9.1.1 Given measurements $f_{ij},\ i = 1, \ldots, m$ and $j = 1, \ldots, n$ from the receiver at m different positions to n transmitters, determine both the positions $\mathbf{r}_i$ of the receivers and the directions $\mathbf{n}_j$ from the transmitters so that for each (i, j) pair:

$$\begin{aligned} f_{ij} = \quad & \mathbf{r}_i^T \mathbf{n}_j + o_j \\ s.t. \quad & \|\mathbf{n}_j\|_2 = 1 \end{aligned} \tag{9.2}$$

where o_j is a constant distance offset for transmitter i.

From the formulation above, we can see that unknowns offsets exist for both FFTOA and FFTDOA problems, thus the TOA and TDOA problems are equivalent in far-field setting. In the following, we will focus on the discussions in terms of FFTDOA.

Lemma 9.1. *A problem with m measurements to n transmitters with unknown offset o_j can without loss of generality be converted to a problem with $m - 1$ measurements to n transmitters with known offset.*

Proof. Because of the unknown constant o_j, the problem does not change in character by modification $\bar{f}_{ij} = f_{i+1,j} - c_j$ where c_j is an arbitrary constant. If we set $c_j = f_{1,j}$, then

$$\begin{aligned} \bar{f}_{ij} \quad &= f_{i+1,j} - f_{1,j} \\ &= (\mathbf{r}_{i+1} - \mathbf{r}_1)^T \mathbf{n}_j \end{aligned}$$

By also setting $\mathbf{r}_1 = \mathbf{0}$ which is equivalent to choosing the first receiver's position $\mathbf{r}_1$ to be at the origin to fix the gauge freedom of the unknown coordinate system, we get $\bar{f}_{ij} = \mathbf{r}_{i+1}^T \mathbf{n}_j$. This is essentially a problem with known offsets for $m - 1$ receivers. □

For the original problem, this is equivalent to setting $o_j = f_{1j}$. For simplicity we will in the sequel assume that $o_j = 0$ and assume that the measurements of one of receivers has already been used to resolve the ambiguity. With the modified measurement $\bar{f}_{ij} = f_{i+1,j} - f_{1,j}$, we have

Problem 9.1.2 Given measurements $\bar{f}_{ij},\ i = 1, \ldots, m$ and $j = 1, \ldots, n$ from the receiver at m different positions to n transmitters, determine both the positions $\mathbf{r}_i$ of the receiver during the relative motion and the direction from the transmitters $\mathbf{n}_j$ so that

$$\begin{aligned} \bar{f}_{i,j} = &\quad \mathbf{r}_i^T \mathbf{n}_j \\ s.t. &\quad \|\mathbf{n}_j\|_2 = 1 \end{aligned} \tag{9.3}$$

Lemma 9.2. *The matrix $\bar{\mathbf{F}} \in \mathbb{R}^{m \times n}$ with elements $\bar{f}_{ij}$ is of rank at most k.*

Proof. The measurement equations are $\bar{f}_{i,j} = \mathbf{r}_i^T \mathbf{n}_j$. By setting

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \ldots & \mathbf{r}_m \end{bmatrix}$$

and

$$\mathbf{N} = \begin{bmatrix} \mathbf{n}_1 & \mathbf{n}_2 & \ldots & \mathbf{n}_n \end{bmatrix}$$

we see that $\bar{\mathbf{F}} = \mathbf{R}^T \mathbf{N}$ where $\mathbf{R} \in \mathbb{R}^{k \times m}$ and $\mathbf{N} \in \mathbb{R}^{k \times n}$. Both $\mathbf{R}$ and $\mathbf{N}$ have at most rank k, therefore the same holds for $\bar{\mathbf{F}}$. □

In the following, we focus on the 3D problems ($k = 3$). The algorithm as well as the theory can be generalized to 2D. Assuming that m and n are large enough and assuming that the motion $\mathbf{r}_i$ and the directions to transmitters $\mathbf{n}_j$ are in general enough constellation, the matrix $\bar{\mathbf{F}}$ will have rank 3. In that case, it is possible to reconstruct both $\mathbf{R}$ and $\mathbf{N}$ up to an unknown transformation. This can be done using singular value decomposition, $\bar{\mathbf{F}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. By setting e.g. $\tilde{\mathbf{R}} = \mathbf{U}_3^T$ and $\tilde{\mathbf{N}} = \mathbf{S}_3\mathbf{V}_3^T$, we get all possible solutions by and $\mathbf{R} = \tilde{\mathbf{R}}\mathbf{A}^{-1}$ and $\mathbf{N} = \mathbf{A}\tilde{\mathbf{N}}$, with $\mathbf{A}$ a general full rank 3×3 matrix. Changing $\mathbf{A}$ corresponds to rotating, affinely stretching and possibly mirroring the coordinate system. The true reconstruction also fulfills $\mathbf{n}_j^T \mathbf{n}_j = 1$, which gives constraints on A of type

$$\mathbf{n}_j^T \mathbf{A}^T \mathbf{A} \mathbf{n}_j = 1,$$

which after substitution $\mathbf{B} = \mathbf{A}^T \mathbf{A}$ becomes linear

$$\mathbf{n}_j^T \mathbf{B} \mathbf{n}_j = 1$$

in the unknown elements of $\mathbf{B}$. Since symmetric 3×3 matrices have 6 degrees of freedom we need at least 6 transmitters to determine the matrix uniquely. Once $\mathbf{B}$ has been determined $\mathbf{A}$ can be determined by Cholesky factorization. This gives the transformation $\mathbf{A}$ up to an unknown rotation and possible mirroring of the coordinate system. We summarize the above in the following theorem.

Theorem 9.1.1. *The minimal case for reconstructing m positions $\mathbf{r}_i$ and n directions $\mathbf{n}_j$ from relative distance measurements $\bar{f}_{i,j}$ as formulated in Problem 9.1.2 is $m = 4$ and $n = 6$.*

Accordingly, we have the following algorithm for the minimal case of the problem. Note that using minimal information $m = 4$ and $n = 6$ results in estimates that fulfill the measurements exactly (up to machine precision) even if the measurements are disturbed by noise.

Algorithm 9.1.1 Far-Field TDOA in 3D - Linear Method

Given the measurement matrix $\mathbf{F} \in \mathbb{R}^{4\times 6}$, outputs the receivers positions $\{\mathbf{r}_i\}$ and transmitter directions $\{\mathbf{n}_j\}$.

1. Construct matrix $\bar{\mathbf{F}}$ such that $\bar{f}_{i,j} = f_{i+1,j} - f_{1,j}$,
2. Calculate a singular value decomposition $\bar{\mathbf{F}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$.
3. Set $\tilde{\mathbf{R}}$ to first 3 rows of $\mathbf{U}^T$ and $\tilde{\mathbf{N}}$ to first 3 rows of $\mathbf{\Sigma}\mathbf{V}^T$.
4. Solve for the six unknowns in the symmetric matrix $\mathbf{B}$ using the 6 linear constraints $\tilde{\mathbf{n}}_j^T \mathbf{B} \tilde{\mathbf{n}}_j = 1$.
5. Calculate $\mathbf{A}$ by Cholesky factorization of $\mathbf{B}$, so that $\mathbf{A}^T\mathbf{A} = \mathbf{B}$.
6. Calculate receiver position as $\mathbf{r}_i = \mathbf{A}^{-T}\tilde{\mathbf{r}}_i$ and transmitter direction as $\mathbf{n}_j = \mathbf{A}\tilde{\mathbf{n}}_j$.

9.1.1 Failure Modes of the Algorithm

The algorithm fails in certain configurations of the positions of receivers and transmitters. This is captured by the following theorem.

Theorem 9.1.2. *The minimal case for reconstructing m positions $\mathbf{r}_i$ and n directions $\mathbf{n}_j$ from relative distance measurements $\bar{f}_{i,j}$ as formulated in Problem 9.1.2 is for $m = 4$ and $n = 6$. As long as the orientations $\mathbf{n}_j$ do not lie on a common quadratic cone $\mathbf{n}_j^T \Omega \mathbf{n}_j = 0$ and the measurement positions $\mathbf{r}_i$ do not lie on a plane, there will not be more than one solution to the problem of determining both $\mathbf{r}_i$ and $\mathbf{n}_j$ up to an unknown translation, orientation and reflection of the coordinate system.*

Proof. The algorithm can fail if the measurement matrix $\bar{\mathbf{F}}$ has rank 2 or lower. This could e.g. happen if either all measurement positions $\mathbf{r}_i$ lie in a plane or if all directions $\mathbf{n}_j$ lie in a plane (or both). The algorithm can also fail if there are two solutions to the

matrix $\mathbf{B}$ in $\mathbf{n}_j^T \mathbf{B} \mathbf{n}_j = 1$. But then the difference $\mathbf{\Omega} = \mathbf{B}^* - \mathbf{B}$ of these two solutions is a 3×3 matrix for which

$$\mathbf{n}_j^T \mathbf{\Omega} \mathbf{n}_j = 0,$$

which in turn implies that the directions $\mathbf{n}_j$ lie on a common conic as represented by the matrix $\mathbf{\Omega}$. □

Another type of failure mode of the algorithm related to the existence of noise in measurements and the far-field approximation. If the data is corrupted by noise or far-field approximation is not valid, it could lead to an estimate of the matrix $\mathbf{B}$ that is not positive definite. Then the algorithm fails because there is no Cholesky factorization of $\mathbf{B}$ into $\mathbf{A}^T \mathbf{A}$. If $\mathbf{B}$ is unique, then there are no real solutions to the problem in this case.

Further Analysis of Failure Modes

If the rank of the matrix $\bar{\mathbf{F}}$ is 2, this could be because the points $\mathbf{r}_i$ lie on a plane or that $\mathbf{n}_j$ lie on a plane. In the case of coplanar $\mathbf{r}_i$, it is still possible to estimate the planar coordinates $\mathbf{R} = \mathbf{U}_2 \mathbf{A}^{-1}$ and $\mathbf{N} = \mathbf{A} \mathbf{S}_2 \mathbf{V}_2^T$ up to an unknown 2×2 matrix $\mathbf{A}$ representing a choice of affine coordinate system. Here we do get inequality constraints that

$$\left| \mathbf{A} \begin{pmatrix} \mathbf{n}_{j,x} \\ \mathbf{n}_{j,y} \end{pmatrix} \right| \leq 1.$$

Each such $\mathbf{A}$ is a potential solution. It is possible to extend with a third coordinate in the normal direction according to

$$\mathbf{n}_{j,z} = \pm \sqrt{1 - \mathbf{n}_{j,x}^2 - \mathbf{n}_{j,y}^2}.$$

Another possibility is that the directions $\mathbf{n}_j$ lie on a plane. In this case it is possible to reconstruct two of the coordinates for both the positions $\mathbf{r}_i$ and the directions $\mathbf{n}_j$. Since the normals are assumed to lie in a plane, we can exploit the equality constraints $\mathbf{n}_j^T \mathbf{A}^T \mathbf{A} \mathbf{n}_j = 1$ similar to the rank 3 case. In this particular case we only need three directions $\mathbf{n}_j$, i.e. the minimal case is for $m = 3$ and $n = 3$. This gives the full reconstruction of both points and directions up to an unknown choice of Euclidean coordinate system and unknown choice of z-coordinate for receiver positions $\mathbf{r}_i$.

If the rank of $\bar{\mathbf{F}}$ is 1, this could be because the directions are parallel or the receivers lie on a line. In the case of parallel directions, similar to the discussions above we can obtain one of the coordinates of the positions $\mathbf{r}_i$, but this is trivial since the measurements $\bar{f}_{i,j}$ are such coordinates by definition. As for cases when receivers lie on a line, we obtain a one-parameter family of reconstructions based on $\mathbf{R} = \mathbf{U}_1 a$ and $\mathbf{N} = a \mathbf{\Sigma}_1 \mathbf{V}_1^T$, where a is an unknown constant that has to fulfill $a \leq 1/l$, where $l = \max_j |\mathbf{\Sigma}_1 \mathbf{V}_{1,j}|$. For each such a it is possible to extend the directions $\mathbf{n}_j$ so that they have length one, but there are several such choices.

9.1.2 Overdetermined Cases

When more measurements are available than the minimal case discussed in the previous section, we need to solve an overdetermined system in least-square sense or with robust error measures e.g. L_1-norm. Here we focus on the following least-square formulation of the problem.

Problem 9.1.3 Given measurement matrix $\bar{\mathbf{F}} = \{\bar{f}_{ij}\}$ from receivers at m different positions to n transmitters, determine both the relative motion of the receiver $\mathbf{r}_i$ and the direction to the transmitters $\mathbf{n}_j$ so that

$$\begin{array}{ll} \min_{\mathbf{R},\mathbf{N}} & \|\bar{\mathbf{F}} - \mathbf{R}^T\mathbf{N}\|_{\mathcal{F}}^2, \\ s.t. & \|\mathbf{n}_j\|_2 = 1, \end{array} \tag{9.4}$$

where $\|.\|_{\mathcal{F}}$ denotes the Frobenius norm.

For the overdetermined cases, it is possible to modify Algorithm 9.1.1 to obtain an efficient but not necessarily optimal algorithm that finds a reconstruction using the following three modifications (i) the best rank 3 approximation can still be found in Step 4-5 using the singular value decomposition, (ii) the estimate of $\mathbf{B}$ in step 6 can be performed in a least squares sense and (iii) re-normalize the columns of $\mathbf{N}$ to length 1. This results in a reconstruction that differs from the measurements, but both steps are relatively robust to noise. The problem of $\mathbf{B}$ not being positive semi-definite can be attacked by nonlinear optimization. Here we try to optimize $\mathbf{A}$ so that $\sum_{j=1}^{n}(\mathbf{n}_j^T\mathbf{A}^T\mathbf{A}\mathbf{n}_j - 1)^2$ is minimized. This can be achieved e.g. by initializing with $\mathbf{A} = \mathbf{I}$ and then using nonlinear optimization of the error function.

