Methods for Optimal Model Fitting and Sensor Calibration

Ask, Erik

2014

Methods for Optimal Model Fitting and Sensor Calibration

Improving and Constructing Polynomial Solvers

Erik Ask

Lund University

Faculty of Engineering
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 2014:4
ISSN 1404-0034

ISBN 978-91-7623-096-1 (print), 978-91-7623-097-8 (digital)
LUTFMA-1050-2014

Printed in Sweden by MediaTryck, Lund 2014

Preface

In this thesis we present contributions to polynomial equation solving and model fitting. The contribution to polynomial equation solving is a methodology to simplify state-of-the-art methods for solving systems of multivariate polynomials when the system displays symmetries. In the area of model fitting there are two main topics. The first is methods for optimal model fitting for geometrical problems. We give theoretical results and develop practical algorithms for several problems in computer vision. All of these problems result in polynomial equations and solvers are derived, some of which have symmetry of the type mentioned above. The second is sensor network calibration, being a model fitting problem as well. We identify specific requirements for when the geometric structure of the bipartite sensor networks can be obtained from sensor-to-sensor measurements and derive solvers for these cases.

Papers

The contents of the thesis is based on the following papers

- Ask, Erik and Kuang, Yubin and Åström, Kalle, "Exploiting p-Fold Symmetries for Faster Polynomial Equation Solving", *Proc. International Conference on Pattern Recognition (ICPR)*, 2012.
- Enqvist, Olof and Ask, Erik and Kahl, Fredrik and Åström, Kalle, "Robust Fitting for Multiple View Geometry", *Proc. European Conference on Computer Vision (ECCV)*, 2012.
- Ask, Erik and Enqvist, Olof and Kahl, Fredrik, "Optimal Geometric Fitting Under the Truncated L_2 -Norm", *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)* , 2013.

- Ask, Erik and Enqvist, Olof and Svärm, Linus and Kahl, Fredrik and Lippolis, Giuseppe, "Tractable and Reliable Registration of 2D Point Sets", *Proc. European Conference on Computer Vision (ECCV)*, 2014.

- Enqvist, Olof and Ask, Erik and Kahl, Fredrik and Åström, Kalle, "Tractable Algorithms for Robust Model Estimation", To Appear in *Proc. International Journal of Computer Vision (IJCV)*, 2014.

- Ask, Erik and Burgess, Simon and Åström, Kalle, "Minimal Structure and Motion Problems for TOA and TDOA Measurements with Collinearity Constraints", *Proc. International Conference on Pattern Recognition Applications and Methods* , 2013.

- Ask, Erik and Kuang, Yubin and Åström, Kalle, "A Unifying Approach to Minimal Problems in Collinear and Planar TDOA Sensor Network Self-Calibration", *Proc. European Signal Processing Conference (EUSIPCO)* , 2014.

- Kuang, Yubin and Ask, Erik and Burgess, Simon and Åström, Kalle, "Understanding TOA and TDOA Network Calibration using Far Field Approximation as Initial Estimate", *Proc. European Signal Processing Conference (EUSIPCO)* , 2014.

Acknowledgements

First and foremost, I would like to thank my supervisor Kalle Åström who introduced me to algebraic geometry and its applications in both imaging and sensor networks. In addition to guiding me in my research, he was also the person who first introduced me to a mathematical view of imaging during my undergraduate studies.

Secondly, I would like to thank all my co-authors for wonderful collaborations during these years. In particular my co-supervisor Olof Enqvist for introducing me, and the world, to brand new ideas allowing for better understanding and methods for model fitting. Without his insights a large part of this thesis would not exist.

I also have to thank everyone in the Mathematical Imaging Group and at the Centre for Mathematical Sciences for research input, fruitful discussions, amazing courses, fantastic company at breaks, lunches, and after-work beers, for taking the edges of setbacks and for always making my workplace feel like a second home.

Finally, I would like to thank my family, for their never faulting support in my endeavours, for not letting me slack off when learning was easy, for giving me the confidence to make mistakes and for always being there when I needed them.

Lissabon, September 1, 2014

Funding This work is supported by the strategic research projects ELLIIT and eSSENCE, Swedish Foundation for Strategic Research projects ENGROSS and VINST(grants no. RIT08-0075 and RIT08-0043), Vinnova funded project (Autometa) and Swedish research council project Polynomial Equations and Geometry in Computer Vision.

Contents

Chapter 1

Introduction

The problem of fitting models to measured data has been studied extensively, not least in the field of computer vision. A central problem in this field is the difficulty in reliably finding corresponding structures and points in different images. This difficulty results in large portions of outlier data. This thesis presents theoretical results improving the understanding of the connection between model parameter estimation and possible outlier-inlier partitions of data point sets. Using these results a multitude of applications can be analyzed in respects to optimal outlier-inlier partitions, multi-model fitting and model fitting under truncated norms. Practical polynomial-time optimal solvers are derived for several applications, including but not limited to multi-view triangulation and image registration.

The second major topic in this thesis is self calibration of sensor networks. Sensor networks play an increasingly important role with the increased availability of mobile, antenna-equipped devices. The application areas can be extended with knowledge of the different sensors relative or absolute positions. We study this problem in the context of bipartite sensor networks. We identify solvability requirements for several configurations, and present a framework for how such problems can be approached. Further we utilize this framework to derive several solvers, whose usefulness we show in both synthetic and real examples.

In both these types of model estimation, as well as in the classical random sampling based approaches minimal cases of polynomial systems play a central role. A majority of the problems tackled in this thesis will have solvers based on recent techniques pertaining to action matrix solvers. New application-specific polynomial equation sets are constructed and elimination templates designed for them. In addition a general improvement to the method is suggested for a large class of polynomial systems. The method is shown to improve the computational speed by significant reductions in the size of elimination templates as well as in the size of the action matrices. In addition, the methodology improves the average

numerical stability of the solvers.

1.1 Organisation of the Thesis

Two main application areas have been studied and the thesis is structured to reflect this. Common to both these areas are polynomial solving, and a central regarding improving polynomial solving in general is given already in Chapter 3. This result is utilized in both application areas.

Chapter 2 This chapter gives a brief and in no ways complete background to the problems and concepts that the thesis builds upon.

1.1.1 Polynomial Equation Systems

Chapter 3 This chapter gives additional details on the methodology used to solve the polynomial systems of the later chapters, and introduces the first main contribution of the author; a method for exploiting symmetries in polynomial equations.

1.1.2 Robust and Tractable Model Fitting

Chapter 4 One of the most common problems in computer vision is model fitting in the presence of outliers. In this chapter this problem is analyzed and theoretical results are given that allows for algorithms guaranteeing optimal solutions in polynomial time. The mathematical formulation includes, but is not limited to, the strict inlier-outlier problem.

Chapter 5 The ideas from Chapter 4 are adapted for optimal solutions of several problems in inlier-outlier and truncated norm sense. A historical motivation of this cost function and theoretical results of when the method is applicable are given.

Chapter 6 The problem of image registration that was solved under truncated L_2-norm in Chapter 5, is further analyzed. A fast algorithm for minimizing the L_1-, and truncated L_1-loss for the problem is presented and compared to previous methods in regards to performance and speed.

1.1.3 Sensor Network Calibration

Chapter 7 In this chapter a subclass of sensor networks is studied, specifically networks where one set of sensors reside in a lower dimension than the other. All configurations for which sensor positions can be obtained from pairwise measurements are identified, and solvers are derived for almost all of these.

Chapter 8 Sensor configurations where the measured distances are large in relation to intra distance of the measuring sensors are analyzed. A far field approximation to the so called TDOA problem is introduced.

1.2 Contributions

The per paper contributions of the author are as follows

- Ask, Erik and Kuang, Yubin and Åström, Kalle, "Exploiting p-Fold Symmetries for Faster Polynomial Equation Solving", *Proc. International Conference on Pattern Recognition (ICPR)*, 2012.

  I came up with and developed the main ideas and methods with help from Kalle Åström. I developed and implemented most of the solvers for the experiments. Yubin Kuang was responsible for one of the synthetic experiments and aided with the other solvers. I ran most experiments and wrote the major part of the paper.

- Enqvist, Olof and Ask, Erik and Kahl, Fredrik and Åström, Kalle, "Robust Fitting for Multiple View Geometry", *Proc. European Conference on Computer Vision (ECCV)*, 2012.

  I worked on details on the central theoretical concepts. I was in charge of deriving and implementing most of the equations and polynomial solvers used. I ran most of the experiments and wrote the corresponding parts of the paper.

- Ask, Erik and Enqvist, Olof and Kahl, Fredrik, "Optimal Geometric Fitting Under the Truncated L_2 -Norm", *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2013.

  This is a continuation of the ideas in the previous paper. I participated in formulating and proving the central concepts of the theory. I implemented

a majority of the solvers and ran most of the experiments. All authors contributed equally to writing the paper.

- Ask, Erik and Enqvist, Olof and Svärm, Linus and Kahl, Fredrik and Lippolis, Giuseppe, "Tractable and Reliable Registration of 2D Point Sets", *Proc. European Conference on Computer Vision (ECCV)*, 2014.