Clearly, we lose any guarantee on the optimality of the solution when we enforce the constraints as in step (iii). However, the solution can serve as a good initialization for subsequent optimization algorithms we present in this section. We discuss how to use alternating optimization and Levenberg-Marquardt algorithm (LMA) to obtain better solution. The first algorithm starts with an initial feasible solution for $\mathbf{R}$ and $\mathbf{N}$, and then it alternates between optimizing $\mathbf{R}$ given $\mathbf{N}$ and vice versa. In the latter approach, the directions are parameterized in a way such that the norm constraints are preserved. For both methods, we need to treat the constraints on the direction vectors properly to ensure convergence.

Alternating Optimization

In order to find the local minima of Problem 9.1.3, we can use a coordinate descent scheme. Specifically, we would like to iteratively optimize the cost function in Problem 9.1.3 with respect to $\mathbf{R}$ given $\mathbf{N}$, and then find the optimal feasible $\mathbf{N}$ with fixed $\mathbf{R}$. If we initialize $\mathbf{N}$ such that it satisfies the norm constraints, we can easily see that the alternating procedure is converging.

To enable the alternating optimization, we need to solve two separate optimization problems. The first one is to find the optimal $\mathbf{R}$ given $\mathbf{N}$. This is the classic least-square problem and is known to be convex and can be solved linearly. On the other hand, solving for optimal $\mathbf{n}_j$ given $\mathbf{R}$ is not convex due to the additional constraints on the $\mathbf{n}_j$'s. In this case, we seek the local minima for each $\mathbf{n}_j$ as a constrained minimization problem. We solve each small constrained minimization problems (3 variables each) independently with interior point method. Alternatively, we can solve the constrained optimization as solving polynomial equations. This can be related to the fact that for a given $\mathbf{R}$, level sets of the cost function with respect to $\mathbf{n}_j$ are surfaces of a ellipsoid in $\mathbb{R}^3$ (the centers are in this case the solution from singular value decomposition). The norm 1 constraints on $\mathbf{n}_j$ geometrically means that the feasible solutions lie on the unit sphere centered at origin. Therefore, the optimal solution of $\mathbf{n}_j$ is one of the points that the ellipsoid is tangent to the unit sphere, which can be found by solving polynomial equations. While there could exist multiple solutions, we can choose the one with minimum Euclidean distances to the center of the ellipsoid. Unlike interior point solver, we always find the global optimum. However, in practice, we found that in the alternating procedure, interior point method and polynomial solving give similar performance.

Levenberg-Marquardt Algorithm

It is well-known that alternating optimization as a coordinate descent scheme converges slowly in practice. Alternatively, we can solve the minimization problem by iteratively finding the best descent direction for $\mathbf{N}$ and $\mathbf{Z}$ simultaneously. Given that Problem 9.1.3 is a nonlinear least-square problem with constraints, Levenberg-Marquardt algorithm can be applied. The difficulty here is again the constraints on the direction vectors $\mathbf{n}_j$. The key idea here is to re-parameterize the direction vectors. Given a direction vector $\mathbf{n}$ having unit length, any direction vectors can be represented by $\exp(\mathbf{S})\mathbf{n}$, where $\mathbf{S}$ is a 3×3 skew-symmetric matrix. This is due to the fact that the exponential map of any such matrix is a rotation matrix. In this case, if we use the (current) direction $\mathbf{n}$ as axis direction, any local change of the direction on the sphere can be easily parameterized via the exponential map. Therefore, the Jacobian with respect to $\mathbf{n}_j$ can be expressed without any constraints. To this end, we can then construct the Jacobian for the Levenberg-Marquardt algorithm to compute the optimal descent direction.

9.2 Synthetic Experiments

In this section, we present experimental results for simulated data. We focus on the performance of the minimal solver as well as verification of the far-field approximation. We also study the performance of two iterative solvers for overdetermined cases.

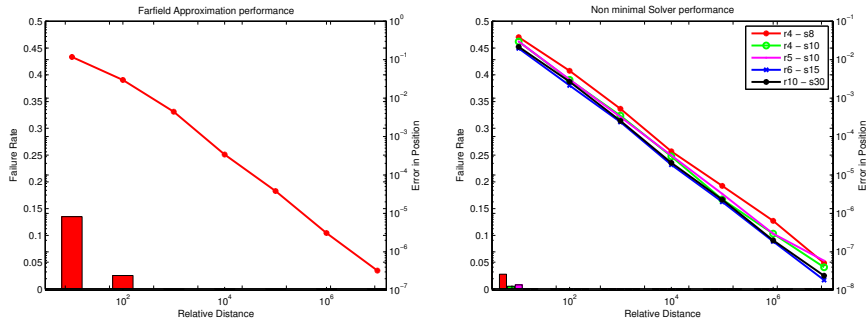


Figure 9.1: Left: Performance of the minimal solver. Bars show failure rate (left y-axis) and curve shows the errors of estimated position as a function of distance (right y-axis). Right: Performance on non-minimal cases. In the legend, rm-sn denotes m receivers and n transmitters. Plots are best viewed in color.

9.2.1 Minimal Solver Accuracy

The numerical performance of the algorithm was evaluated on non-degenerate random problems where the far-field approximation is true. In essence, we generate the directions as general 3D directions and the receiver positions $\mathbf{r}_i$ in general 3D positions such that relative distance measurements $f_{ij} = \mathbf{r}_i^T \mathbf{n}_j$. Specifically, the receiver positions were selected as the corners of a tetrahedron with arc-length one. The errors are then evaluated as the Euclidean distances between the estimated receiver positions and the ground truth positions after registration. Average error of 10000 such tests was 6.8×10^{-15}, which is close to machine epsilon.

9.2.2 Far-Field Approximation Accuracy

Minimal Case

To evaluate the effects of the far-field approximation, we first generate 3D positions for both transmitters and receivers with different ratio between the transmitter-receiver distances and the intra-receiver distances. The constellation of receivers is a unit-length tetrahedron centered at the origin and transmitters are randomly placed on a sphere also centered at the origin. We study the errors as a function of radius of the sphere (that is relative distance of the transmitters to the receivers). As expected, the far-field approximation gets better (smaller errors when compared to ground truth) as the relative distance increases (Figure 9.1, Left). We also investigate the failure rate of the minimal solver with respect to the far-field approximation. A failure corresponds to a case in which the $\mathbf{B}$ matrix in Algorithm 9.1.1 is not positive definite. As can

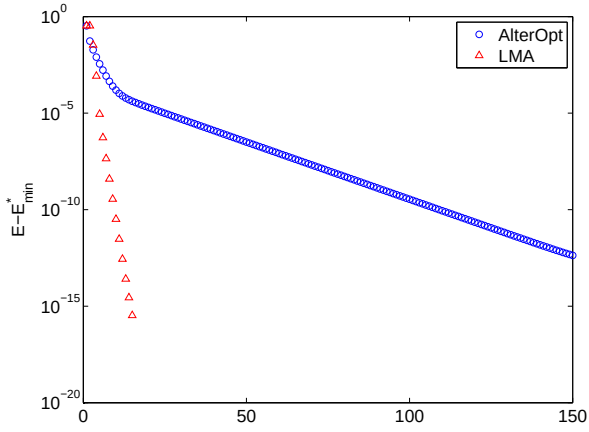


Figure 9.2: Convergence of alternating optimization and LMA on simulated TDOA measurements with Gaussian noise $\sigma = 0.1$. Here $m = 10$ and $n = 10$.

be seen (Figure 9.1, Left), this is infrequent even at small relative distances when one would not expect a far-field approximation to work.

Initialization for Overdetermined Cases

As described in Section 9.1.2, Algorithm 9.1.1 can be used for overdetermined cases. In these situations, these solutions can serve as initial guess for the nonlinear optimization problems. The additional constraints should however affect the numerical stability and it is interesting to evaluate the algorithm for initial estimates in overdetermined cases. In this experiment, we generate the positions of the receivers in a unit-cube, while the transmitters are placed on a sphere centered at the origin with varying radius. In 9.1 (right), we can see that as we increase the number of m and n, the failure rate drops, in some of the cases to zero. We can also see that adding more measurements will in general result in smaller errors.

9.2.3 Overdetermined Cases

We also investigate the performance of the two iterative schemes for over-determined cases. In all experiments below, we initialize both the alternating optimization and LMA based on the linear solver. The simulated data is of a true far-field approximation with Gaussian white noise, i.e. measurements are simulated as $f_{i,j} = \mathbf{r}_i \mathbf{n}_j + o_j + \epsilon_{ij}$ where the i.i.d Gaussian noise $\epsilon_{ij} \in N(0, \sigma)$. In Figure 9.2, we can see that alternating optimization and LMA all decrease the reconstruction errors after initialized by the linear solver. On the other hand, from figure 9.2, we can see that LMA converges in

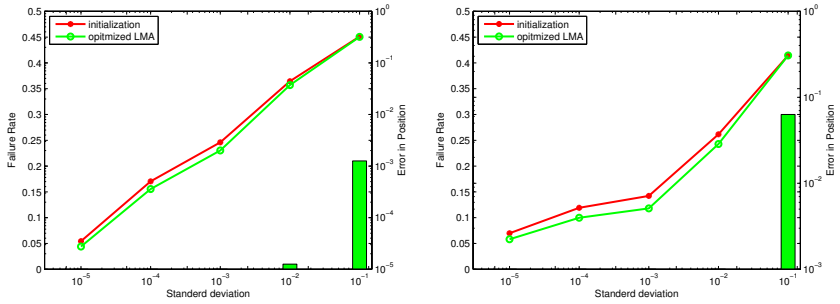


Figure 9.3: Performance on non-minimal cases with simulated TDOA measurements with Gaussian white noise. The mean errors in positions of the receivers are plotted against standard deviations of the noise. Here $m = 5, n = 10$ and the relative distance to receivers and transmitters are 10^7 (left) and 10^2 (right). Failure rates for the initialization are also shown for completeness.

around 20 iterations which is much faster than alternating scheme (150 iterations) and results in relatively lower reconstruction errors. This verifies the superiority of LMA over coordinate descent. This observation is consistent over different m and n as well as a variety of noise levels. Note that here for all the experiments, we set the damping factor λ for LMA to 1.

It it also of interest to view the complete system when the measurements $f_{i,j}$ does not fulfill the far-field approximation and when the measurements are corrupted by noise. The relative distances of the simulated transmitters and receivers are set to 10^2 for a mediocre far-field approximation and 10^7 for a good far-field approximation. TDOA measurements f_{ij} are then simulated and perturbed with Gaussian white noise. It is shown in Figure 9.3, the solution of the linear method as initialization is fairly good, and in many cases the LMA further decreases the position errors. The system is also fairly robust to noise.

9.3 Real Experiment

In this section, we apply the far-field TDOA self-calibration algorithm to anchor free positioning using UWB measurements from one single transmitter to a single moving receiver. Base on estimated time delay of *multipath components* (MPC) on the receivers, simultaneous estimation of both receiver motion and transmitter positions and its reflections are performed using the factorization based approach followed by nonlinear least squares optimization.

9.3.1 Signal model

If we assume that pulse distortion can be neglected, the impulse response of the UWB channel can be modeled as [74]

$$h(\tau, t) = \sum_l \alpha_l(t)\delta(\tau - \tau_l(t)), \tag{9.5}$$

where t denotes time, $\delta(\cdot)$ is the Kronecker delta function, $\alpha_l(t)$ and $\tau_l(t)$ are the channel gain and delay of the lth multipath component (MPC), respectively. We assume that the scatterers in the environment and the transmitter (Tx) are fixed during the movement of the receiver (Rx), and hence that the only change of the impulse is due to movements of the Rx (or vice versa). The impulse response is sampled at different positions in space, and we replace the time continuous variable t with a sample index i. For the tracking method described later on, we further assume that there is a maximum movement between the samples of the impulse response. For the position tracking, for each i we extract the gains, $\alpha_{i,l}$, and delays, $\tau_{i,l}$, of the 100 strongest MPCs from the impulse responses by the method described in [90]. This method is basically is a variant of the CLEAN algorithm [29], and is based on a detect and subtract approach when extracting the MPCs. The major MPCs typically stem from the dominating scatterers in the environment and the change of delay between successive samples of a particular MPC reflects the movement of the antenna in relation to this scatterer.

9.3.2 Finding Correspondences among MPCs

After multiplication with the speed of light c, each delay corresponds to a propagation distance $d_{i,k} = c\tau_{i,k}$, between the transmitter and receiver for that particular MPC, possibly as it has been scattered and reflected in the surroundings. Each scatterer, being a planar surface or a smaller reflecting object, gives rise to a virtual transmitter position $\mathbf{s}_j$. If one can find the correspondences, i.e. for each virtual transmitter $\mathbf{s}_j$ can find all those distances $d_{i,j}$, then one obtains a structure from motion problem of the following type: given measurements $d_{i,j}$ determine both transmitter positions $\mathbf{s}_j$ and receiver positions $\mathbf{r}_i$ such that $d_{i,j} = \|\mathbf{r}_i - \mathbf{s}_j\|_2$. As we will see in the next section, the far-field approximation (9.1) can be safely applied to our experiment setup where receiver motions is constrained within a small region in space while the virtual transmitters are sufficiently far away.

9.3.3 Measurement Setup

Measurements were conducted in a furnished lecture room of size 8.1 x 6.3 x 2.6 m^3 (Figure 9.4). A transmit antenna at a height of 1.47 m was put in the middle of the lecture room. The receiver antenna is put at a distance on a straight line of sight away

from the transmit antenna, the x direction. The motion of the receiver is controlled by a 3D positioner which gives ground truth motion with an accuracy of 50 μm. As a proof of concept, and due to practical constraints with the positioner, the movements are limited to a cube measuring 0.30 x 0.30 x 0.30 m^3. The receiving antenna first moves linearly in x, y, z directions and then moves on the surface of xy, xz and yz, making a shape of square on each surface. The steps between successive antenna positions are 1 cm each, but not that we make no assumptions on the movements other than that there is a maximum distance between measurements. The measurements were performed with an HP 8720C vector analyzer (VNA) using SkyCross UWB antennas SMT-3TO10M-A at both the Tx and Rx end. The VNA measures S_{21}, the channel transfer function, for 1601 frequency points, sweeping the whole bandwidth of 3.1 GHz to 10.6 GHz. Note that the expected delay resolution, as measured by the inverse of the bandwidth is 133 ps, corresponding to a distance of 4 cm. The IF bandwidth was set to 100 Hz in order to minimize the impact of noise. The environment was static during the measurements and except for the moving receiver there are no other movements in the close environment. The measurements of the matching strength between transmitted and received signal are shown in Figure 9.5 (Left).

9.3.4 Data Processing

The results from the signal matching are the travel times $\tau_{i,l}$ for the 100 strongest peaks $i = 1, \ldots, 100$ at each of the 404 measurements positions where $l = 1, \ldots, 404$. These are shown in Figure 9.5 (*right*). We used a semi-automatic tracking method to find as long and as complete matched tracks as possible, for 6 of the most prominent tracks in the data, $j = 1, \ldots, 6$, i.e. $d_{i,j} = d_{i,l(i,j)}$. As can be seen in Figure 9.5, it is relatively straightforward to determine the travel times for the direct path from the single stationary transmitter to the moving receiver. As we will see later on the five other tracks correspond to reflections in roughly planar structures (walls) in the building. Each such path can thus be considered as originating from a stationary transmitter in an unknown (reflected) position. For 387 of these 404 time instants i a matched distance can be found to all 6 (real or virtual) transmitters. After computing $\bar{\mathbf{D}}$ as in Algorithm 9.1.1 from the 387×6 measurement matrix, it is observed that $\bar{\mathbf{D}}$ is almost of rank 2, which indicates that either (i) the receiver motion is roughly planar or (ii) the directions n_j to the transmitters are roughly planar. The planar version of Algorithm 9.1.1 (planar version) is then used to obtain an initial estimate of $(\mathbf{r}_i, \mathbf{n}_j, o_j)$, which is used as a starting point for a nonlinear least squares refinement of the parameters

$$(\mathbf{r}_i, \mathbf{n}_j, o_j) = \operatorname{argmin}_{\mathbf{r}_i, \mathbf{n}_j, o_j, \|\mathbf{n}_j\|_2 = 1} \sum_{i,j} \left\| d_{i,j} - (\mathbf{r}_i^T \mathbf{n}_j + o_j) \right\|^2, \tag{9.6}$$

using the Levenberg-Marquart algorithm.

Figure 9.4: Overview of the measured scenario. Tx antenna in front and Rx antenna in the back.

9.3.5 Position Estimates

The results of estimated receiver motion is shown in Figure 9.7 (*left*). Note that since the tracked paths all correspond to reflections in the horizontal plane (mainly walls), we found no reflections in the floor or ceiling, only the planar coordinates of the motion can be estimated. We believe that the reason that we find no reflections from the ceiling or floor is that the antenna patterns are fairly omnidirectional in the horizontal plane, but have a very low gain in the elevation angles close to 0 or 180 degrees. Due to this, the vertical motion cannot be obtained.

Once the initial estimate has been found for 387 of the 404 receiver positions, it is straightforward to extend the solution to all receiver positions and to additional transmitter positions. In our experiments, assuming the motion of the receiver is smooth, we initialized these additional 17 receiver positions by linear interpolation and refined the whole reconstruction using nonlinear minimization. Notice that for the nonlinear optimization it is possible to refine all parameters even if there are missing data simply by letting the sum in (9.6) be over all index pairs (i, j) for which there are measurements.

As a final step, we implemented a RANSAC approach, cf. [35], for finding additional multipath component matches. In this step we exploited the fact that such

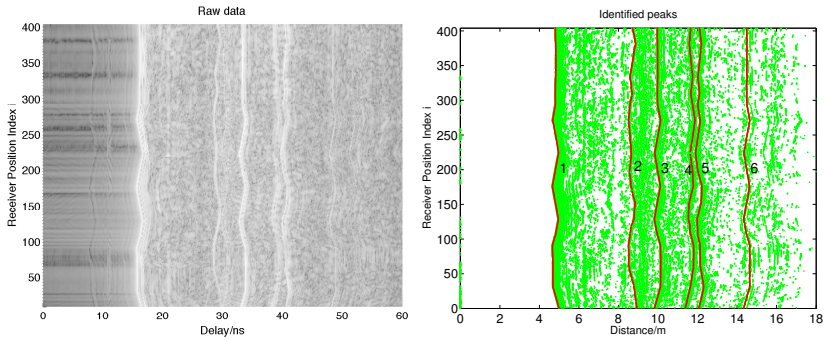


Figure 9.5: Left: Matching strength between transmitted signal and received signal for each receiver position index i and each time delay. Notice that the strong peaks form tracks along the i-direction. Right: 100 strongest peaks (green) and the 6 best tracks (red) for each receiver position index i.

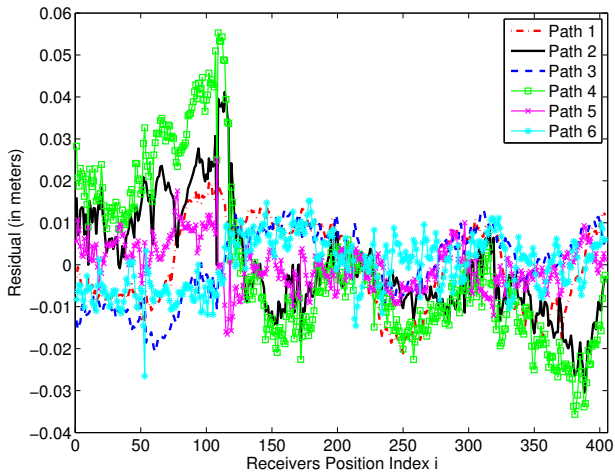


Figure 9.6: Residuals between the estimated distances $\mathbf{r}_i^T \mathbf{n}_j + o_j$ and the measured distances $d_{i,j}$ after nonlinear least squares optimization. Notice that the residuals are in the order of one centimeter.

multipath component tracks are constrained to two parameters (θ, w), i.e.

$$d_i \approx \mathbf{r}_i^T \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} + w. \tag{9.7}$$

Here the changing the parameter w corresponds to horizontal translations in Figure 9.5 (*right*) and changing parameter θ corresponds to changing the shape of the matched curve. In the RANSAC loop, one iterates on (i) hypothesizing that two peaks correspond to the same transmitter, (ii) for such a hypothesis one then calculates the two parameters (θ, w) and thus the corresponding matched curve and finally (iii) assessing how many additional matches this curve contains within a threshold. By iterating (i)-(iii) and choosing the matched curve with the most inliers one can obtain additional multipath component matches. Again once a good inlier set has been found it is refined with nonlinear optimization. Unfortunately, no additional multipath component was found with the RANSAC step.

The final reconstruction is shown in Figure 9.7 (*right*). In the figure is shown both the real transmitter location $\mathbf{s}_1$ as a square, the reflected transmitter locations $\mathbf{s}_2 \ldots \mathbf{s}_6$ as circles and the receiver positions $\mathbf{r}_i$ as dots. Note that the receiver motion is relatively small and difficult to perceive in this figure. In the figure we have also illustrated the geometry of the reflective surfaces, which in this case act as the major scatterers.

In order to make a comparison between ground truth motion $\mathbf{r}_{\text{true},i}$ and the estimated motion $\mathbf{r}_i$, we first rotate and translate the ground truth motion, i.e. we optimize

$$(\mathbf{R}, \mathbf{b}) = \text{argmin}_{\mathbf{R},\mathbf{b},\mathbf{R}^\mathbf{T}\mathbf{R}=\mathbf{I}} \sum_{\mathbf{i}} \|\mathbf{r_i} - (\mathbf{R}\mathbf{r}_{\text{true},\mathbf{i}} + \mathbf{b})\|^{\mathbf{2}} \tag{9.8}$$

and then set

$$\mathbf{r}_{\text{fit},i} = \mathbf{R}\mathbf{r}_{\text{true},i} + \mathbf{b}. \tag{9.9}$$

In Figure 9.7 (*left*) we show both the estimated receiver positions $\mathbf{r}_i$ and overlaid the fit $\mathbf{r}_{\text{fit},i}$ of the ground truth motion. The estimated standard deviation (RMS error) for $n = 404$ receiver positions

$$\sigma^\star = \sqrt{\frac{1}{(n-3)} \sum_{i=1}^{n} \|\mathbf{r}_{\text{fit},i} - \mathbf{r}_i\|^2} \tag{9.10}$$

was 1.34 cm. Notice that most residuals (cf. Figure 9.6) are in the order of ± 1 cm, whereas the residuals corresponding to transmitter $j = 4$ and receivers $i \approx 50 \ldots 120$ there are significantly larger residuals. One hypothesis here is there are matching errors here that influence the reconstruction. The receiver positions $i \approx 50 \ldots 120$, correspond to the wiggly upper part in Figure 9.7 (*left*).

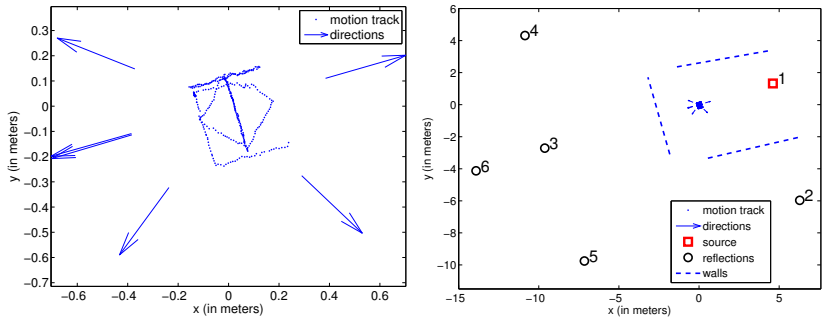


Figure 9.7: Left: ground truth motion (red circles) and the estimated receiver positions r_i (blue dots, 387 of the 404 measurement positions). In the figure is also shown the directions to the real and virtual antenna positions ($-\mathbf{n}_j$'s). Right: Positions of the 'source' (red square) and virtual (reflected) transmitters (black circles). The hypothesized reflective wall positions based on the relative positions between source and reflections are also plotted.

9.3.6 Discussions

In this real experiment, we have showed that how the far-field approximation TDOA calibration scheme can be applied to anchor-free indoor localization. We achieve with roughly centimeter precision using UWB measurements from a single transmitter in an unknown environment to a moving antenna. The absolute travel times of the MPCs from the transmitter to the receivers are measured using a VNA both for the direct path and for reflections in natural indoor features, such as walls. The reconstruction of both receiver positions and real and virtual (reflected) transmitters is cast as a structure from motion problem. Such problems have received increased attention lately and the knowledge on how to solve such problems are refined. Using a combination of factorization, calibration and nonlinear least squares optimization we obtain such estimates of receiver and transmitter positions. A crucial problem here is also the correspondence problem, i.e. the matching of identified distances to transmitter ids. We have used a semi-automatic approach to get an initial estimate. We have also shown how such structure from motion algorithms can be used in a RANSAC fashion to obtain additional matched tracks. Future research includes the development of algorithms for automatic matching already for the initial estimate.

Chapter 10

Far-Field UTOA

In this chapter, we extend the far-field approximation to a more general setting where the clocks for both receivers and transmitters are unsynchronized i.e. far-field unsynchronized time-of-arrival (FFUTOA) problem. By utilizing the same approximation formulation, we describe an extension to the linear factorization technique that recover unknown offsets for both receivers and transmitters.

10.1 Far-Field UTOA Self-Calibration

To start with, we first recall the regular UTOA problem where neither receivers nor transmitters are synchronized. Specifically, the measurements between the receiver i and transmitter j will be of the form $f_{ij} = \|\mathbf{r}_i - \mathbf{s}_j\| + q_i + \tilde{o}_j$ where $q_i, \tilde{o}_j$ are unknown offsets for receivers and transmitters respectively. Following the derivation in (9.3), UTOA measurements in far-field setup can be approximated by

$$\begin{aligned} f_{ij} &= \|\mathbf{r}_i - \mathbf{s}_j\| + q_i + \tilde{o}_j \\ &\approx \|\mathbf{r}_1 - \mathbf{s}_j\| + (\mathbf{r}_i - \mathbf{r}_1)^T \mathbf{n}_j + q_i + \tilde{o}_j \\ &= \mathbf{r}_i^T \mathbf{n}_j + \bar{o}_j + q_i + \tilde{o}_j \end{aligned} \tag{10.1}$$

where $\bar{o}_j = \|\mathbf{r}_1 - \mathbf{s}_j\| - \mathbf{r}_1^T \mathbf{n}_j$ and $\mathbf{n}_j$ is the direction of unit length from transmitter j to the receivers. By setting $o_j = \bar{o}_j + \tilde{o}_j$ we get the far field approximation

$$f_{ij} \approx \mathbf{r}_i^T \mathbf{n}_j + q_i + o_j.$$

When the far-field approximation is valid, we will call f_{ij} far-field UTOA (FFUTOA) measurements.

Problem 10.1.1 Given FFUTOA measurements f_{ij} taken from m receivers and n transmitters, estimate receiver positions $\mathbf{r}_i$, directions $\mathbf{n}_j$ from transmitter j to receivers, receiver and transmitter offsets q_i, o_j such that

$$\begin{aligned} f_{ij} &= \mathbf{r}_i^T \mathbf{n}_j + q_i + o_j, \\ & s.t.\ \|\mathbf{n}_j\|_2 = 1. \end{aligned} \tag{10.2}$$

Note that the problem is symmetric in receivers and transmitters, i.e. if each receiver instead could be viewed as having a common direction to all transmitters, the same problem can be solved. We denote $\mathbf{q} = [q_1, \ldots, q_m]^T$, $\mathbf{o} = [o_1, \ldots, o_n]^T$, $\mathbf{R} = [\mathbf{r}_1, \ldots, \mathbf{r}_m]$ and $\mathbf{N} = [\mathbf{n}_1, \ldots, \mathbf{n}_n]$. In the following discussion, we assume that the receivers and transmitters are in general affine space of dimension k, i.e. $\mathbf{r}_i \in \mathbb{R}_k$ and $\mathbf{n}_j \in \mathbb{R}_k$. The unknown parameters $(\mathbf{R}, \mathbf{N}, \mathbf{q}, \mathbf{o})$ have certain degrees of freedom that does not change the measurements i.e. *gauge freedom*. Based on (10.2), any translation $\mathbf{t}$, rotation matrix $\mathbf{R}_{\text{rot}}$ of coordinate system and offset change c can be applied to the solution as follows without changing the measurements f_{ij}.