  I sketched early version of the central proofs, and refined them together with my co-authors. I ran most experiments and did most of the time and complexity analysis. I wrote most part of the experimental sections of the papers, and some parts of the theoretical section.

- Enqvist, Olof and Ask, Erik and Kahl, Fredrik and Åström, Kalle, "Tractable Algorithms for Robust Model Estimation", To Appear in *Proc. International Journal of Computer Vision (IJCV)*, 2014.

  This is an extended version of the ECCV 2012 paper and CVPR 2013 papers listed above. In addition to contributions stated above, I performed new experiments to establish average runtimes for convergence of our methods and standard methods. In particular the break points at which our methods outperform standard methods are established. I also wrote this part of the paper.

- Ask, Erik and Burgess, Simon and Åström, Kalle, "Minimal Structure and Motion Problems for TOA and TDOA Measurements with Collinearity Constraints", *Proc. International Conference on Pattern Recognition Applications and Methods*, 2013.

  The idea to investigate this problem came from Kalle. I established the requirements for solvable cases and derived all equations and implemented all solvers. I did all the numerical stability experiments, Simon was in charge of the real data experiments. I wrote most of the paper together with Kalle.

- Ask, Erik and Kuang, Yubin and Åström, Kalle, "A Unifying Approach to Minimal Problems in Collinear and Planar TDOA Sensor Network Self-Calibration", *Proc. European Signal Processing Conference (EUSIPCO)*, 2014.

  The idea to study the problems in this paper came from me and Kalle Åström. I dervied the constraints for which configurations of dimension

defficient sensor networks could be calibrated using TDOA measurements. I adapted the rank constraint formulation from a previous paper from Yubin to the dimension defficient case. Me and Yubin Kuang developed and implemented all solvers. I did most of the experiments, both synthetic and real data, together with Kalle Åström

- Kuang, Yubin and Ask, Erik and Burgess, Simon and Åström, Kalle, "Understanding TOA and TDOA Network Calibration using Far Field Approximation as Initial Estimate", *Proc. European Signal Processing Conference (EUSIPCO)*, 2014.

  I contributed mainly to running experiments and some minor implementation work. I also wrote parts of the experimental section.

Other publications

During the course of my graduate studies, I have also contributed to the following paper, which is thematically different from the rest and omitted from this thesis.

- Haner, Sebastian and Svärm, Linus, and Ask, Erik and Heyden, Anders, "Joint Under and Over Water Calibration of a Swimmer Tracking System", Submitted to *Proc. International Conference on Pattern Recognition Applications and Methods.*, 2015.

  Sebastian was responsible for most of the theory and methods. I contributed with discussions about the methodology, I also created a semi automatic system for calibration marker detection, and wrote the corresponding part of the paper.

Chapter 2
Preliminaries

In this chapter we introduce the model fitting problem. For clarity we will limit ourselves to the context of computer vision, but the most concepts translates directly to sensor network calibration. We also introduce a bare minimum of geometry required for computer vision and sensor networks. Finally we give a brief overview of the current state-of-the-art techniques for multivariate polynomial solving.

2.1 Model Fitting

The model fitting problem is at its core fairly straightforward.

Problem 2.1.1 Given a set of data $\mathcal{X}$ can we explain the data as a realization of model M.

In this thesis we are specifically interested in adpating geometrical models to measured data. A geometrical model could be a, e.g., a transformation of point sets, positions of sensors, a 3D structure observed through known cameras. We also have little or no interest in the model selection aspect of mathematical modelling, i.e., we are not so much interested in questions like what model describes the data as much as the question does specifically model M describe it. A very simple example would be given a set of points, can we explain the points as values on a straight line. In the noise-free case one could take two points, find the equation for the line through them and check if all other points coincide with the line. If we are happy with the model, we can e.g. use it to obtain new points, all points could be represented using only a few parameters. Note also that we did not consider other models like sinusoidal, quadratic, cubic, i.e., we did not do model selection.

The example illustrates a common methodology. Assume a model type (line), find a model (i.e., find the line equation), check for consistency. In real applications data is obtained through measurements, and is plagued by measurement errors. In addition, some measurements are inherently bad and should ideally not be considered at all. The later is generally refered to as outliers and they are very common in computer vision. This complicates both the fitting of models and the evaluation of its correctness. This raises the question of how one can identify and remove outliers from the data. While some outliers can easily be discarded there are fringe cases, when removing or keeping a data point is not obvious.

We will adress these issues throughout the thesis. We here give some background as to why outliers are so prolific in computer vision, and describe common ways of dealing with them.

2.1.1 Image Correspondences

Given two images I_1 and I_2, an image correspondence is a pair of points $(x, y) \in I_1$, $(u, v) \in I_2$ that represents the same underlying structure, normally a point in 3D. While this is simple enough to say, autmatically finding such correspondences has proven to be very difficult. The basic problem is that any points in the images has to be identified by their appearance. Naturally two identical images would result in no useful information, and so the appearance between the images must be different. It could be a new modality of an MRI scan, a picture of a 3D structure from another angle. The common approach is typically

- Find points that are likely well-defined spatially, e.g., corners. To avoid multiple almost identical points one usually only selects the most prolific if several candidates are identified too close to each other.

- Extract properties in a local region and vectorize. That is analyze things like color content, gradients, similarities between pixels and quantize the results.

- Compare vectors for different points.

The de-facto standard currently and for the last 10 years is the so called SIFT descriptor [49]. It will be used for all our experiments.

Data Noise

The first concern raised was data that would only almost fit a model. In practice this is almost always the case for all applications, and computer vision is not an exception. The common method of evaluating a datapoint $\mathbf{x}$ is to define a residual function $r(\theta, \mathbf{x})$ that given model parameters θ, measures the conformance. A concrete example could be to measure the reprojection error in an image for the triangulation problem below. In addition one needs a loss function $l(r(\theta, \mathbf{x}))$ that relates the residuals to a cost that we wish to minimize, formally the problem of model estimation then becomes

$$\min_{\theta} \sum_{\mathbf{x_i}} l(r(\theta, \mathbf{x})). \tag{2.1}$$

The by far most common choice of loss function is the L_2-norm.

Outliers

In addition to the small errors described above, there is in computer vision in practice correspondences that are directly erroneous in that the corresponding points are not related. A typical example would be attempting to match images of repetitive structures. Even more concretely, given two images of a typical apartment building, a majority of windows, doors and other structures are not unique. As we will see later some problems are so difficult to obtain matches for at all that the tolerance for what is similar has to be very forgiving, or no matches will occur at all.

2.1.2 Random Sampling

A standard approach when fitting models in the presence of outliers is to randomly select subsets of data, fit a model, and evaluate conformance with the full dataset. This is commonly known as the RANSAC approach and was introduced in [27]. The common criteria for outlier is if the modelling error is larger or smaller than a fixed threshold. There is an advantage of picking as small a set as possible, as this increases the likelihood of all of the points being inliers. If a given dataset has an inlier ratio of w, the probability that among n random points there is at least one outlier is $(1 - w^n)$. If one wants a probability of p that, at least once, only inliers are selected then one must select at least k samples, with k satisfying

$$1 - p = (1 - w^n)^k. \tag{2.2}$$

It is clear that the term $(1 - w^n)$ should be as small as possible, and since $w \leq 1$ we should select n as small as possible.

The RANSAC method has been highly succesful, but gives in general no guarantees on the solution. That is one can not be sure that the resulting outlier set is the smallest possible given the criteria posed.

2.2 Computer Vision

This section will introduce projective geometry, the most common camera model and some of the applications we study in the thesis.

2.2.1 The Projective Plane

A line in a $2D$ plane can be parametrized as $ax + by + c = 0$ and has a natural representative in $\mathbb{R}^3$ by the vector $(a, b, c)^T$. The correspondence between lines and vectors in $\mathbb{R}^3$ is not one-to-one and there is an equivalence relation

$$(a, b, c) \equiv k(a, b, c)\,, \tag{2.3}$$

as scaled versions of a vector correspond to the same line, for any non-zero k. Formally

Definition 2.1. *An equivalence class of vectors under equivalence relation* (2.3) *is known as a homogeneous vector. Any particular non-zero vector* $(a, b, c)^T$ *is a representative of this class.*

A point (x, y) in a $2D$ plane lies on a particular line (a, b, c) if and only if $ax + by + c = 0$ which we can express as $(x, y, 1)(a, b, c)^T = 0$. Here we introduced a representative of the $2D$ point in $\mathbb{R}^3$ by augmenting the point with a 1. Points (x, y) then have a natural representative (kx, ky, k) in homogeneous coordinates. We now introduce the Projective space $\mathbb{P}^2$ as

Definition 2.2. *The projective space* $\mathbb{P}^2$ *is the set of homogeneous vectors in* $\mathbb{R}^3 \setminus (0, 0, 0)$.

The removal of the point $(0, 0, 0)$ should be obvious as it does not describe any line.