- Translation

$$\begin{aligned} \mathbf{r}_{i,\text{trans}} &= \mathbf{r}_i + \mathbf{t} \\ o_{j,\text{trans}} &= o_j - \mathbf{t}^T \mathbf{n}_j \end{aligned}$$

- Rotation

$$\begin{aligned} \mathbf{r}_{i,\text{rot}} &= \mathbf{R}_{\text{rot}} \mathbf{r}_i, \\ \mathbf{n}_{j,\text{rot}} &= \mathbf{R}_{\text{rot}} \mathbf{n}_j \end{aligned}$$

- Offset

$$\begin{aligned} q_{i,\text{offs}} &= q_i + c \\ o_{j,\text{offs}} &= o_j - c \end{aligned}$$

Thus, we can only reconstruct the unknowns up to these degrees of freedom.

10.2 A Matrix Factorization Method

In this section, we utilize the bilinear expression and derive a linear technique to solve Problem 10.1.1 for receiver positions, transmitter directions and offsets. Without loss of generality we assume that the solution is partially normalized for gauge freedom as the first receiver $\mathbf{r}_1 = \mathbf{0}$ and $q_1 = 0$, see Section 10.1.

Using the FFUTOA measurements f_{ij}, collected in the matrix $\mathbf{F} = \{f_{ij}\}$, we can obtain unknowns o_j based on $f_{1,j} = \mathbf{r}_1^T \mathbf{n}_j + q_1 + o_j = o_j$, since $\mathbf{r}_1 = \mathbf{0}$ and $q_1 = 0$. We then form $\bar{\mathbf{F}} = \{\bar{f}_{ij}\}$ where $\bar{f}_{ij} = f_{i+1,j} - f_{1,j}$ This is done by subtracting the first row containing o_j from all other rows of $\mathbf{F}$ and removing the first row of zeros to obtain a new matrix that fulfill

$$\bar{\mathbf{F}} = \begin{bmatrix} \tilde{\mathbf{R}}^T & \mathbf{q} \end{bmatrix} \begin{bmatrix} \mathbf{N} \\ \mathbb{1}_n^T \end{bmatrix} \tag{10.3}$$

where $\mathbb{1}_n$ is a vector of n ones and $\tilde{\mathbf{R}} = \begin{bmatrix}\mathbf{r}_2 & \dots & \mathbf{r}_m\end{bmatrix}$. $\bar{\mathbf{F}}$ is a product of two matrices of rank $\leq k+1$ and is thus itself of rank $\leq k+1$. This is used in [15]. Here we further reduce the rank of the factorization by subtracting the first column of $\bar{\mathbf{F}}$ from all the other columns. This manipulation of $\bar{\mathbf{F}}$ can be done using the compaction matrix $\mathbf{C}_n$ of size $n \times (n-1)$

$$\mathbf{C}_n = \begin{bmatrix} -1 & -1 & \dots & -1 \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}. \tag{10.4}$$

Then we have

$$\tilde{\mathbf{F}} = \bar{\mathbf{F}}\mathbf{C}_n = \begin{bmatrix}\tilde{\mathbf{R}}^T & \mathbf{q}\end{bmatrix} \begin{bmatrix}\tilde{\mathbf{N}} \\ \mathbf{0}\end{bmatrix} = \tilde{\mathbf{R}}^T\tilde{\mathbf{N}}, \tag{10.5}$$

where $\tilde{\mathbf{N}}$ is a $k \times (n-1)$ matrix with the jth column $\tilde{\mathbf{n}}_j = \mathbf{n}_{j+1} - \mathbf{n}_1$. Given that $\tilde{\mathbf{R}} \in \mathbb{R}^{k\times(m-1)}$ and $\tilde{\mathbf{N}} \in \mathbb{N}^{k\times(n-1)}$, now we have a rank-$k$ factorization. To enable such rank-k factorization for $\tilde{\mathbf{F}} \in \mathbb{R}^{(m-1)\times(n-1)}$, at least (k+1) receivers and (k+1) transmitters are required. After applying singular value decomposition to $\tilde{\mathbf{F}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ we obtain the rank-k factorization such that $\tilde{\mathbf{F}} = \bar{\mathbf{R}}^T\bar{\mathbf{N}}$ where $\bar{\mathbf{R}} = \mathbf{U}_k^T$ and $\bar{\mathbf{N}} = \mathbf{\Sigma}_k\mathbf{V}_k^T$. $\mathbf{U}_k$, $\mathbf{\Sigma}_k$ and $\mathbf{V}_k$ are the truncated parts of the SVD corresponding to the k largest singular values. This factorization of $\tilde{\mathbf{F}}$ is unique up to an unknown transformation $\mathbf{L}$ i.e. $\tilde{\mathbf{F}} = \bar{\mathbf{r}}^T\mathbf{L}^{-1}\mathbf{L}\bar{\mathbf{n}}$. We will find $\tilde{\mathbf{n}}_j = \mathbf{L}\bar{\mathbf{n}}$ i.e. $\mathbf{n}_{j+1} - \mathbf{n}_1 = \mathbf{L}\bar{\mathbf{n}}_j$ by using the constraints that

$$\begin{aligned} \tilde{\mathbf{n}}_j^T\tilde{\mathbf{n}}_j &= (\mathbf{n}_{j+1} - \mathbf{n}_1)^T(\mathbf{n}_{j+1} - \mathbf{n}_1) \\ &= 2 - 2\mathbf{n}_{j+1}^T\mathbf{n}_1 \\ &= 2 - 2(\mathbf{L}\bar{\mathbf{n}}_j + \mathbf{n}_1)^T\mathbf{n}_1 \\ &= -2\bar{\mathbf{n}}_j^T\mathbf{L}^T\mathbf{n}_1 \\ &= \bar{\mathbf{n}}_j^T\mathbf{L}^T\mathbf{L}\bar{\mathbf{n}}_j\ . \end{aligned} \tag{10.6}$$

We apply a change of variables with a symmetric matrix $\mathbf{H} = \mathbf{L}^T\mathbf{L} \in \mathbb{R}^{k\times k}$ and a $k \times 1$ vector $\mathbf{v} = \mathbf{L}^T\mathbf{n}_1$. From (10.6), we have the following equation for transmitter j:

$$\bar{\mathbf{n}}_j^T\mathbf{H}\bar{\mathbf{n}}_j + 2\bar{\mathbf{n}}_j^T\mathbf{v} = 0. \tag{10.7}$$

These equations are linear in the elements of $\mathbf{H}$ and $\mathbf{v}$ which have in total $n_v = k(k+1)/2+k$ variables. In general, with $n_v - 1$ such equations (thus n_v transmitters),

We can extract the solutions for $\mathbf{H}$ and $\mathbf{v}$ from the solution to the linear equation which is valid up to an unknown scaling factor and sign. We can determine the sign by using that $\mathbf{H}$ is positive definite and compute $\mathbf{L}$ by applying Cholesky factorization $\mathbf{H} = \mathbf{L}^T\mathbf{L}$.

We can find the scale by using the constraint $\|\mathbf{n}_1\|_2 = \|\mathbf{L}^{-T}\mathbf{v}\|_2 = 1$. Note that fixing the scale in this way will also guarantee that $\mathbf{n}_j^T\mathbf{n}_j = (\mathbf{L}^T\bar{\mathbf{n}}_j + \mathbf{n}_1)^T(\mathbf{L}^T\bar{\mathbf{n}}_j + \mathbf{n}_1) = \underbrace{\bar{\mathbf{n}}_j^T\mathbf{L}^T\mathbf{L}\bar{\mathbf{n}}_j + 2\bar{\mathbf{n}}_j^T\mathbf{L}^T\mathbf{n}_1}_{=0 \text{ by (10.6)}} + \mathbf{n}_1^T\mathbf{n}_1 = \mathbf{n}_1^T\mathbf{n}_1 = 1$. Summarizing these steps gives Algorithm 10.2.1.

Algorithm 10.2.1 Far-Field UTOA Matrix Factorization Method

Given FFUTOA measurement matrix $\mathbf{F} \in \mathbb{R}^{m\times n}$, outputs receiver positions $\{\mathbf{r}_i\}$, transmitter directions $\{\mathbf{n}_j\}$, receiver offsets $\mathbf{q}$ and transmitter offsets $\mathbf{o}$.

1. Set $o_j = \mathbf{f}_{1,j}$, $\mathbf{r}_1 = \mathbf{0}$, $q_1 = 0$, and form matrix $\tilde{\mathbf{F}} = \tilde{f}_{ij}$ where $\tilde{f}_{ij} = f_{i+1,j+1} - f_{1,j+1} - f_{i+1,1} + f_{1,1}$
2. Calculate *singular value decomposition* of $\tilde{\mathbf{F}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ and set $\bar{\mathbf{R}}$ to first k rows of $\mathbf{U}^T$ and $\bar{\mathbf{N}}$ to first k rows of $\mathbf{\Sigma}\mathbf{V}^T$
3. For the unknowns in the symmetric matrix $\mathbf{H}$ and vector $\mathbf{v}$, get the solution space for the equations $\bar{\mathbf{n}}_j^T\mathbf{H}\bar{\mathbf{n}}_j + 2\bar{\mathbf{n}}_j^T\mathbf{v} = 0$ where $\bar{\mathbf{n}}_j$ is the jth column of $\bar{\mathbf{N}}$
4. Set the sign of the solution $\mathbf{H}, \mathbf{v}$ such that $h_{11} > 0$
5. Calculate the Cholesky decomposition $\mathbf{H} = \mathbf{L}^T\mathbf{L}$
6. Fix the scale of the solutions $\mathbf{L}, \mathbf{v}$ so that $\|\mathbf{L}^{-T}\mathbf{v}\| = 1$
7. Set $\mathbf{n}_1 = \mathbf{L}^{-T}\mathbf{v}$, $\mathbf{n}_{j+1} = \mathbf{L}\bar{\mathbf{n}}_j + \mathbf{n}_1$ and $\mathbf{r} = \mathbf{L}^{-T}\hat{\mathbf{r}}$

The condition for the validity of the algorithm is (i) $\mathbf{F}$ must have rank k, (ii) the linear equations (10.7) must only have a null space of dimension one, (iii) $\mathbf{H}$ must be positive definite. To sum up, to ensure the rank-k factorization, at least $k+1$ receivers and transmitters are needed. On the other hand, solving for the unknown $\mathbf{L}$ and $\mathbf{v}$, at least $k(k+1)/2 + k$ transmitters are required. For instance, in general 3D space i.e. $k = 3$, the minimal case is 4 transmitters (receivers) and 9 receivers (transmitters). The algorithm is general for arbitrary k.

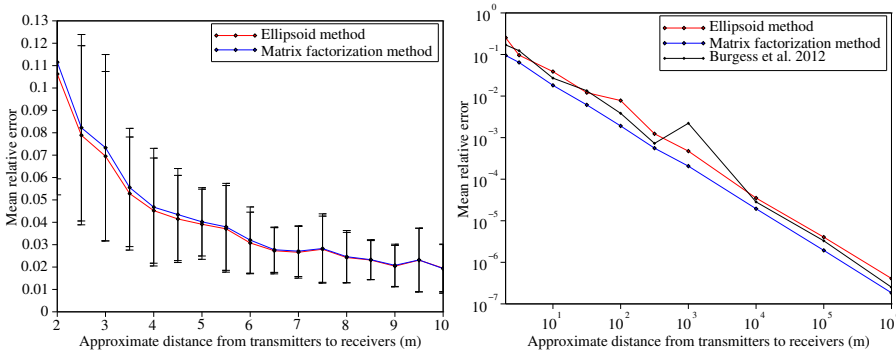


Figure 10.1: Mean relative error of reconstructed receiver positions for 100 runs, plotted against the approximate distance from receivers to transmitters. Bars are ± 1 standard deviation for the different transmitter distances.

10.3 Experimental Validation

To verify the matrix factorization method, we evaluate its performance with both synthetic and real data. We also compare its numerical stability and accuracy with two other methods [15] and [17], respectively. Specifically, [15] uses a similar factorization technique based on rank-$(k+1)$ constraints thus requiring one more receivers and the *ellipsoid method* in [17] solves the minimal problem based on a geometrical formulation. In the following experiment setup, we assume $k = 3$. For all simulations, offsets q_i and o_j are drawn from i.i.d. uniform distributions over $[0, 10]$.

10.3.1 Synthetic Experiments

We first study the numerical stability of the matrix factorization method for minimal problems. We generated noise-free random problems where the measurement matrix $\tilde{\mathbf{F}}$ are FFUTOA (10.2). Receivers are drawn from i.i.d. uniform distributions in a unit cube centered around the origin. 9 transmitter directions and 4 receivers were simulated. Here, we compared the matrix factorization method with the ellipsoid method. The error distribution for 1000 such experiments can be seen in Figure 10.4 (left). We can see that both methods are fairly stable and the matrix factorization method performs slightly better (fewer large errors).

To evaluate the effects of far-field assumption, we first place 4 receivers at a tetrahedron centered around the origin with side length 1 and the 5th receiver is uniformly distributed in the cube of which the tetrahedron of the 4 receivers were inscribed to. To be able to control how much further away transmitters were from receivers than the distances between receivers, transmitter positions $\mathbf{s}_j$ were uniformly distributed over

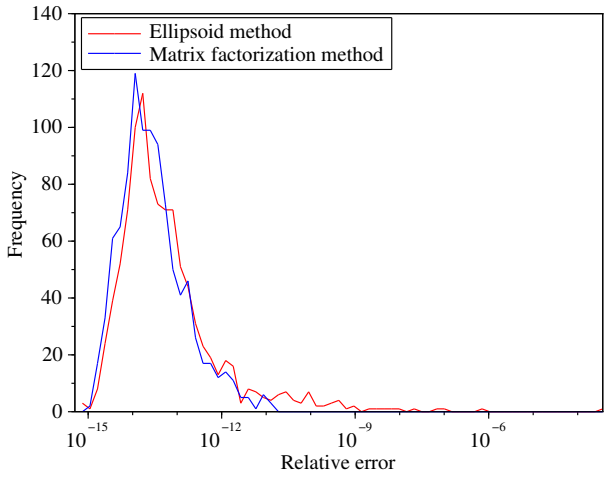


Figure 10.2: Numerical stability of matrix factorization method and ellipsoid method in 1000 simulated experiments for exact far-field measurement.

a sphere centered at the origin of radius d. In this experiments, 15 transmitters were used. UTOA measurements were then constructed as $f_{ij} = \|\mathbf{r}_i - \mathbf{s}_j\|_2 + q_i + o_j$.

The mean relative errors for 100 random runs with respect to different d's can be seen in Figure 10.1 (left) for the minimal 4 receivers and in Figure 10.1 (right) for 5 receivers. Figure 10.1 (left) shows that using only four receivers, both the matrix factorization method and the ellipsoid method can get under 5% relative error when the transmitters are approximately 4 times further away than the inter-distance between receivers. In Figure 10.1 (right), we have also shown the comparison with the method in [15] of which at least five receivers are needed. The results indicate that for varying relative distances, the matrix factorization method presented here is in general better than the ellipsoid method and the overdetermined solver in [15]. Mean execution time is 2.1 ms, 8.0ms and 30 ms for matrix factorization method, the ellipsoid method and the method in [15] each.

To test the noise sensitivity of the methods, Gaussian noise was added to the measurements. In Figure 10.3.1, relative errors of reconstructed receiver positions are plotted against the standard deviation of the noise. The results indicate that the ellipsoid method is slightly better with higher noise level when using 4 receivers, and the matrix factorization method outperforms both the ellipsoid and the method in [15] using five receivers.

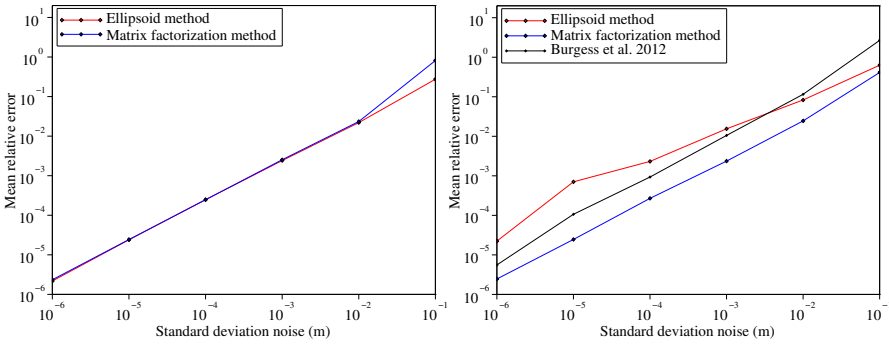


Figure 10.3: Measurements with additive Gaussian white noise. The standard deviation is plotted against the mean relative error of reconstructed receiver positions for 100 runs for 4 receivers, 15 transmitters (left) and 5 receivers, 15 transmitters (right). For this experiment, we choose $d = 10^7$.

10.3.2 Real Data

To test the solvers on real measurements, we generate the far-field UTOA measurements in an indoor stadium with sound recordings. The measurements were obtained from an experimental setup using 8 SHURE SV100 microphones as receivers and random distinct manually made sounds as transmitters. The microphones were connected to a M-Audio Fast Track Ultra 8R audio interface and placed within a cube of roughly $100 \times 105 \times 60\,\text{cm}^3$ centered in the stadium. A picture of the experiment setup can be seen in Figure 10.4. The 19 sound sources were approximately 30 m away from the microphones. Given that the microphones are synced, we can create TDOA measurements after matching beginning of each sound across different microphones with a heuristic cross correlation algorithm. To generate UTOA measurements, we synthetically generate offsets for each microphone as i.i.d. random silence intervals of 0 to 1 second at the beginning of each sound track, effectively starting the recordings at different unknown times.

As we have more than five microphones, we compare all three available algorithms i.e. the matrix factorization method, ellipsoid method and the method in [15]. To evaluate the results, we have compared with the reconstruction of microphone positions from computer vision. The mean reconstruction errors on the microphone positions are 15 cm, 14 cm and 5 cm for the ellipsoid method, the method in [15] and the matrix factorization method, respectively. Most of the error are in the floor-to-roof direction. This can be explained by the sounds all being made close to ground level and thus the transmitter directions will be close to being in a plane, giving poor resolution in the floor-to-roof direction.

Figure 10.4: Setup for indoor experiment using microphones and distinct manually made sounds.

10.4 Conclusions

In this chapter, we have studied the problem of far-field UTOA self-calibration. We identify the minimal cases for this problem e.g. in 3D, at least 4 receivers and 9 transmitters are required. We show that in general there exists one solution to the minimal case and present a linear solver based on matrix factorization. The linear solver is general to 2D and 3D configurations and it is applicable to both minimal and overdetermined problems.

From the discussion in Chapter 9 and this chapter, we can see that in far-field setups, the bipartite self-calibration problems with TOA, TDOA and UTOA measurements are solvable linearly. On the other hand, the geometric problems in near-field settings is much more involved. So far for near-field setups, only minimal problems for TOA self-calibration (Chapter 7) and close-to-minimal problems for TDOA measurements (Chapter 8) are solved. In our initial study with near-field UTOA, we see that the resulting polynomial systems even for the unknown offsets involve large number of unknowns and equations of very high degrees. To solve these problems, we will require the advancements in polynomial solving or other effective and robust strategies to initialize the corresponding non-linear optimization.

Part IV

Image Retrieval

Chapter 11
Entropy Optimization

Feature quantization is a crucial component for efficient large-scale image retrieval and object recognition. By quantizing local features into visual words, one hopes that features that match each other obtain the same word ID. Then, similarities between images can be measured with respect to the corresponding histograms of visual words. Given the appearance variations of local features, traditional quantization methods do not take into account the distribution of matched features. In this chapter, we investigate how to encode additional prior information on the feature distribution via entropy optimization by leveraging ground truth correspondence data. We propose a computationally efficient optimization scheme for large-scale vocabulary training.

11.1 Introduction

In large-scale image retrieval and object recognition, most state of the art techniques are based on the bags of words (BOW) technique. [79, 84, 85, 91]. By quantizing local features (e.g. SIFT [73]) (sampled densly or from keypoints) into a visual vocabulary it is possible to index images similarly to how documents are indexed for text retrieval. The time-consuming exhaustive nearest neighbor search for local feature matching is approximated by feature quantization. The main advantage of BOW for retrieval is the efficient similarity computation between images based on histograms of visual words. Feature quantization is the process of clustering features into discrete unordered sets based on certain criteria. Generally, in image retrieval and object recognition, the criteria can be similarity between features, labels of the features and so on, which lead to unsupervised and supervised feature quantization. For example, k-means and its variants are widely used as unsupervised feature quantization methods to generate large visual vocabularies from e.g. SIFT features based on Euclidean distances. For local feature matching, such a similarity measure is generally a proper criterion. However, due to lighting conditions, perspective transformation, etc. local features can be very different from each other. In this case, unsupervised feature quantization based solely on similarity might fail to capture the intra-class variation of local features. On the other hand, supervised feature quantization utilizes correspondence labels (extracted

as ground truth from some databases) and improves matching performance with respect to such intra-class variation.

In this chapter, we study a supervised feature quantization approach based on entropy optimization. By minimizing the entropy of the quantized vocabulary, we obtain (i) higher matching true positive rate on corresponding local features and (ii) better separation of unmatched features. While the computational complexity for the underlying optimization is high, we propose analytical and numerical schemes to enable large-scale training. We explore the generalization issues of this approach by extensive experiments on datasets with ground truth. Furthermore, we propose a training formulation in the spirit of max-margin clustering that achieves better image retrieval performance than the baseline hierarchical k-means which is widely used.

Related Work Supervised feature quantization has been studied from different perspectives in computer vision. For image categorization, the aim of supervised feature quantization is to incorporate semantic categorical information into the training vocabulary. In such a way, the histogram representation of images encodes the patterns of each category more accurately. Winn et al. [100] optimized the intra-class compactness and inter-class discrimination by merging words from unsupervised k-means. In [83], Perronnin et al. modeled class-specific visual vocabularies with Gaussian mixture models and combine them with a universal vocabulary. In [47], Ji et al. introduced hidden Markov random fields for semantic embedding of local features to facilitate large-scale categorization tasks. On the other hand, with entropy as a criterion, Moosmann et al. constructed random forests based on class labels [77] and Lazebnik et al. [69] simultaneously optimized the cluster centers initialized by k-means and the posterior class distributions. For image retrieval and object recognition, feature quantization is utilized to approximate and speed up the matching process between images. There exist several variants on utilizing existing correspondence information for supervised feature quantization. One way of supervision is to learn optimal projection or apply metric learning before quantization such that the matched pairs of features have small distances than non-matched pairs in the new mapping [22, 86, 94]. All methods achieves substantial improvement in the retrieval tasks. Finally, there are works based on k-means and vocabulary trees. By using a huge dataset with ground truth correspondences, Mikulik et al. [75] train unsupervised vocabulary tree and the apply a supervised soft-assignment of visual words based on the distribution of matched feature points. On the other hand, in [58], initialized by k-means, entropy-based optimization is used to improve the matching performance of the visual vocabulary. Recent works also aim to construct discriminative hashing function for feature quantization with ground truth information [97].

Our approach works on the original feature space and encodes the ground truth correspondences in the process of vocabulary generation. Unlike [69], we focus on optimizing the feature quantization for large-scale feature matching. We also extend

the work in [58] with a formulation for image retrieval, efficient computation for finer quantization and larger correspondence class. There are several limitations of the work in [58]. Firstly, the ground-truth set experimented is too small to generalize. Secondly, very coarse hierarchical quantization ($K = 3$ at each level) is used and it suffered clearly from the quantization errors. We see that the quantization errors can seriously affect the overall true positive rate and false positive rate that suppress the gain from optimization. In this work, we focus on efficient entropy optimization over K in the order of 10^2 and large number of correspondence classes.

11.2 Vocabulary Optimization

In this section, we present the formulation for entropy-based vocabulary optimization. In this formulation, we work with data with partial matching ground truth e.g. local features with labels specifying their corresponding 3D points. To optimize the energy, we have used gradient-descent method. We also utilize the low-rank property to speed up the gradient computation. Finally, we discuss the connection of this formulation to large margin clustering.

11.2.1 Formulation

Entropy traditionally used in information theory for coding has also been applied in supervised learning of vocabulary [58, 69, 77]. In [58], entropy has been shown to be a good criterion to optimize feature matching with respect to true positive rate (TPR) and false positive rate (FPR) in the sense that minimizing the entropy increases the TPR and in the meantime decreases FPR. Given N features of N_c correspondence classes, for a visual vocabulary of N_K words, the entropy is defined as

$$E = -\sum_{k=1}^{N_K} r_k \sum_{j=1}^{N_c} p_{jk} \log p_{jk}, \tag{11.1}$$

where r_k is the percentage of features in cluster k and p_{kj} is percentage of features belonging to correspondence class j that falls in word k. We have used logarithm with base 2 here. By minimizing entropy, one can make each word in the vocabulary more discriminative such that features belonging to the correspondence class tend to fall into the same word.

Now let us denote the total number of features in cluster k as n_k and the number of features of correpondence class j in clustering k as h_{jk}. Substituting $r_k = \frac{n_k}{N}$, and

$p_{jk} = \frac{h_{jk}}{n_k}$ into (11.1), we have

$$\begin{aligned} E &= -\sum_{k=1}^{N_K} \frac{n_k}{N} \sum_{j=1}^{N_c} \frac{h_{jk}}{n_k} \log p_{jk} \\ &= -\frac{1}{N} \sum_{j=1}^{N_c} \sum_{k=1}^{N_K} h_{jk} \log p_{jk} \end{aligned} \tag{11.2}$$

The entropy defined above is not continuous with respect to the word assignments. To enable optimization with gradient descent in the continuous settings, we smooth the word assignment with soft-assignment weights. The weight of a feature $\mathbf{x}_i$ assigned to word k with cluster center $\mathbf{c}_k$ is defined as

$$w_{ik} = \frac{v_{ik}}{\sum_{j=1}^{N_K} v_{ik}}, \tag{11.3}$$

where $v_{ik} = exp(\frac{-\|\mathbf{x}_i - \mathbf{c}_k\|_2}{m})$ and m is the size of the margin that controls the degree of distance smoothing. $\|.\|_2$ denotes the L_2 norm. For each feature $\mathbf{x}_i$, the weights are normalized such that $\sum_{k=1}^{N_K} w_{ik} = 1$. We can immediately see that both n_k and h_{kj} can be written as functions of w_{ik}'s: $n_k = \sum_{i=1}^{N} w_{ik}$ and $h_{kj} = \sum_{i \in \pi_j} w_{ik}$, where π_j is the set of features belonging the correspondence class j.

Optimizing the entropy in (11.1) with respect to the N_K cluster centers amounts to the following minimization problem:

$$\min_{\mathbf{c}} -\frac{1}{N} \sum_{j=1}^{N_c} \sum_{i \in \pi_j} \sum_{k=1}^{N_K} w_{ik} \log \frac{\sum_{i \in \pi_j} w_{ik}}{\sum_{i=1}^{N} w_{ik}}, \tag{11.4}$$

where $\mathbf{c} = [\mathbf{c}_1^T, \mathbf{c}_2^T, \ldots, \mathbf{c}_{N_K}^T]^T$.