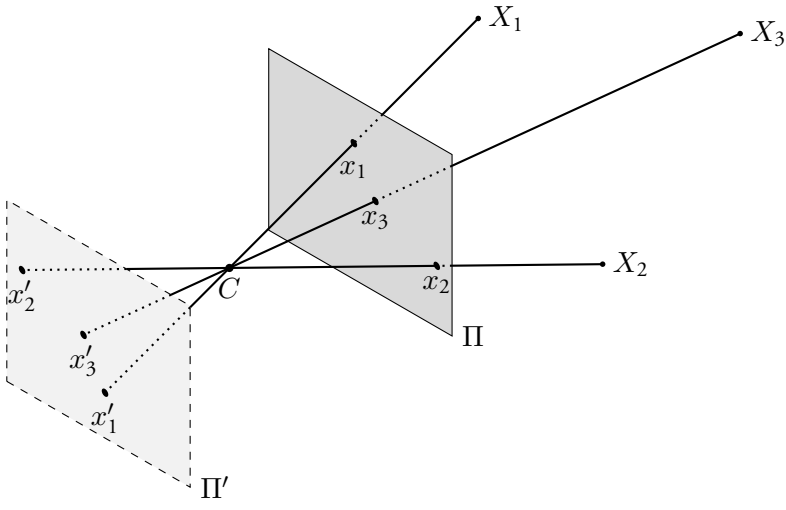


Figure 2.1: The geometry of a pinhole camera. points $\mathbf{X_i}$ in $3D$ are projected through the point C and are intersected with the plane Π to give image points $\mathbf{x}_i$. In a "real" pinhole camera the rays would project onto a plane Π' behind C giving the coordinates $\mathbf{x}_i'$. The images are identical up to a choice of coordinate system.

2.2.2 Pinhole Camera Model

The most common mathematical model of a camera is the pinhole camera model. This model is a representation of an ideal pinhole camera, in which light passes through a tiny hole onto a plane. The principle is illustrated in Figure 2.1. Rays from X_i through C projects onto the plane Π'. For each X_i in $3D$ we get a point x_i' in the $2D$ subspace defined by Π'. It is also clear that intersections between in the plane Π' and intersections in the plane Π will result in the same image up to a rigid transformation. Neither is more accurate than the other but the latter is in general more convenient to work with. The line perpendicular to the image plane through the pinhole, in this case the Z-axis is called the principal axis. Its intersection with the image plane is the principal point and the perpendicular distance between C and Π is the focal length, f.

The intersection of a $3D$ point $(X, Y, Z)^T$ and the plane

$$\Pi = \{(X, Y, Z) \parallel Z = f\},$$

is $(fX/Z, fY/Z, f)^T$. Ignoring the final coordinate the mapping is

$$(X, Y, Z) \mapsto (fX/Z, fY/Z) \ . \tag{2.4}$$

This derivation is straightforward and follows from properties of similar triangles as illustrated in Figure 2.2.

Using homogeneous coordinates for the points $\mathbf{x}_i$ and $\mathbf{X}_i$ the mapping in (2.4) can be written as

$$\begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} \ . \tag{2.5}$$

This expression assumes that the origin of the coordinates in the image plane is the principal point. This is often not the case but is rectified by a translation and the new mapping is

$$(X, Y, Z) \mapsto (fX/Z + p_x, fY/Z + p_y) \ , \tag{2.6}$$

with (p_x, p_y) the coordinates of the principal point. Including the principal point, (2.5) is written

$$\begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} \ . \tag{2.7}$$

If we introduce

$$K = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \tag{2.8}$$

we can reformulate (2.7) as

$$\mathbf{x} = K \begin{bmatrix} \mathtt{I} & \mathbf{0} \end{bmatrix} \mathbf{X} \ . \tag{2.9}$$

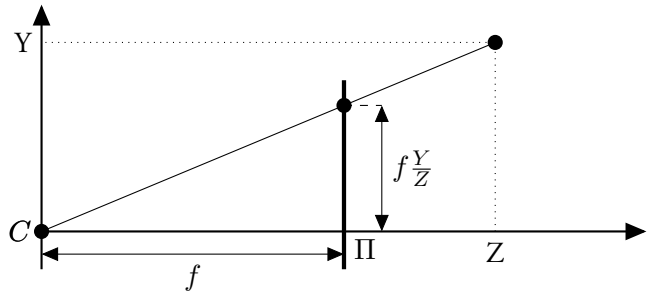


Figure 2.2: Projection of a point $(0, Y, Z)$ into the image plane Π. The image coordinate y is derived from the relation $y/f = Y/Z$.

Here all parameters relating to internal properties of the camera are collected in K, the camera calibration matrix.

Another assumption above is that the camera center lies at the origin. As the coordinate system can be choosen freely this is often a valid and preferable as it simplifies the model. However the model is easily adapted to handle general camera centres. If we have X a point in world coordinate frame, a camera centre at C we can express the point as if we were in a camera coordinate frame as

$$\mathbf{X}_{cam} = R(\mathbf{X} - C) = R\mathbf{X} + t\,. \tag{2.10}$$

This gives us the camera equation as

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = K \begin{bmatrix} R & \mathbf{t} \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = P\mathbf{X}\,. \tag{2.11}$$

Here λ is the scaling factor that normalizes the left hand side so that the homogeneous image coordinate has a 1 as its last element.

One can include additional parameters accounting for non-uniform width and height of pixels in CCD-sensors, and for modelling skew in the sensor. Neither of these will be relevant for the problems studied later. Additionally real cameras have lenses causing radial distortions that are not linear and cannot be described in a linear model.

For a more complete treatise on the pinhole camera model, general projective cameras and homogeneous coordinates see [33].

2.2.3 Epipolar Geometry

Given two uncalibrated cameras P and $\tilde{P}$ with image points $\mathbf{x}_i$, $\tilde{\mathbf{x}}_i$ corresponding to world points $\mathbf{X}_i$, there exist a matrix $\mathbf{F}$ such that

$$\mathbf{x}_i \mathbf{F} \tilde{\mathbf{x}}_i = 0 \; \forall i \, . \tag{2.12}$$

The matrix $\mathbf{F} \in \mathbb{R}^{3\times 3}$ is called the fundamental matrix and satisifes $\det(\mathbf{F}) = 0$ and is only defined up to scale. Conversely any 3×3 matrix with zero determinant is a fundamental matrix for some cameras.

If the camera matrices $\mathbf{K}$ $\tilde{\mathbf{K}}$ are known, coordinates can be pre-normalized as

$$\mathbf{K}^{-1}\mathbf{x}_i = \mathbf{x}'_i = \begin{bmatrix} \mathtt{I} & \mathbf{0} \end{bmatrix} \mathbf{X}_i \tag{2.13}$$

$$\tilde{\mathbf{K}}^{-1}\tilde{\mathbf{x}}_i = \tilde{\mathbf{x}}'_i = \begin{bmatrix} R & \mathbf{t} \end{bmatrix} \mathbf{X}_i \, , \tag{2.14}$$

where the coordinate system is chosen such that the first camera lies in the origin, and the location of the second is determined by a rotation $R \in SO(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$. Then the bilinear constraint in 2.12 can be written as

$$\mathbf{x}'_i \mathbf{E} \tilde{\mathbf{x}}'_i = 0 \, . \tag{2.15}$$

with $\mathbf{E} = [\mathbf{t}]_\times R \in \mathbb{R}^{3\times 3}$ being the essential matrix. This matrix has the properties of $\mathbf{F}$ but has only five degrees of freedom as opposed to seven for $\mathbf{F}$.

2.2.4 Triangulation

Assume that a point $\mathbf{X}_i$ has been captured by two cameras cameras P_1 and P_2, as points $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_i^{(2)}$. If the internal as well as external parameters of the cameras are known, $\mathbf{X}_i$ can ideally be determined as the intersection of the rays from the respective camera centers C_j through the planes Π_j at said points. Due to errors as those described previously the rays will rarely intersect in real applications. One method of finding a good candidate for $\mathbf{X}_i$ could be to find the shortest line between the rays and select the midpoint. This has the drawback that depending on the distance from the respective camera, the reprojected candidate point could end up relatively far from the $\mathbf{x}_i$ in either image. An alternate approach could be to find a candidate point such that the reprojected difference of the point to $\mathbf{x}_i$ is minimized. Geometrically, for a fixed tolerance of the error and assuming euclidean distance, the valid placements of a candidate $\mathbf{X}_i$ point lie in a cone,

defined by the circle around $\mathbf{x}_i$. Several cameras gives several cones and the intersection, if any gives a valid region.

In later chapters we study triangulation in multiple cameras using reprojected error as residuals.