In [69], the probability of each feature belonging to each correspondence class can also be updated. In the case here, we assume the correspondence classes estimated from geometry models are of high quality. Due to the fact that N_c is large in our setting, it is generally quite difficult to obtain good estimation of such probabilities, which is also quite different from the scenario in [69] where categorical labels of local patches obtained from image labels are generally very noisy. Therefore, we only focus on optimizing $\mathbf{c}$. Here m is seen as parameter and is determined by cross-validation. For the non-linear optimization, we initialize the center $\mathbf{c}$ with k-means and use gradient descent method L-BFGS to obtain a local minima. We derive the analytical gradient and relevant efficient implementation in the next section.

11.2.2 Efficient Gradient Computation

The gradient of E with respect to the centers $\mathbf{c}$ can be derived analytically. Given (11.2), we have

$$\nabla E = -\frac{1}{N}\sum_{j=1}^{N_c}\sum_{k=1}^{N_K}\left(\nabla h_{jk}\log p_{jk} - \frac{1}{\ln(2)}n_k\nabla p_{jk}\right) \tag{11.5}$$

Since $\sum_{k=1}^{N_K} n_k p_{jk} = |\pi_j|$ is a constant, we can see that

$$\nabla(\sum_{k=1}^{N_K} n_k p_{jk}) = \sum_{k=1}^{N_K} n_k \nabla p_{jk} = 0.$$

We then have

$$\nabla E = -\frac{1}{N}\sum_{j=1}^{N_c}\sum_{k=1}^{N_K}\nabla h_{jk}\log p_{jk} \tag{11.6}$$

And we know that $h_{kj} = \sum_{i\in\pi_j} w_{ik}$, therefore, we have $\nabla h_{jk} = \sum_{i\in\pi_j}\nabla w_{ik}$ and

$$\nabla E = -\frac{1}{N}\sum_{j=1}^{N_c}\sum_{i\in\pi_j}\sum_{k=1}^{N_K}\nabla w_{ik}\log p_{jk} \tag{11.7}$$

where $\nabla w_{ik} = (\frac{\partial w_{ik}}{\partial \mathbf{c}_1}^T, \ldots, \frac{\partial w_{ik}}{\partial \mathbf{c}_{N_K}}^T)^T$. It can be shown that, given (11.3) ,

$$\frac{\partial w_{ik}}{\partial \mathbf{c}_{k'}} = \frac{1}{2}(\delta_{kk'}w_{ik'} - w_{ik'}w_{ik})\frac{\mathbf{x}_i - \mathbf{c}_{k'}}{m||\mathbf{x}_i - \mathbf{c}_{k'}||_2} \tag{11.8}$$

where $\delta_{kk'} = 1$ if $k = k'$, else $\delta_{kk'} = 0$.

Regarding computational complexity, ∇w_{ik} is a vector of length dN_K and computing it takes $\mathcal{O}(dN_K)$, where d is the number of dimension of the features. Therefore, the overall computational complexity for computing ∇E is $\mathcal{O}(dN{N_K}^2)$. We can enable more efficient gradient computation by utilizing the structure of the problem. Firstly, we can observe that

$$\sum_{k=1}^{N_K}\nabla w_{ik} log(p_{jk}) = \frac{\partial \mathbf{w}_i}{\partial \mathbf{v}_i}\frac{\partial \mathbf{v}_i}{\partial \mathbf{c}}\,\boldsymbol{\alpha} \tag{11.9}$$

where $\mathbf{w}_i = (w_{i1}, \ldots, w_{i,N_K})^T$, $\mathbf{v}_i = (v_{i1}, \ldots, v_{i,N_K})^T$ and $\boldsymbol{\alpha} = (\log p_{j1}, \ldots, \log p_{j,N_K})^T$. On the other hand, we have

$$\frac{\partial \mathbf{w}_i}{\partial \mathbf{v}_i} = \frac{1}{\sum_{k=1}^{N_K} v_{ik}}\,\mathbf{I}_{d\times N_K} - \frac{1}{\sum_{k=1}^{N_K} v_{ik}}\,\mathbf{w}_i\,\mathbb{1}_{d\times N_K}^T \tag{11.10}$$

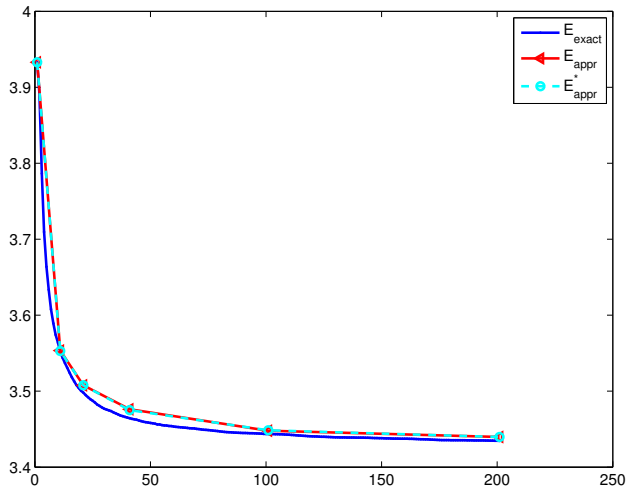


Figure 11.1: The convergence of the L-BFGS with exact and approximate function and gradient calculation. E_{exact} is the convergence of the exact calculation. E_{appr} and E^*_{appr} are the convergence with approximate calculation, while evaluated for all features and only those features within distance threshold μm. The relative difference between the approximate and exact computation is of order 10^{-4}, which takes $40s$ and $275s$ respectively.

where $\mathbf{I}_{d\times N_K}$ is a identity matrix of size $dN_K \times dN_K$ and $\mathbb{1}^T_{d\times N_K}$ is a $dN_K \times 1$ vector of all 1's, and

$$\frac{\partial \mathbf{v}_i}{\partial \mathbf{c}}\,\boldsymbol{\alpha} = (\log p_{i1}\frac{\partial v_{i1}}{\partial \mathbf{c}_1}^T, \ldots, \log p_{i,N_K}\frac{\partial v_{i,N_K}}{\partial \mathbf{c}_{N_K}}^T)^T \tag{11.11}$$

We can see that $\boldsymbol{\beta} = \frac{\partial \mathbf{v}_i}{\partial \mathbf{c}}\,\boldsymbol{\alpha}$ is a vector of length dN_K. Substituting (11.10) and (11.11) into (11.9), we have

$$\sum_{k=1}^{N_K} \nabla w_{ik} \log p_{jk} = \frac{1}{\sum_{k=1}^{N_K} v_{ik}}\,\left(\boldsymbol{\beta} - \mathbf{w}_i\,(\mathbb{1}^T\,\boldsymbol{\beta})\right) \tag{11.12}$$

Here both $\boldsymbol{\beta}$ and the inner product $\mathbb{1}^T\,\boldsymbol{\beta}$ can be calculated in $\mathcal{O}(dN_K)$. Therefore, by exploring calculation ordering of the terms, the overall computational complexity for ∇E is reduced to $\mathcal{O}(dNN_K)$. As N and N_K increase for large-scale training, utilizing this scheme is crucial.

11.2.3 Approximate Computation

In the L-BFGS iterations, both the entropy and its gradient are evaluated multiple times. As the size of the vocabulary and the number of features increase, the optimization procedure takes considerable amount of time even with the scheme discussed in Section 11.2.2. One way to speed up the optimization to further reduce the computational complexity for the energy and gradient computation. To do this, we first observe that as N_K increases, only a small number of centers will contribute to the sum $\sum_{k=1}^{N_K}$ in both (11.2) and (11.6). This is because a specific feature tends to have large euclidean distances to most of the centers which means that w_{ij} is very small for j's. In this case, for each feature, we can compute the sum only over the active set of centers $\Phi_{i,\mu} = \{j|w_{ij} \leq \mu m\}$, where μ is the parameter controlling the magnitude of approximation. Specifically, with sufficiently large μ, we have equivalently the exact computation since then $\Phi_{i,\mu} = \{1, 2, \ldots, N_K\}$. Generally, with large N_K and small μ, $|\Phi_{i,\mu}| \ll N_K$. This enables fast approximate calculation, if we pre-compute $\Phi_{i,\mu}$. However, as we update the centers, the active set is also altered for each feature. To overcome this, we also update the active set as outer iterations. Specifically, we update the active sets for all the features after a few approximate L-BFGS updates on $\mathbf{c}$. Empirically, we observe that the active set becomes relatively stable (close to those of local minima) after 2-3 outer iterations updates. This motivate the idea of progressively decrease update frequency of the active sets. For instance, for 100 approximate L-BFGS iterations, we update the active set at 10th and 30th iteration respectively. As it is shown in Figure 11.1, we achieve similar convergence as exact computation with such approximation in significantly less amount of time.

11.2.4 Connection to Max-Margin Clustering

In this section, we discuss another application of the entropy optimization and its connection to max-margin clustering. For unsupervised learning, max margin clustering tends to have better generalization as its supervised counterpart support vector machine. In this training scenario, each feature is treated as belonging to a singular class (with only one feature). For each feature i, if the weights w_{ik}'s are scattered over N_K clusters, the entropy will increase. Therefore, minimizing the entropy as defined in the previous section, we tend to refine the centers such that each single feature is close to only a very few of centers. Due to the duality of Vononoi diagram (separating planes) and the cluster centers, the minimization is equivalent to pulling features away from the separating planes, which resembles the mechanism of max margin clustering.

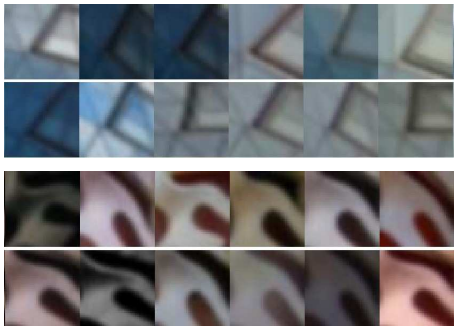


Figure 11.2: Patches of two correspondence classes from the Prague dataset with lighting variation and perspective transformation

11.3 Ground-Truth Dataset

To encode the learned vocabulary with correspondence information, ground-truth data is needed. Specifically, in this work, we focus on local descriptors e.g. SIFT of patches around 3D points of a scene where the correspondences are already extracted from geometric models. A good ground-truth dataset should encapsulate for each 3D point, a set of local descriptors of large appearance variations due to viewing angles or lighting conditions etc. This is crucial for the generalization of the vocabulary learning.

There exist several large datasets with partial matching information e.g. the UBC patch data [99] and Prague patch data [75]. UBC patch data contains three landmarks (Statue of Liberty, Notredame and Yosemite) with approximately 1.5M features of 500K correspondence classes. On average, there are 2-5 features for each class in the UBC patch set. By correspondence class, we mean features that correspond to the same 3D point. On the other hand, Prague patch data consists of 564M features belonging to 111M correspondences classes. Some of correspondence classes in this dataset contains up to 10^2 features, which have high possibility of capturing varieties of the same patch. Therefore, in our work here, we used Prague patch data for experiments. In Figure 11.2, we show features of correspondence classes tracked by graph-based geometry models from [75].

11.4 Experiments

We demonstrate the performance of the entropy formulation in (11.4) in different settings. We compare its performance against widely used k-means. Generally, we evaluate the resulting vocabularies with respect to TPR and FPR. To understand the

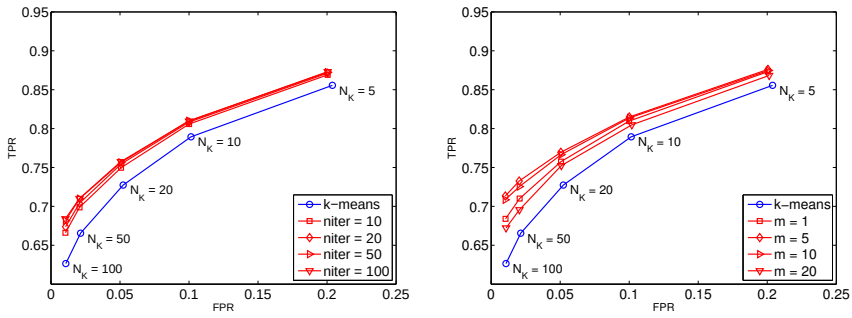


Figure 11.3: The effects of iterations and m for vocabularies of different sizes (N_K = $5, 10, 20, 50, 100$) ($S1$). Left: the effects of number of iterations when m = 5; Right: the effects of m with 100 iterations.

generalization of the method, given a subset S of data with correspondence ground truth, we split data into training set and test set in two ways: ($S1$) for each correspondence class in S, we randomly select 50% of features in that class and include them to the training set, and the others as part of test set; ($S2$) we construct the training set by randomly selecting 50% of the correspondence classes in S (i.e. all features in those classes), and the test set consists of the features in the non-overlapping set of correspondence classes.

For the following experiments, we generate S by randomly picking 20K tracks from the Prague patch set. To guarantee that each correspondence class in the training set has balanced number of features, we limit the number of features in each correspondence class to be 20 to 60. To evaluate the vocabularies, we need to generate matched pairs and non-matched pairs of features. Given the partial ground truth, all distinct pairs of features in the each correspondence class form the matched pairs. And we construct non-matched pairs by randomly pairing up features from two different correspondences classes. The number of possible non-matched pairs is quadratic to the number of correspondence classes. Therefore, we randomly construct $500N$ pairs which should sufficient to avoid bias in estimating FPR.