2.2.5 Image Registration

Registration is the problem of transforming data into a common coordinate system. Image registration then aims to relate images so that identical structures in different images are mapped to the same pixel coordinates. Typical applications however are aligning MRI slices, correlating stained tissue samples and similar problems where the aquisition in itself is not projective in nature or performed in such a way that the images can be related by mappings from $\mathbb{R}^2$ to $\mathbb{R}^2$. In many instances the structures do not undergo any or very minor internal change between images, and the problem is reduced to obtain a rotation and translation, i.e., there are only three degrees of freedom. Formally

Definition 2.3. *Rigid image registration is the problem of obtaining a rotation $R \in SO(2)$ and translation $\mathbf{t} \in \mathbb{R}^2$ such that for any two corresponding points $\mathbf{x}$ and $\tilde{\mathbf{x}}$ we have*

$$\mathbf{x} = R\tilde{\mathbf{x}} + \mathbf{t}\,. \tag{2.16}$$

2.3 Bipartite Sensor Networks

A sensor network is any collection of sensors that collect signals or data to facilitate combined analysis. In this thesis we are interested in estimating positions of both sensors and discrete events through time measurements of the events. Other than the implied assumption that all sensors share data, there is no real meaning in the distinction of sensors and events. For our purposes there is no difference if we have handclaps and microphones, speakers and microphones, antenna arrays or any other similar configuration. Therefore in the proceeding discussions and throughout the remainder of the thesis all entities for which we wish to determine positions will be denoted sensors and will be either of type receiver, denoted $\mathbf{r}$, or transmitter, denoted $\mathbf{s}$. In this formulation the sensor network includes both receivers and transmitters.

Such a sensor network can be described by a graph, in which nodes are sensors, and edges are communication paths, an example is given in Fig 2.3. Graphs in

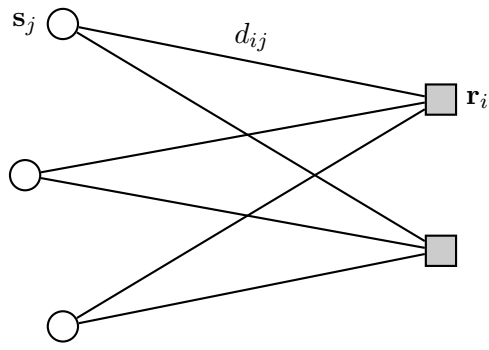


Figure 2.3: A configuration of 3 transmitters (circles) and 2 receivers (squares) with measured pairwise distances d_{ij}.

which edges only exist between two disjoint sets of nodes are called bipartite. As all networks we are interested in have such a representation we have the following definition

Definition 2.4. *A bipartite sensor network with receivers, $\mathbf{r}_i$, and transmitters, $\mathbf{s}_j$, can be represented as a bipartite graph, where the sets $\mathcal{R} = \{\mathbf{r}_i\}$ and $\mathcal{S} = \{\mathbf{s}_j\}$ form the disjoint sets of nodes.*

We define the setting when the distances d_{ij} can be obtained directly as

Definition 2.5. *Time of arrival (TOA) measurements are defined as the time it takes for a signal sent from a specific location $\mathbf{s}$ to arrive at another location $\mathbf{r}$ in a homogeneous medium. The relation between sensor locations r and s and the TOA measurements f is*

$$fv_s = ||\mathbf{s} - \mathbf{r}||_2 = d_{ij}, \tag{2.17}$$

where v_s is the signal speed.

In essence knowledge of f requires a setting in which $\mathbf{r}$ and $\mathbf{s}$ are synchronized in some fashion. There are several scenarios in which this can be achieved; e.g. microphones and speakers connected to a common source, internal clocks and predefined transmission schedule or reflecting signal back to sender.

In scenarios when no common knowledge of when signals are sent and received the TOA formulation is insufficient and needs to be generalized. More precisely

Definition 2.6. *Time difference of arrival (TDOA) measurements are defined as the relative difference in time it takes from a signal sent from a specific location* $\mathbf{s}$ *to arrive at another location* $\mathbf{r}$ *in a homogeneous medium. The relation between locations and measurements* f *is*

$$fv_s = ||\mathbf{s} - \mathbf{r}||_2 + ov_s = d_{ij} + ov_s, \tag{2.18}$$

where v_s *is the signal speed and* o *an unknown offset.*

The name TDOA is motivated by that for two positions $\mathbf{r}_1$ and $\mathbf{r}_2$ and one position $\mathbf{s}$ with associated o one gets

$$\begin{aligned} f_1 v_s &= ||\mathbf{s} - \mathbf{r}_1||_2 + ov_s \\ f_2 v_s &= ||\mathbf{s} - \mathbf{r}_2||_2 + ov_s \\ &\Rightarrow \\ (f_2 - f_1)v_s &= ||\mathbf{s} - \mathbf{r}_1||_2 - ||\mathbf{s} - \mathbf{r}_1||_2 \ , \end{aligned}$$

i.e., the unknown offset is eliminated when taking the difference of measurements.

This still requires synchronization in the sense that all positions $\mathbf{r}_i$ must adhere to a common timeframe, however as all measurements needs to be collected to a common device for any analysis this is in practice usually the case or easily achievable.

For the remainder of the thesis we will, without loss of generality, assume either that $v_s = 1$ or that the conversion from time to distance is already performed and that f has unit meters.

2.4 Minimal Cases

The notion of a minimal case is used throughout this thesis. It is central both in classical outlier eliminiation schemes such as RANSAC and in the proposed optimization methods of Chapters 4 and 5. We formally define a minimal case as

Definition 2.7. *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.*

In practice this means that we have sufficient correspondences to solve for all of the model parameters.

2.5 Algebraic Geometry

A majority of the minimal cases studied in this thesis are described by multivariate polynomial systems or can be posed as such through appropriate choice of parameter space. To solve these we need tools and techniques as well as understanding of algebraic geometry. For a more complete introduction or detailed proofs of the concepts discussed in this section we refer to [19].

2.5.1 Multivariate Polynomial Systems

The basic building block is the monomial which is defined as

Definition 2.8. *A monomial in* $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ *is a product of the form*

$$x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_1^{\alpha_1} \,, \tag{2.19}$$

with $\alpha_i \in \mathbb{N}_0$. *The total degree of a monomial is the sum* $\alpha_1 + \alpha_2 + \cdots + \alpha_n$.

We will use the notation

$$\mathbf{x}^{\alpha} = x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_1^{\alpha_1} \,, \tag{2.20}$$

for monomials and denote the total degree with

$$|\alpha| = \alpha_1 + \alpha_2 + \cdots + \alpha_n \,. \tag{2.21}$$

Using this we get

Definition 2.9. *A polynomial* f *in* $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ *is a finite sum of the form*

$$f(\mathbf{x}) = \sum_{\alpha} c_\alpha \mathbf{x}^{\alpha} \,,\ c_\alpha \in \mathbb{C} \,. \tag{2.22}$$

The set of all such polynomials is denoted $\mathbb{C}(\mathbf{x})$.

The degree of a polynomial is the largest total degree of any of its monomials. The concept of polynomial systems can now be introduced as the following problem

Problem 2.5.1 Given a set of polynomials $f_i(\mathbf{x}) \in \mathbb{C}(\mathbf{x})$ in n variables $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ determine the complete set of solutions to the equation system

$$\begin{aligned} f_1(\mathbf{x}) &= 0 \\ &\vdots \quad . \\ f_m(\mathbf{x}) &= 0 \end{aligned} \tag{2.23}$$

Univariate polynomials is a subclass of multivariate and not a subject of study in this thesis. Unless stated otherwise a system of polynomial equations is always a multivariate system. A system of polynomials can have no, finite or infinite number of solutions. For example the system

$$x_1 - 1 = 0 \tag{2.24}$$
$$x_2(x_1 - 1) + x_1 = 0 \tag{2.25}$$

has no solutions whereas the system

$$x_1 - 1 = 0 \tag{2.26}$$
$$x_2(x_1 - 1) = 0 \tag{2.27}$$

has infinite solutions $\mathbf{x} = (1, t)^T$ for all $t \in \mathbb{C}$.

To solve polynomial system some techniques and results from algebraic geometry are necessary. We start by introducing some central concepts. The zero set (solution set) of a polynomial system defines an affine variety V. As stated in Section 2.4 we are only interested in systems with finite, but non-zero, number of solutions, i.e., systems with a finite variety. Given a polynomial system the set of k polynomials $\{f_i(\mathbf{x})\}$ generates an *ideal* I as

Definition 2.10. *The generated Ideal I of a set of polynomials is*

$$I = \left\{ \sum_{i=1}^{k} h_i(\mathbf{x}) f_i(\mathbf{x}) \ : \ \forall\, h_1, \ldots h_k \in \mathbb{C}(\mathbf{x}) \right\} . \tag{2.28}$$

In particular, by setting

$$h_i = 0 \ , \ \forall i \neq j \tag{2.29}$$

we get that $f_j(\mathbf{x}) \in I$ for all j, i.e., the original equations are part of the ideal.

An ideal is said to be radical if I is identical to the set of all polynomials that vanish on V. Two polynomials f and g are said to be equivalent with respect to I iff $f - g \in I$. With this we define the quotient space as

Definition 2.11. *The quotient space $\mathbb{C}(\mathbf{x})/I$ are all equivalence classes modulo I.*

For a finite and radical ideal I the quotient space $\mathbb{C}/I$ is isomorphic to $\mathbb{C}^r$ where $r = |V|$ is the number of solutions to the set of polynomials that form I. For a proof of this statement as well as a more complete introduction see [19].