11.4.1 Parameter Sensitivity

We investigate the effects of different choices of parameters i.e. size of the margin m, the number of iterations in the L-BFGS. For all experiments in this section, we split the data according to ($S1$). Firstly, we would like to understand how the overall performance is affected by the convergence of the optimization. On the left of Figure 11.3, for different N_K, we can see that as we increase the number of iteration from 50 to 100, one only gain very slightly in performance. This suggests that in large-

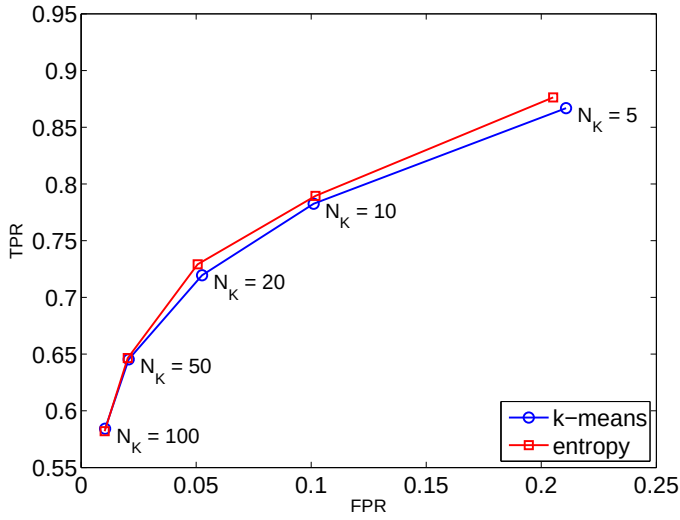


Figure 11.4: Compare entropy-optimized vocabulary with k-means under non-overlapping correspondence classes ($S2$)

scale application, we can trade-off training time without too much loss in performance by limiting the number of iterations. To overcome the local minima, we also try optimization with multiple k-means initialization. In our experiments, we do not gain much improvement with the extra initialization. On the other hand, it turns out that the size of margin m can also affect the performance. In essence, m is dependent on the distribution of the data e.g. the magnitude of the variances within each correspondence class. On the right of Figure 11.3, it can be seen that during testing, one achieves the best performance with $m = 5$ for the data we test on. The inferior performance when $m = 1$ and $m = 20$, can be related to under-smoothing and over-smoothing of the distances to centers, respectively.

11.4.2 Generalization

We can see that for training and test setting $S1$, by encoding matched information into the entropy optimization gives much better performance compared to k-means. To further understand the generalization of the method, we test the method with data split setting ($S2$). In this case, it is expected that the method has more difficulty to generalize. Since in ($S1$), the distribution of each correspondence class is similar both in the training set and test set. However, in ($S2$), the distribution of the correspondence classes in the test set can be very different from those in the training set. In Figure 11.4,

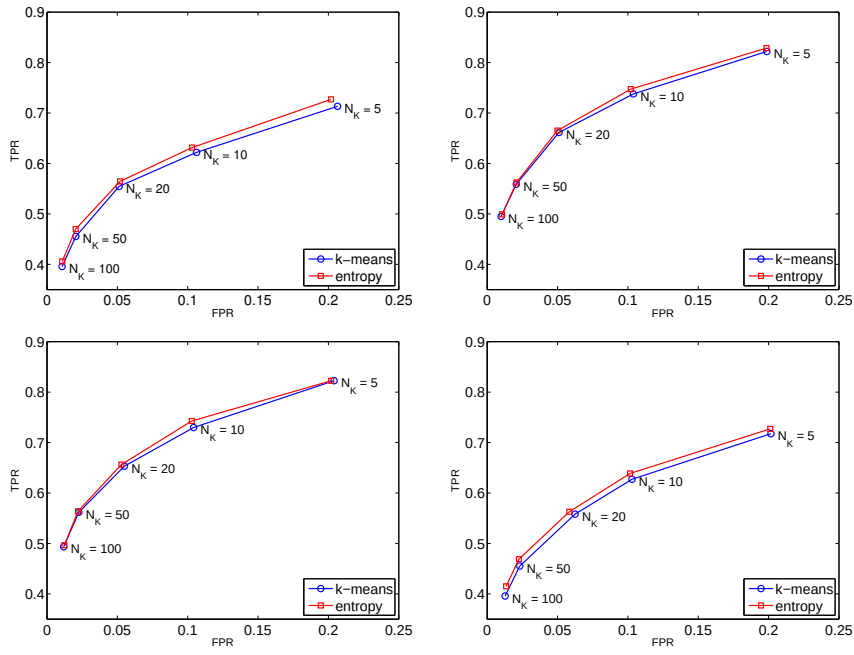


Figure 11.5: Entropy-optimized vocabulary on subspaces $[x_1, ..., x_{32}]$, . . . , $[x_{97}, ..., x_{128}]$ under non-overlapping correspondence classes ($S2$).

we can see that the method only generalize well for vocabularies of small sizes i.e. when $N_K = 5, 10$. Otherwise, for large K, the vocabulary is equivalent or worse than the unsupervised k-means, which is a clear indication of overfitting. This behavior could be explained by the difference of the distributions of training set and test set, as well as the curse of dimensionality.

11.4.3 Optimization over Subspace

To gain better insight of the generalization of the method, we also try entropy optimization on subspace of the SIFT features. The reason we investigate this settings is to see how the dimensionality would alter the behavior of the method. The subspaces we work with are simply subdivision of the 128 dimension of the SIFT feature $\mathbf{x}$ evenly with a factor of s e.g. when s = 32, the subspaces are $[x_1, ..., x_{32}], ..., [x_{97}, ..., x_{128}]$. In Figure 11.5, we can see that the entropy-based method improves over the k-means slightly but consistently for all N_K's for subspaces $[x_1, ..., x_{32}]$ and $[x_{97}, ..., x_{128}]$. On the other hand, for subspaces $[x_{33}, ..., x_{64}]$ and $[x_{65}, ..., x_{96}]$, applying optimiza-

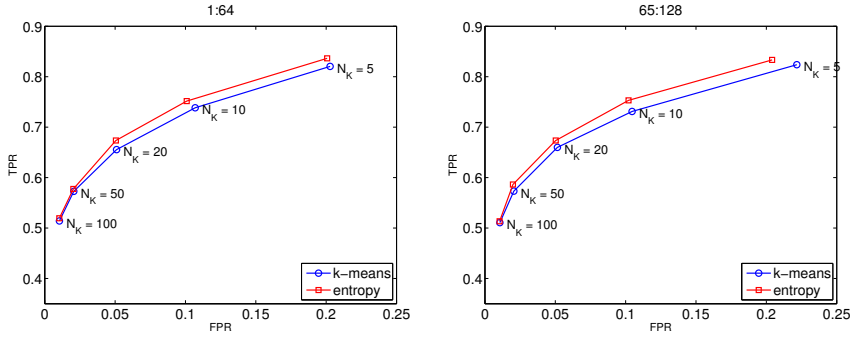


Figure 11.6: Entropy-optimized vocabulary on subspaces $[x_1, ..., x_{64}]$, . . . , $[x_{65}, ..., x_{128}]$ under non-overlapping correspondence classes ($S2$).

Level	k-means	Eopt ($m = 5$)	Eopt ($m = 10$)
3	0.4642	0.4738	0.4743

Table 11.1: Image retrieval performance of hierarchical k-means and entropy-optimized vocabulary.

tion does not gain much improvement against k-means. We have similar observations for different partitions of the SIFT feature spaces (e.g. $s = 8, 16, 64$). Similarly, as seen in Figure 11.6, for the subspace division with 64 dimensions $[x_1, ..., x_{64}]$ and $[x_{65}, ..., x_{128}]$, we still only gain better matching performance for small N_K. This suggests that it is possible to obtain better generalization by reducing the dimensionality of the feature space. However, further investigation is required on better subspace projection than the natural partition here

11.4.4 Image Retrieval

To evaluate the method in a more formal setting, we use the entropy-optimized vocabulary as the quantization step in bags-of-words recognition pipeline . We test the method on the Oxford 5K dataset [84, 85]. The task is to retrieve similar images to the 55 query images (5 for each of the 11 landmarks in Oxford) in the dataset of 5062 images. The performance is then evaluated with mean Average Precision (mAP) score. Higher mAP indicates that the underlying system on average retrieves the similar corresponding images at the top of the ranked list.

In this case, we treat each feature as a correspondence class and the optimized vocabulary will tend to have large margin between features. As an initial evaluation,

to make the entropy-optimization feasible for such large-scale data, we follow the hierarchical k-means strategy. We first applied the hierarchical k-means on the top levels, and use entropy optimization at the last level to reduce the number of features to optimize over. Specifically, we train a vocabulary tree with $L-1$ levels and K splits at each level, at the level L, we apply the entropy optimization. In Table 11.1, we show the retrieval performance of the optimize vocabulary against normal k-means with L = 3, K = 100 ($1M$ words). We can see that the entropy optimization does improve the mAP by approximately 1% with some margin sizes (further increasing m to 20 deteriorate the performance) .The results on the same tree with $L = 2$, show similar performance boost when $m = 5$, but are actually worse when we increase margin to 10 (the word distribution of vocabulary becomes non-uniform). We suggest that such improvement is due to the fact that the entropy optimization increases the margins of the dual separating planes of the k-means centers. In this way, corresponding features (of smaller distances) would have lower probability being separated by separating planes.

11.5 Conclusions

In this chapter, we study and extend the idea of entropy-optimized feature quantization in large-scale training. The approach outperforms the unsupervised k-means when the distribution of training data and test data is similar. However, our experiments show that the gain of the optimization is less obvious due to the difference of the distribution of training and test data. This is related to the high dimensionality of the SIFT features. On the other hand, we explore the resemblance of the entropy optimization and max-margin clustering. By optimizing the entropy on single-feature correspondence class, the method tends to produce quantization that respects both the intra-cluster variation and pair-wise distances. The effectiveness of the idea is verified in image retrieval task.

To improve the generalization of the approach, one idea is to study the optimal subspace projection that enables the better generalization to diverse distributions of training data and test data. Another possibility is to quantized the local features into coarse clusters with k-means, and then apply the entropy optimization on each cluster. In this way, one can ensure similar distributions locally in each cluster, which might then facilitate the generalization of the optimization.

Bibliography

[1] Mongi A. Abidi and T Chandra. A new efficient and direct solution for pose estimation using quadrangular targets: Algorithm and evaluation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(5):534–538, 1995. 56

[2] Erik Ask, Simon Burgess, and Kalle Åström. Minimal structure and motion problems for toa and tdoa measurements with collinearity constraints. In *ICPRAM*, 2013. 107

[3] Erik Ask, Yubin Kuang, and Kalle Åström. Exploiting p-fold symmetries for faster polynomial equation solving. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 3232–3235. IEEE, 2012. 2, 37

[4] K. Åström and A. Heyden. Stochastic analysis of scale-space smoothing. *Advances in Applied Probability*, 30(1), 1999. 65

[5] J. Barreto and K. Daniilidis. Fundamental matrix for cameras with radial distortion. In *IEEE International Conference on Computer Vision*, Beijing, China, 2005. 71, 72, 78

[6] Pratik Biswas, Tzu-Chen Lian, Ta-Chung Wang, and Yinyu Ye. Semidefinite programming based algorithms for sensor network localization. *ACM Trans. Sen. Netw.*, 2(2):188–220, May 2006. 89

[7] E.D. Bolker and B. Roth. When is a bipartite graph a rigid framework. *Pacific J. Math*, 90(1):27–44, 1980. 89, 90, 93, 96, 99

[8] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011. 113

[9] José Henrique Brito, Roland Angst, Kevin Köser, and Marc Pollefeys. Radial distortion self-calibration. 2013. 72

[10] José Henrique Brito, Roland Angst, Kevin Köser, Christopher Zach, Pedro Branco, Manuel Joao Ferreira, and Marc Pollefeys. Unknown radial distortion centers in multiple view geometry problems. In *ACCV 2012*, pages 136–149. Springer, 2012. 72, 73

[11] José Henrique Brito, Christopher Zach, Kevin Köser, Manuel João Ferreira, and Marc Pollefeys. One-sided radial fundamental matrix estimation. In *BMVC*, 2012. 72, 73, 78, 82, 83

[12] Martin Bujnak, Zuzana Kukelova, and Tomas Pajdla. A general solution to the p4p problem for camera with unknown focal length. In *Proc. Conf. Computer Vision and Pattern Recognition, Anchorage, USA*, 2008. 56, 59, 62, 63

[13] Martin Bujnak, Zuzana Kukelova, and Tomas Pajdla. 3d reconstruction from image collections with a single known focal length. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1803–1810. IEEE, 2009. 71

[14] Martin Bujnak, Zuzana Kukelova, and Tomas Pajdla. New efficient solution to the absolute pose problem for camera with unknown focal length and radial distortion. In *Computer Vision–ACCV 2010*, pages 11–24. Springer, 2011. 56, 59

[15] Simon Burgess, Yubin Kuang, and Kalle Astrom. Node localization in unsynchronized time of arrival sensor networks. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2042–2046. IEEE, 2012. 141, 143, 144, 145

[16] Simon Burgess, Yubin Kuang, and Kalle Aström. Toa sensor network calibration for receiver and transmitter spaces with difference in dimension. In *21st European Signal Processing Conference (EUSIPCO 2013)*, 2013. 2, 98

[17] Simon Burgess, Yubin Kuang, Johannes Wendeberg, Kalle Åström, and Christian Schindelhauer. Minimal solvers for unsynchronized tdoa sensor network calibration using far field approximation. In *9th International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics (ALGOSENSORS 2013)*. Springer, 2013. 3, 143

[18] M. Byröd, M. Brown, and K. Åström. Minimal solutions for panoramic stitching with radial distortion. In *Proc. British Machine Vision Conference, London, United Kingdom*, 2009. 25, 30

[19] Martin Byröd, Klas Josephson, and Kalle Åström. A column-pivoting based strategy for monomial ordering in numerical gröbner basis calculations. In *The 10th European Conference on Computer Vision*, 2008. 16, 21, 22, 23, 24, 26, 27, 28, 29

[20] Martin Byröd, Klas Josephson, and Kalle Åström. Fast and stable polynomial equation solving and its application to computer vision. *Int. Journal of Computer Vision*, 84(3):237–255, 2009. 15, 16, 21, 24, 26, 27, 30, 61, 75

[21] Martin Byröd, Zuzana Kukelova, Klas Josephson, Tomas Pajdla, and Kalle Åström. Fast and robust numerical solutions to minimal problems for cameras with radial distortion. In *Proc. Conf. Computer Vision and Pattern Recognition, Anchorage, USA*, 2008. 21, 22, 24, 25, 26, 27, 29

[22] H. Cai, K. Mikolajczyk, and J Matas. Learning linear discriminant projections for dimensionality reduction of image descriptors. *Transactions on Pattern Analysis and Machine Intelligence*, 2010. 150