2.5.2 The Action Matrix

In this section we will introduce one of the central concepts in solving multivariate polynomial systems, the action matrix. As it can be viewed as a multivariate extension of the companion matrix, we will introduce that first.

Companion matrix

The companion matrix is a construction that allows polynomial systems to be solved using tools from linear algebra and matrix theory. Consider

$$h(x) = x^n + c_{n-1}x^{n-1} + \cdots + c_1 x + c_0, \tag{2.30}$$

a univariate polynomial of degree n with coefficients c_i, $i = 1, n$. In order to find the roots of the polynomial we start from the simple observation that

$$x \cdot x^k = x^{k+1} \ , \tag{2.31}$$

and that $h(x) = 0$ implies that

$$x \cdot x^{n-1} = x^n = -c_{n-1}x^{n-1} - \cdots - c_1 x - c_0 \ . \tag{2.32}$$

If we introduce the vector $\mathbf{b} = [x^{n-1} \ x^{n-2} \ \ldots \ x \ 1]^T$ we can express the relations in 2.31 and 2.32 on matrix form as

$$\underbrace{\begin{pmatrix} -c_{n-1} & -c_{n-2} & \cdots & -c_1 & -c_0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}}_{\mathcal{C}} \underbrace{\begin{pmatrix} x^{n-1} \\ x^{n-2} \\ \vdots \\ x \\ 1 \end{pmatrix}}_{\mathbf{b}} = \underbrace{\begin{pmatrix} x^n \\ x^{n-1} \\ \vdots \\ x^2 \\ x \end{pmatrix}}_{x\mathbf{b}} . \tag{2.33}$$

So any x that fulfills $h(x) = 0$ can be identified as an eigenvalue to $\mathcal{C}$. As all elements of $\mathcal{C}$ are known, solving the polynomial (2.30), is equivalent to solving the eigenvalue problem (2.33).

Multivariate Extension

With some care the technique above can be extended to the multivariate case, as was first done by Lazard in 1981 [45]. Consider the linear map

$$T_a : f(\mathbf{x}) \to a(\mathbf{x})f(\mathbf{x}) \tag{2.34}$$

associated to some $a(\mathbf{x}) \in \mathbb{C}((\mathbf{x})$. Since $|V| < \infty$ we can select a linear basis $\mathcal{B}$ for the space $C(\mathbf{x})/I$. This allows us to represent T_a using a $|V| \times |V|$ matrix $\mathbf{m}_a$. The matrix $\mathbf{m_a}$ is known as the action matrix. The eigenvalues of $\mathbf{m}_a$ are $a(x)$ evaluated on V and the eigenvectors are the elements of $\mathcal{B}$ evaluated on V. Specifically if $[1, x_1, x_2, \ldots, x_n] \subset \mathcal{B}$ the eigenvectors gives the possible solutions to Problem 2.5.1. There are no particular requirements on the polynomial $a(\mathbf{x})$, but for our purposes it will always be a monomial and is refered to as the action monomial.

As a concrete example we return to the polynomial $h(x)$ in (2.30). A representative basis for $\mathbb{C}(x)/I$ is $\mathcal{B} = \{1, x, x^2, \ldots, x^{n-1}\}$. Too see this we first make the observation that any polynomial of degree $n-1$ is expressible in this basis. In particular

$$g(x) = -c_{n-1}x^{n-1} - \cdots - c_1 x - c_0 \tag{2.35}$$

is a polynomial of degree $n-1$. Secondly we consider the the simplest nth degree polynomial $f(x) = x^n$. In this construction it is clear that

$$f - g = h(x) \in I\,, \tag{2.36}$$

and the nth degree polynomial f is equivalent to a polynomial represented in the basis $\mathcal{B}$. The argument can be extended to higher order polynomials in more terms. A suitable choice of action monomial is x, and our linear map then becomes $T_x(x^{k-1}) = x^k$ for $k < n$ and for $k = n$ we get the relation in (2.32). The matrix representation $\mathbf{m}_x$ of T_x is then the matrix $\mathcal{C}$ in (2.33).

2.5.3 Constructing the Action Matrix

While the action matrix describes a relation that, in general, embeds the necessary information for solving a multivariate system, both it and the basis needs to be constructed. One approach is to use Buchberger's algorithm to compute a Gröbner basis for the ideal generated by the system. The obtained basis is used to construct the action matrix. This approach is very slow, and due to round of errors often not even feasible in floating point arithmetics.

The current method of choice is to use a technique called single elimination with basis selection [26, 12, 11]. The central idea is to reduce the system using techniques from linear algebra and matrix theory. The first step is to form a expanded equation system and separate the coefficents and monomials as

$$\mathbf{C}_{\text{exp}}\mathbf{X}_{\text{exp}} = 0 \tag{2.37}$$

with $\mathbf{C}_{\text{exp}}$ the coefficents of an expanded set of equations and $\mathbf{X}_{\text{exp}}$ a vector collecting all monomials among these equations. For reasons that will be obvious in the following discussions the set of original equations should only be expanded with linearly independent equations that evaluate to zero on the variety V. This is accomplished by multiplying the original set of equations with different choices of monomials. For practical reasons often all possible monomials up to a certain degree. The choice of how many equations should be added to the original set is problem specific, but using this methodology the expanded set of equations is a finite subset of the generated ideal of the original equations, and hence the variety is unaltered. Unless relevant we will omit the distinction exp and by

$$\mathbf{C}\mathbf{X} = 0 \tag{2.38}$$

always refer to a expanded set of equations. This formulation allows for elimination of leading terms by methods from numerical linear algebra.

Basis Selection

To construct a suitable basis while eliminations are performed, a basis selection scheme was introduced in [12]. First the set $\mathcal{M}$ of all monomials in $\mathbf{X}$ are partitioned as

$$\mathcal{M} = \mathcal{E} \cup \mathcal{R} \cup \mathcal{P}\,. \tag{2.39}$$

Specifically $\mathcal{P}$ contains the monomials that remain in $\mathcal{M}$ after multiplication with the action monomial $a(\mathbf{x})$ and is called the permissible set. The set $\mathcal{R} = \{a(\mathbf{x})\mathbf{x}^{\alpha_k} \notin \mathcal{P} : \mathbf{x}^{\alpha_k} \in \mathcal{P}\}$ is called reducible and finally the set $\mathcal{E} = \mathcal{M} \setminus (\mathcal{P} \cup \mathcal{R})$ is the excessive set. Using this partition (2.38) can be written

$$\begin{bmatrix}\mathbf{C}_{\mathcal{E}} & \mathbf{C}_{\mathcal{R}} & \mathbf{C}_{\mathcal{P}}\end{bmatrix} \begin{bmatrix}\mathbf{X}_{\mathcal{E}} \\ \mathbf{X}_{\mathcal{R}} \\ \mathbf{X}_{\mathcal{P}}\end{bmatrix} = \mathbf{0}\,. \tag{2.40}$$

The goal is to select $\mathcal{B}$ from the set permissible set $\mathcal{P}$.

First the excessive monomials are eliminated through linear elimination, usually QR-factorization, resulting in

$$\begin{bmatrix}\mathbf{U}_{\mathcal{E}_1} & \mathbf{C}_{\mathcal{R}_1} & \mathbf{C}_{\mathcal{P}_1} \\ \mathbf{0} & \mathbf{U}_{\mathcal{R}_2} & \mathbf{C}_{\mathcal{P}_2} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_{\mathcal{P}_3}\end{bmatrix} \begin{bmatrix}\mathbf{X}_{\mathcal{E}} \\ \mathbf{X}_{\mathcal{R}} \\ \mathbf{X}_{\mathcal{P}}\end{bmatrix} = \mathbf{0}\,, \tag{2.41}$$

with $\mathbf{U}_{\mathcal{E}_\infty}$ and $\mathbf{U}_{\mathcal{R}_2}$ upper triangular. Since the set $\mathcal{E}$ gives us no information on the mapping $T_{a(\mathbf{x})}$ the top rows are removed, resulting in the system

$$\begin{bmatrix}\mathbf{U}_{\mathcal{R}_\in} & \mathbf{C}_{\mathcal{P}_2} \\ \mathbf{0} & \mathbf{C}_{\mathcal{P}_3}\end{bmatrix} \begin{bmatrix}\mathbf{X}_{\mathcal{R}} \\ \mathbf{X}_{\mathcal{P}}\end{bmatrix} = \mathbf{0}\,. \tag{2.42}$$

The final step is to reduce $\mathbf{C}_{\mathcal{P}_\ni}$ into upper triangular form, and from $\mathcal{P}$ select a basis $\mathcal{B}$. This is achieved by column pivoting QR factorization, resulting in a reordering and subsequent split of $\mathbf{X}_{\mathcal{P}}$ into $[\mathbf{X}_{\mathcal{P}'}\ \mathbf{X}_{\mathcal{B}}]$ with $\mathcal{B}$ the last $|V| = r$ elements after reordering. This gives

$$\begin{bmatrix}\mathbf{U}_{\mathcal{R}_\in} & \mathbf{C}_{\mathcal{P}'} & \mathbf{C}_{\mathcal{B}_1} \\ \mathbf{0} & \mathbf{U}_{\mathcal{P}'} & \mathbf{C}_{\mathcal{B}_2}\end{bmatrix} \begin{bmatrix}\mathbf{X}_{\mathcal{R}} \\ \mathbf{X}_{\mathcal{P}'} \\ \mathbf{X}_{\mathcal{B}}\end{bmatrix} = \mathbf{0}\,. \tag{2.43}$$

By a direct re-organization into

$$\begin{bmatrix}\mathbf{X}_{\mathcal{R}} \\ \mathbf{X}_{\mathcal{P}'}\end{bmatrix} = -\begin{bmatrix}\mathbf{U}_{\mathcal{R}_2} & \mathbf{C}_{\mathcal{P}'} \\ \mathbf{0} & \mathbf{U}_{\mathcal{P}'}\end{bmatrix}^{-1} \begin{bmatrix}\mathbf{C}_{\mathcal{B}_1} \\ \mathbf{C}_{\mathcal{B}_2}\end{bmatrix} \mathbf{X}_{\mathcal{B}}\,, \tag{2.44}$$

all actions $T_{a(\mathbf{x})}$ on the basis $\mathcal{B}$ can now be identified and consequently, $\mathbf{m}_a$ can be constructed. For any monomial $\mathbf{x}^\alpha \in \mathcal{B}$ either (i) $a(\mathbf{x})\mathbf{x}^\alpha \in \mathcal{B}$ and the

corresponding entry in $\mathbf{m}_a$ is trivial or (ii) $a(\mathbf{x})\mathbf{x}^\alpha \in \mathcal{R}_2 \cup \mathcal{P}'$ and the entry in $\mathbf{m}_a$ can be extracted from (2.44).

Note here that we have assumed that all operations were possible including the final matrix inversion. If some step fails, we can try generating more equations or try another action monomial or even reformulating the original equations. However, this only has to be once for every problem as these aspects only rely on the structure of the polynomials. The scheme described above is the column-pivoting method from [11] and is the basis for polynomial equation solving in this thesis.

Example: Constructing an Action Matrix

We look at the system

$$x^2 + 3x + y + 1 = 0 \tag{2.45}$$
$$x + y + 9 = 0. \tag{2.46}$$

We expand the lower equation with x, generating a third equation

$$x^2 + xy + 9x = 0. \tag{2.47}$$

Now we have the monomials x^2, xy, x, y and 1. If we decide on the action polynomial y we get 1 and x as permissible and xy and y as reducible and x^2 as excessive.

$$\begin{pmatrix} 1 & 0 & 1 & 3 & 1 \\ 0 & 0 & 1 & 1 & 9 \\ 1 & 1 & 0 & 9 & 0 \end{pmatrix} \begin{pmatrix} x^2 \\ xy \\ y \\ x \\ 1 \end{pmatrix} = 0. \tag{2.48}$$

By QR methods we eliminate the excessive monomials from the other equations

$$\begin{pmatrix} -1.41 & -0.707 & -0.707 & -8.49 & -0.707 \\ 0 & -0.707 & 0.707 & -4.24 & 0.707 \\ 0 & 0 & -1 & -1 & -9 \end{pmatrix} \begin{pmatrix} x^2 \\ xy \\ y \\ x \\ 1 \end{pmatrix} = 0, \tag{2.49}$$

and drop the excessive monomials together with the first equation,

$$\begin{pmatrix} -0.707 & 0.707 & -4.24 & 0.707 \\ 0 & -1 & -1 & -9 \end{pmatrix} \begin{pmatrix} xy \\ y \\ x \\ 1 \end{pmatrix} = 0 \tag{2.50}$$

and set $\mathcal{B} = \{x, 1\}$

$$\begin{pmatrix} -0.707 & 0.707 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} xy \\ y \end{pmatrix} = - \begin{pmatrix} -4.24 & 0.707 \\ -1 & -9 \end{pmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix}. \tag{2.51}$$

This system is solved, yielding

$$\begin{pmatrix} xy \\ y \end{pmatrix} = \begin{pmatrix} -7 & -8 \\ -1 & -9 \end{pmatrix} \begin{pmatrix} x \\ 1 \end{pmatrix}. \tag{2.52}$$

From this we directly read the action on the basis monomials and hence the action matrix

$$y\mathbf{B} = y \begin{pmatrix} x \\ 1 \end{pmatrix} = \begin{pmatrix} -7 & -8 \\ -1 & -9 \end{pmatrix} \mathbf{B} = \mathbf{m_a B}. \tag{2.53}$$

The eigenvalues of this matrix are -5 and -11. The eigenvector corresponding to -5 is $(0.970, -0.243)^\tau$. This should be a rescaled version of the basis monomials at evaluated at a solution, i.e.

$$\begin{pmatrix} x \\ 1 \end{pmatrix} = \lambda \begin{pmatrix} 0.970 \\ -0.243 \end{pmatrix} = \begin{pmatrix} -4 \\ 1 \end{pmatrix}. \tag{2.54}$$

This gives us the first solution $x = -4$, $y = -5$ and by looking at the other eigenvalue we find $x = 2$, $y = -11$.

Part I

Polynomial Equation Systems

Chapter 3

Symmetries in Polynomial Systems

In recent years tackling geometrical problems by polynomial equation solving has shown great results. For instance so called minimal structure and motion problems, e.g. [15, 60], whose solutions are essential for RANSAC algorithms to find inliers in noisy data [29, 65, 66]. These algorithms rely on the ability to efficiently solve a large number of cases in order to find the best set of inliers. There is thus a need for fast and numerically stable algorithms for solving particular systems of polynomials. Once a large enough inlier set is found, local optimization is normally used to fit all data in a least squares sences. Many geometrical problems found in computer vision or sensor networks have solutions that are only unique up to a symmetry. A direct example would be sensor network calibration with no known global coordinate system, described later. In some cases the choice of parametrization itself results in symmetries. One example is the space $SO(3)$ parametrized using unit quaternions.

In this chapter we explore the effect of symmetries in polynomial equation solving. We give a formal definition of what constitutes a symmetry and relate it to desired properties of the eigenvalue problem described in Section 2.5.2. The required modifications to the methods in the previous chapter is introduced in order to exploit the symmetry for improved stability and speed in the derived solvers.

The main contribution are simplifications that can be used (i) if the zero vector is one of the solutions or (ii) if the equations display certain classes of symmetries. Such structures are quite common. We evaluate the simplifications on a few example problems and demonstrate that without losing accuracy, and more commonly improving, significant solver speed improvements are possible.

3.1 P-fold Symmetry

We illustrate the basic principle with the following toy example

$$\begin{cases} x_1^2 - x_2^2 &= 0 \\ x_2^3 x_1 + 1 &= 0 \end{cases}, \tag{3.1}$$

with solutions

$$\begin{array}{rcl} (1,-1) & , & (-1,1), \\ (i,-i) & , & (-i,i), \\ \frac{1}{\sqrt{2}}(1+i,i+1) & , & \frac{-1}{\sqrt{2}}(1+i,1+i), \\ \frac{1}{\sqrt{2}}(1+i,1-i) & , & \frac{-1}{\sqrt{2}}(1+i,1-i). \end{array} \tag{3.2}$$

We see that the solutions are coupled and only differ in sign. If we were to generate new equations using x_1 or x_2 we would get no equations that could be used to eliminate or substitute in the original set. In fact if we multiply with all monomials up to a total degree of n only the equations generated by even monomials are connected to the original set in the context of eliminations as described in Section 2.5.3.

We now give a formal definition of symmetry of a polynomial system.

Definition 3.12. *A system of polynomials $\{f_i(\mathbf{x})\}$ is p-fold symmetric if for all monomials in $\{f_i(\mathbf{x})\}$ the sum of the exponents on $\mathbf{x}$ has the same remainder q modulo p.*

The example in (3.1) is 2-fold as the monomials have degree either 0, 2 or 4. We also note an ambiguity here in that a system with e.g. all monomials of degree 4 would be both 2-fold and 4-fold.

Symmetric system are of interest because of the structure of the solution set. The definition of symmetry in solution set is as follows

Definition 3.13. *A solution set is said to be p-fold symmetric if for each solution $\mathbf{x}^*$ the points $\mathbf{x}_k^* = e^{i2k\pi/p}\mathbf{x}^*$ are also solutions for $j \in \mathbb{Z}$.*

The relation between symmetric polynomial system and symmetric solutions is given by Theorem 3.14.