[23] J.F. Cai, E.J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *Siam Journal of Optimization*, 20(4):1956–1982, 2010. 114

[24] YT Chan and KC Ho. A simple and efficient estimator for hyperbolic location. *Signal Processing, IEEE Transactions on*, 42(8):1905–1915, 1994. 115

[25] Homer H Chen. Pose determination from line-to-plane correspondences: existence condition and closed-form solutions. In *Computer Vision, 1990. Proceedings, Third International Conference on*, pages 374–378. IEEE, 1990. 55

[26] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Springer Verlag, 1997. 14

[27] D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry*. Springer Verlag, 1998. 27

[28] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Springer, 2007. 14, 15

[29] R.-M. Cramer, R. Scholtz, and M. Win. Evaluation of an ultra-wide-band propagation channel. *IEEE Transactions on Antennas and Propagation*, 50(5):561 – 570, May 2002. 133

[30] M. Crocco, A. Del Bue, M. Bustreo, and V. Murino. A closed form solution to the microphone position self-calibration problem. In *37th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2012), Kyoto, Japan*, March 2012. 90, 105

[31] Marco Crocco, Alessio Del Bue, and Vittorio Murino. A bilinear approach to the position self-calibration of multiple sensors. *Trans. Sig. Proc.*, 60(2):660–673, feb 2012. 90, 120

[32] Michel Dhome, Marc Richetin, J-T Lapreste, and Gerard Rives. Determination of the attitude of 3d objects from a single perspective view. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(12):1265–1278, 1989. 55

[33] D. Eisenbud, Daniel R. Grayson, E. Stillman Michael, and Bernd Sturmfels, editors. *Computations in algebraic geometry with Macaulay 2*. Springer-Verlag, 2001. 45, 74, 76, 77, 78

[34] Ricardo Fabbri, BenjaminB. Kimia, and PeterJ. Giblin. Camera pose estimation using first-order curve differential geometry. In *ECCV*, 2012. 55, 59, 68

[35] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–95, 1981. 55, 72, 135

[36] A. W. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *Proc. of Computer Vision and Pattern Recognition Conference*, pages 125–132, 2001. 11, 71, 72, 73

[37] N. D. Gaubitch, W. B. Kleijn, and R. Heusdens. Auto-localization in ad-hoc microphone arrays. In *ICASSP*, 2013. 107

[38] D. Grayson and M. Stillman. Macaulay 2. Available at http://www.math.uiuc.edu/Macaulay2/, 1993-2002. An open source computer algebra software. 61, 93

[39] Daniel R. Grayson and Michael E. Stillman. Macaulay2, a software system for research in algebraic geometry. http://www.math.uiuc.edu/Macaulay2/. 44, 110

[40] R. M. Haralick, C. Lee, K. Ottenberg, and M. Nölle. Analysis and solutions for the three point perspective pose estimation problem. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 592–598, 1991. 55

[41] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 57, 72, 74

[42] Richard Hartley and Sing Bing Kang. Parameter-free radial distortion correction with center of distortion estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(8):1309–1321, 2007. 71

[43] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 9, 10, 41, 79

[44] Richard I Hartley. In defense of the eight-point algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(6):580–593, 1997. 83

[45] Joel A Hesch and Stergios I Roumeliotis. A direct least-squares (dls) method for pnp. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 383–390. IEEE, 2011. 41, 47, 48, 56

[46] Arnold Irschara, Christopher Zach, J-M Frahm, and Horst Bischof. From structure-from-motion point clouds to fast location recognition. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2599–2606. IEEE, 2009. 55

[47] R. Ji, H. Yao, X. Sun, B. Zhong, and W. Gao. Towards semantic embedding in visual vocabulary. 2010. 150

[48] Fangyuan Jiang, Yubin Kuang, and Kalle Åström. Time delay estimation for tdoa self-calibration using truncated nuclear norm. *Proc. of ICASSP*, 2013. 2

[49] Hailin Jin. A three-point minimal solution for panoramic stitching with lens distortion. In *Computer Vision and Pattern Recognition 24-26 June 2008, Anchorage, Alaska, USA*, 2008. 25, 29

[50] K. Josephson and M. Byröd. Pose estimation with radial distortion and unknown focal length. In *Proc. Conf. Computer Vision and Pattern Recognition, San Fransisco, USA*, 2009. 56

[51] K. Josephson, M. Byröd, F. Kahl, and K. Åström. Image-based localization using hybrid feature correspondences. In *The second international ISPRS workshop BenCOS 2007, Towards Benchmarking Automated Calibration, Orientation, and Surface Reconstruction from Images*, 2007. 56

[52] Kevin Köser and Reinhard Koch. Differential spatial resection-pose estimation using a single local image feature. In *ECCV 2008*, pages 312–325. Springer, 2008. 56

[53] Y. Kuang, E. Ask, S. Burgess, and K. Åström. Understanding toa and tdoa network calibration using far field approximation as initial estimate. In *ICPRAM*, 2012. 3

[54] Yubin Kuang and Kalle Åström. Numerically stable optimization of polynomial solvers for minimal problems. In *ECCV 2012*, pages 100–113. Springer Berlin Heidelberg, 2012. 1, 62

[55] Yubin Kuang and Kalle Åström. Numerically stable optimization of polynomial solvers for minimal problems. In Andrew Fitzgibbon, editor, *Lecture Notes in Computer Science*, volume 7574, pages 100–113. Springerm, Heidelberg, 2012. 103

[56] Yubin Kuang and Kalle Åström. Pose estimation with unknown focal length using points, directions and lines. *Proceedings of IEEE International Conference on Computer Vision (ICCV2013)*, 2013. 2

[57] Yubin Kuang and Kalle Åström. Stratified sensor network self-calibration from tdoa measurements. In *EUSIPCO*, 2013. 2

[58] Yubin Kuang, Kalle Åström, Lars Kopp, Magnus Oskarsson, and Martin Byröd. Optimizing visual vocabularies using soft assignment entropies. In *ACCV 2010*, pages 255–268. 2011. 3, 150, 151

[59] Yubin Kuang, Kalle Åström, and Fredrik Tufvesson. Single antenna anchor-free uwb positioning based on multipath propagation. In *Communications (ICC), 2013 IEEE International Conference on*, pages 5814–5818. IEEE, 2013. 3

[60] Yubin Kuang, Simon Burgess, Anna Torstensson, and Kalle Åström. A complete characterization and solution to the microphone position self-calibration problem. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 3875–3879. IEEE, 2013. 2

[61] Yubin Kuang, M Byrod, and K Åström. Supervised feature quantization with entropy optimization. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1386–1393. IEEE, 2011. 3

[62] Yubin Kuang, Jan Erik Solem, Fredrik Kahl, and Kalle Åström. Minimal solvers for relative pose with a single unknown radial distortion. In *Computer Vision and Pattern Recognition, Conference on*. IEEE, 2014. 2

[63] Yubin Kuang, Yinqiang Zheng, and Kalle Åström. Partial symmetry in polynomial systems and its application in computer vision. In *Computer Vision and Pattern Recognition, Conference on*. IEEE, 2014. 2

[64] Z. Kukelova, M. Bujnak, and T. Pajdla. Automatic generator of minimal problem solvers. In *Proc. 10th European Conf. on Computer Vision, Marseille, France*, 2008. 15, 16, 21, 24, 25, 26, 27, 29, 44, 45, 46, 48, 49, 50, 61, 62

[65] Z. Kukelova and T. Pajdla. A minimal solution to the autocalibration of radial distortion. In *In Proc. Conf. Computer Vision and Pattern Recognition*, 2007. 71, 72, 74, 75

[66] Z. Kukelova and T. Pajdla. Two minimal problems for cameras with radial distortion. In *OMNIVIS*, 2007. 25, 26, 29, 71

[67] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Real-time solution to the absolute pose problem with unknown radial distortion and focal length. *Proceedings of IEEE International Conference on Computer Vision (ICCV2013)*, 2013. 56

[68] Zuzana Kukelova, Martin Byröd, Klas Josephson, Tomas Pajdla, and Kalle Åström. Fast and robust numerical solutions to minimal problems for cameras with radial distortion. *Computer Vision and Image Understanding*, 114(2):234–244, 2010. 71, 72, 74, 78, 82

[69] S. Lazebnik and M. Raginsky. Supervised learning of quantizer codebooks by information loss minimization. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 31(7):1294 – 1309, 2009. 150, 151, 152

[70] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International Journal of Computer Vision*, 81(2):155–166, 2009. 56

[71] Hongdong Li and Richard Hartley. A non-iterative method for correcting lens distortion from nine point correspondences. *OMNIVIS 2005*, 2005. 72, 85

[72] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. 7th Int. Conf. on Computer Vision, Kerkyra, Greece*, pages 1150–1157, 1999. 83

[73] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2):91–110, 2004. 149

[74] N. Michelusi, U. Mitra, A. F. Molisch, and M. Zorzi. UWB sparse/diffuse channels, part i: Channel models and bayesian estimators. *IEEE Transactions on Signal Processing*, 60(10):5307 –5319, Oct. 2012. 133

[75] A. Mikulik, M. Perdoch, O. Chum, and J. Matas. Learning a fine vocabulary. In *Proc. 10th European Conf. on Computer Vision, Marseille, France*, 2010. 150, 156

[76] Faraz M Mirzaei and Stergios I Roumeliotis. Globally optimal pose estimation from line correspondences. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5581–5588. IEEE, 2011. 41, 43, 49, 50, 56

[77] F. Moosmann, B. Triggs, and F. Jurie. Randomized clustering forests for building fast and discriminative visual vocabularies. In *Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, Canada*, 2006. 150, 151

[78] Oleg Naroditsky and Kostas Daniilidis. Optimizing polynomial solvers for minimal geometry problems. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 975–982. IEEE, 2011. 24, 25, 26, 30

[79] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, June 2006. 149

[80] Carl Olsson, Fredrik Kahl, and Magnus Oskarsson. Branch and bound methods for euclidean registration problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):783–794, 2009. 42, 48

[81] N. Ono, H. Kohno, N. Ito, and S. Sagayama. Blind alignment of asynchronously recorded signals for distributed microphone array. In *WASPAA'09*, 2009. 107

[82] Adrian Penate-Sanchez, Juan Andrade-Cetto, and Francesc Moreno-Noguer. Exhaustive linearization for robust camera pose and focal length estimation. 2013. 56

[83] F. Perronnin. Universal and adapted vocabularies for generic visual categorization. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(7):1243–1256, 2008. 150

[84] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 149, 160

[85] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 149, 160

[86] J. Philbin, M. Isard, J. Sivic, and A. Zisserman. Descriptor learning for efficient retrieval. In *Proc. 10th European Conf. on Computer Vision, Marseille, France*, 2010. 150

[87] M. Pollefeys and D. Nister. Direct computation of sound and microphone locations from time-difference-of-arrival data. In *Proc. of International Conference on Acoustics, Speech and Signal Processing*, 2008. 89, 100, 107, 108, 110, 116, 117, 120

[88] N.B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller. Anchor-free distributed localization in sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 340–341. ACM, 2003. 89

[89] Srikumar Ramalingam, Sofien Bouaziz, and Peter Sturm. Pose estimation using both points and lines for geo-localization. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4716–4723. IEEE, 2011. 55

[90] T. Santos, J. Karedal, P. Almers, F. Tufvesson, and A. Molisch. Modeling the ultra-wideband outdoor channel: Measurements and parameter extraction method. *IEEE Transactions on Wireless Communications*, 9(1):282 –290, January 2010. 133

[91] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, 2003. 149, 160

[92] H. Stewénius. *Gröbner Basis Methods for Minimal Problems in Computer Vision*. PhD thesis, Lund University, April 2005. 89, 96, 102, 103, 107

[93] H. Stewénius, F. Schaffalitzky, and D. Nistér. How hard is three-view triangulation really? In *Proc. Int. Conf. on Computer Vision*, pages 686–693, Beijing, China, 2005. 43, 94

[94] Christoph Strecha, Alex Bronstein, Michael Bronstein, and Pascal Fua. Ldahash: Improved matching with smaller descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99(PrePrints), 2011. 150

[95] M. Bujnak T. Pajdla, Z. Kukelova. http://cmp.felk.cvut.cz/minimal/index.php. 14

[96] B. Triggs. Camera pose and calibration from 4 or 5 known 3d points. In *Proc. 7th Int. Conf. on Computer Vision, Kerkyra, Greece*, pages 278–284. IEEE Computer Society Press, 1999. 56, 59

[97] Yair Weiss, Antonio Torralba, and Robert Fergus. Spectral hashing. *NIPS*, 9(1):6, 2008. 150

[98] J. Wendeberg and C. Schindelhauer. Polynomial time approximation algorithms for localization based on unknown signals. In *ALGOSENSORS*, 2012. 107

[99] S. Winder and M. Brown. Learning local image descriptors. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR07)*, Minneapolis, June 2007. 156

[100] J. Winn, T. Criminisi, and T. Minka. Object categorization by learned visual dictionary. 2005. 150

[101] G. Young and A.S. Householder. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3(1):19–22, 1941. 89

[102] D. Zhang, Y. Hu, J. Ye, X. Li, and X. He. Matrix completion by truncated nuclear norm regularization. In *CVPR*, pages 2192–2199, 2012. 112

[103] Lilian Zhang, Chi Xu, Kok-Meng Lee, and Reinhard Koch. Robust and efficient pose estimation from line correspondences. In *Computer Vision–ACCV 2012*, pages 217–230. Springer, 2013. 49

[104] Yinqiang Zheng, Yubin Kuang, Shigeki Sugimoto, Kalle Åström, and Masatoshi Okutomi. Revisiting the pnp problem: A fast, general and optimal solution. *Proceedings of IEEE International Conference on Computer Vision (ICCV2013)*, pages 2344–2351, 2013. 2, 41

[105] Z. Zhu, A.M.C. So, and Y. Ye. Universal rigidity and edge sparsification for sensor network localization. *SIAM Journal on Optimization*, 20(6):3059–3081, 2010. 89