Theorem 3.14. *A system of polynomial system with p-fold symmetry has a solution set with p-fold symmetry.*

Proof. First assume the polynomial system $\{f_i(\mathbf{x})\}$ is p-fold symmetric. Then for any monomial $\mathbf{x}^\alpha$ in this system, we have

$$|\alpha| \bmod p = q\,. \tag{3.3}$$

For any polynomial $f_i(\mathbf{x}) = \sum_j c_{ij}\mathbf{x}^{\alpha_{ij}}$ the following is true

$$\begin{aligned} f_i(e^{i2k\pi/p}\mathbf{x}) &= \sum_j c_{ij} e^{i2k\pi|\alpha_{ij}|/p}\mathbf{x}^{\alpha_{ij}} \\ &= \sum_j c_{ij} e^{i2k\pi q/p}\mathbf{x}^{\alpha_{ij}} \\ &= e^{i2k\pi q/p}\sum_j c_{ij}\mathbf{x}^{\alpha_{ij}} \\ &= e^{i2k\pi q/p} f_i(\mathbf{x})\,, \end{aligned} \tag{3.4}$$

in particular for any $\mathbf{x}^*$ s.t. $f_i(\mathbf{x}^*) = 0$ we also have $f_i(e^{i2k\pi/p}\mathbf{x}^*) = 0$. This proves the statement. □

3.2 Zero Solutions

In cases where no equation has a constant, one solution is always the zero solution. By necessity this happens when all monomials are of odd total degree, but is not limited to this scenario. Nevertheless one can use the consant 1 as the n'th basis monomial in $\mathcal{B}$. Assume for simplicity that x_1 is chosen as the $n-1$'th basis monomial. Then we obtain an action matrix of type

$$M = \begin{pmatrix} a_{1,1} & \dots & a_{1,n-1} & 0 \\ \dots & \dots & \dots & 0 \\ a_{n-1,1} & \dots & a_{n-1,n-1} & 1 \\ 0 & \dots & 0 & 0 \end{pmatrix}.$$

Since the $(0,\dots,0,1)$ is mapped to $(0,\dots,1,0)$ and since no other reduction involves the constant. From this it follows immediately that $(0,\dots,0,1)^T$ is an eigenvector with eigenvalue 0, i.e. the zero solution. Furthermore any of the $n-1$ eigenvectors to $A = [a_{ij}]$ can be used to produce a corresponding eigenvector to M^T. Thus without loss of generality we can consider the eignevalue problem for the matrix A instead. In practice if there is a 0 solution it can be extracted before solving the full system.

3.3 Solving p-fold Systems

We here present the modifications to the elimination technique described in Section 2.5.3.

3.3.1 Symmetric Action Matrix

As we have established a p-fold symmetric system has p-fold symmetric solution set. This of course implies that among the $|V| = r$ solutions obtainable in the eigenvectors and egienvalues of the action matrix $\mathbf{m_a}$ only $r_p = p/r$ are necessary to find all possible solutions. To exploit p-fold symmetries we aim to construct a mapping $T_{a(\mathbf{x})} : f \mapsto fa(\mathbf{x})$ with $a(\mathbf{x})$ any monomial that is p-fold symmetric with $q = 0$. As the p ambigious solutions collapse into a single point on $a(\mathbf{x})$ the dimension of the solution space effectively becomes r_p, and we denote that basis $\mathcal{B}_r$. As a result $\mathbf{m}_a$ is of size $r_p \times r_p$ instead of $r \times r$. This reduction results in a faster more stable eigenvalue problem when solving the system.

3.3.2 Expanded Equation Set

In the previous section we stated some properties of the modified action matrix and connected basis $\mathcal{B}_r$. However, when determining suitable elements of $\mathcal{B}_r$ by the same methodology presented in 2.5.3 some care is required. As the action matrix we construct only solves the system up to the symmetry, all of the elements of $\mathcal{B}_r$ must be p-fold symmetric. In short this means that no elements of the set $\mathcal{P}$ can contain any monomial that breaks symmetry. By extension as the action monomial $a(\mathbf{x})$ is p-fold symmetric with $q = 0$, all monomials in $\mathcal{R}$ are p-fold symmetric as well. In conclusion, should any non-symmetric monomials exist in the expanded set $\mathcal{M}$ they would by necessity be members of $\mathcal{E}$, the excessive set.

Conversely, supposing the original set is p-fold, generating with any monomial not p-fold symmetric would make all monomials in the new equation break symmetry, and the corresponding entries in a coefficent matrix C would have no column-wise overlap with the original equations, i.e. it could not be used for any reduction as per the operations from Section 2.5.3.

One quickly realises that the only modification necessary is to ensure that no polynomial in the expanded set breaks the symmetry. This is easily achieved by only multiplying the original equations with monomials with p-fold symmetry with $q = 0$. This will in general reduce the size of C, automatically ensure that $\mathcal{P}$ and by extension $\mathcal{R}$ has the desired properties, and that $\mathcal{E}$ is not larger

than necessary. Moreover, a suitable basis for $\mathcal{B}_r$ can be generated using column pivoting as before.

3.3.3 Extracting Solutions

We assume that the system has no 0-solution, either in itself or by the technique described above. While the modified elimination technique reduces the computational load in the elimination scheme and eigenvalue extraction, we have not arrived at the final solution. If we denote the resulting eigenvectors v extracting the solution $\mathbf{x}$ from v can be understood in the following sense. Each element v_i is up to a constant λ equal to the value of a monomial, i.e.

$$\lambda v_i = x_1^{\alpha_{i1}} x_2^{\alpha_{i2}} \dots x_n^{\alpha_{in}}. \tag{3.5}$$

If 1 is one of the monomials we can find the scale factor λ directly. Otherwise we treat it as one of the unknowns by setting $x_{n+1} = 1/\lambda, \quad \alpha_{i,n+1} = 1$ and work with

$$v_i = x_1^{\alpha_{i1}} x_2^{\alpha_{i2}} \dots x_n^{\alpha_{in}} x_{n+1}^{\alpha_{i,n+1}} \tag{3.6}$$

Let A be a matrix with elements $A_{ij} = \alpha_{ij}$. For vectors x of size $a \times 1$ and integer matrices A of size $b \times a$ define the exponential x^A as

$$\mathbf{x}^A := \begin{bmatrix} \mathbf{x}^{\alpha_1} \\ \mathbf{x}^{\alpha_2} \\ \vdots \\ x^{\alpha_b} \end{bmatrix} = \begin{bmatrix} x_1^{\alpha_{11}} x_2^{\alpha_{12}} \dots x_a^{\alpha_{1a}} \\ x_1^{\alpha_{21}} x_2^{\alpha_{22}} \dots x_a^{\alpha_{2a}} \\ \vdots \\ x_1^{\alpha_{b1}} x_2^{\alpha_{b2}} \dots x_a^{\alpha_{ba}} \end{bmatrix}. \tag{3.7}$$

This definition is analogous to the definition of monomials for a vector $\mathbf{x}$ of unknowns, and in particular $\mathbf{x}^I = \mathbf{x}$. As $\mathbf{x}$ takes values in $\mathbb{C}$ the elementwise logarithm of $\mathbf{x}$ is only defined up to a choice of branch. More concretely, for $x_i = r\exp(i\theta)$ one logarithm is $w = \log(r) + i\theta$ and adding multiples of $2\pi i$ gives the others. However since A is an integer matrix,

$$\mathbf{x}^A = \exp(A\log(\mathbf{x})), \tag{3.8}$$

with exp and log elementwise operations, does not depend on the choice of branch. The problem of calculating $\mathbf{x}$ from v can thus be written. Find all $\mathbf{x}$

such that $v = \mathbf{x}^A$. If there exists an integer matrix B, such that $BA = I$, then there is only one solution and it can be written as $\mathbf{x} = v^B$, since

$$\begin{aligned} v^B &= (\mathbf{x}^A)^B \\ &= (\exp(A \log \mathbf{x}))^B \\ &= \exp(B \log(\exp(A \log(\mathbf{x}))) \\ &= \exp(BA \log(\mathbf{x})) \\ &= \mathbf{x}^{BA} = \mathbf{x}^I \,. \end{aligned} \tag{3.9}$$

One way of generating such a matrix is to search for submatrices in A of size $a \times a$ with determinant 1 or -1. Given such a submatrix A_s, its inverse

$$A_s^{-1} = \frac{1}{\det A_s} \operatorname{adj} A,$$

is also an integer matrix and can be used to construct an inverse B as above.

In the general case, one may search for an invertible submatrix A_s, whose absolute value of the determinant is as low as possible. For the p-fold cases we wish to solve, the lowest possible determinant (other than 0) is p, and we have observed in practice that we can always find a submatrix with $p = |\det A_s|$. For this submatrix let

$$B = |\det A_s| \, A_s^{-1} = \pm \operatorname{adj} A_s, \tag{3.10}$$

Then B is clearly an integer matrix and

$$A_s = |\det A_s| \, I = pI. \tag{3.11}$$

This gives

$$v_s^B = (x_s^A)^B = x^{(BA_s)} = x^{pI} \,, \tag{3.12}$$

here v_s is the part of v corresponding to A_s. Now it is possible to solve for x up to an unknown phase of type

$$\mathbf{x} = \mathbf{x}_0 \cdot \exp(ik2\pi/p), \tag{3.13}$$

where k is a $n \times 1$ vector in $\mathbb{Z}_p$. Thus the absolute value of x is well defined and the phase is known up to a p-fold uncertainty. By plugging in this solution in the original equations we obtain

$$\log(v) = A \log(x_0) + A(ik2\pi/p) + ij2\pi, \tag{3.14}$$

which can be written

$$Ak = \underbrace{p(\log(v) - A\log(\mathbf{x}_0))/(i2\pi)}_{z} + pj, \tag{3.15}$$

which can be interpreted as a system of linear equations $Ak = z$ in $\mathbb{Z}_p$. It is straigtforward to write a solver for such problems. In the case of p being prime is particularily simple. The solution has in general 1 free parameter and can be written

$$k = k_0 + sk_1 \ , \text{ with } s \in \mathbb{Z}_p \, , \tag{3.16}$$

which when substituted in (3.13) gives the solutions.

Direct Extraction

While the method from the previous section shows a general method of extracting the solutions under p-fold symmetry, it is often unnecessarily complex. There was a caveat in the description of the action matrix method, that if

$$(1, x_1, x_2, \ldots, x_n) \subset \mathcal{B} \, , \tag{3.17}$$

the solutions could be extracted directly from the basis. The analogous situation for a p-fold system is that the basis $\mathcal{B}_r$ contains elements necessary to uniquely determine the solutions by no or simple arithmetic. An example could be for a 2-fold system in x and y a basis such that

$$(1, x^2, xy) \subset \mathcal{B}_r \, . \tag{3.18}$$

then given v the position corresponding to 1 is used to normalize $\bar{v} = v/\|v\|$, after that one solves $x^2 = \bar{v}_i$ and for the possible solutions solves $xy = \bar{v}_j$, with i and j the elements of v corresponding to x^2 and xy respectively. In our experience finding this type of elements have always been possible.

3.4 Experimental validation

In this section, we test our proposed methods on both synthetic and real problems. We compare with the technique from [11].

3.4.1 Applications

We give here some examples where p-fold symmetries occur in real problems.

Problem 3.4.1 Panoramic stitching with unknown focal length and radial distortion, presented and solved in [10]. This system is symmetric only in the focal length f and it is sufficient to set $f^2 = \tilde{f}$ and use a standard solver to exploit the symmetry.

Problem 3.4.2 Estimating positions on a line given relative distance measurements to points in space, from Chapter 7 where it is used to solve structure from sound problems. It can be shown that the minimal case systems only include monomials of odd total degree and no constants.

Problem 3.4.3 Estimating rotations from 3 correspondences at a preselected residual, from Chapter 4. This system contains only monomials of even total degree, including constants.

3.4.2 Synthetic 2- and 3-fold systems

We first study a 2-fold system with synthetic examples:

$$\begin{cases} x_1^4 + c_{12}x_2^2 + c_{13}x_1x_2 + c_{14} &= 0 \\ x_2^4 - c_{22}x_2^2x_3^2 + c_{23} &= 0 \\ x_1^2 + c_{32}x_2x_3 + c_{33} &= 0 \end{cases}, \tag{3.19}$$

where c_{ij} are chosen randomly in $[0.2, 1.2]$. This system has 16 solutions by analysis with Macaulay. For the standard solver, 1092 equations in 796 monomials are needed to find stable solutions. For our p-fold solver, only 234 equations in 210 monomials are required giving a 6 times speed up. We use the root mean square error of our solutions in the equations 3.19 as residuals. Solving 100 systems gives 1600 evaluations and the per evaluation difference is shown in Figure 3.1. Approximately 85 percent of the times the p-fold solver performs better. On average the $\log_{10}(res)$ was -11.1 for the standard and -12.1 for the p-fold solver.

We also study the 3-fold system

$$\begin{cases} x_1^3 + c_{12}x_2^2x_3 + c_{13}x_1x_2x_3 &= 0 \\ x_2^3 - c_{22}x_1x_3^2 &= 0 \\ x_3^3 + 1 &= 0 \end{cases}, \tag{3.20}$$

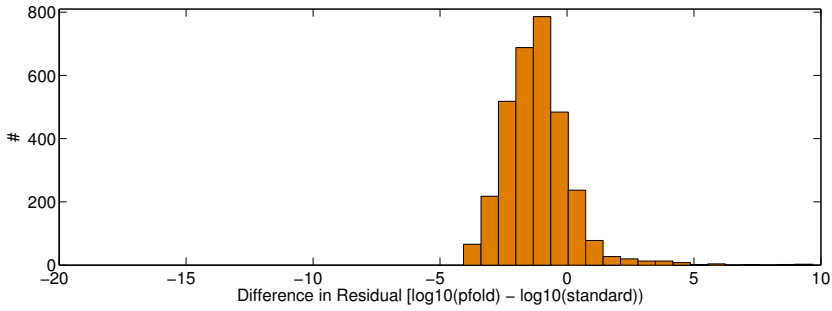


Figure 3.1: Performance difference for the System 3.19

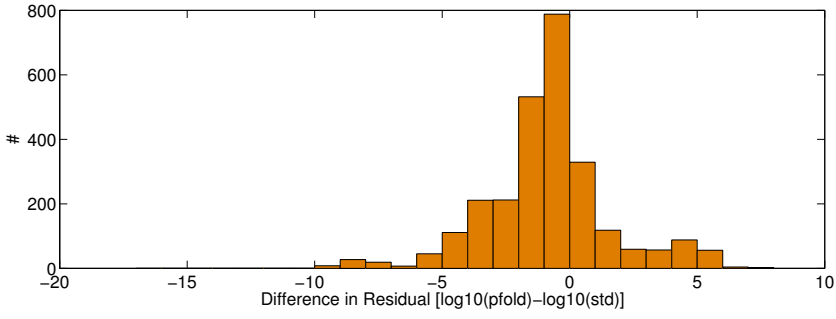


Figure 3.2: Performance difference for system 3.20

where c_{ij} are chosen randomly in $[0.2, 1.2]$. This 3-fold system has 27 solutions. For the standard solver a total of 660 equations and 455 monomials were required for the solver to consistently find solutions. For the p-fold solver 282 equations and 185 monomials were sufficient giving a speedup of a factor 5. With residuals as above and 100 systems, the per evaluation difference is shown in 3.2. In about 75 percent of the times the p-fold solver has better performance. On average the $\log_{10}(res)$ was -9.3 for the standard and -10.2 for the p-fold solver.

3.4.3 Real applications

For the real applications we generate examples of Problems 3.4.2 and 3.4.3 and compare solutions to ground truth. In the case of 3.4.2 we limit us to the cases with 5 points on a line and 2 off line. Performance for a standard solver and a pfold solver is shown in Figure 3.3. On average the standard solver has residuals

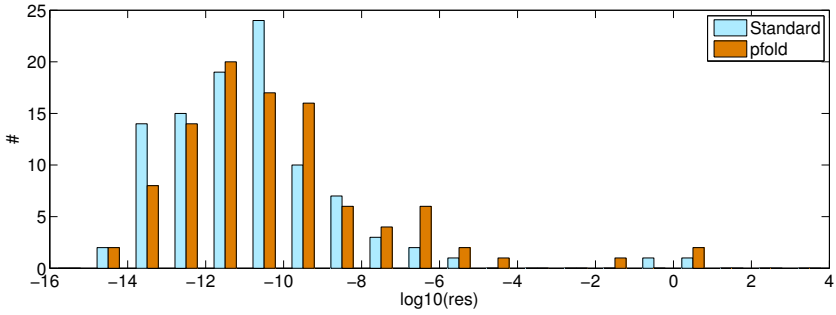


Figure 3.3: Performance solving the Problem 3.4.2.

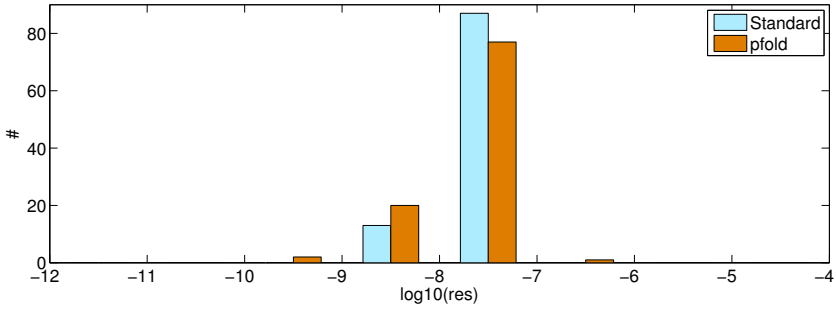


Figure 3.4: Performance solving Problem 3.4.3.

0.31 orders of magnitude less. The size of the standard solver is 1008 equations and 486 monomials, for the p-fold solver 648 equations in 312 monomials and it is 3.5 times faster.

For the Problem 3.4.3 we use the quaternion q for our parametrization of the rotation matrix R_q and get the system

$$\begin{cases} x_i R_q \tilde{x}_i - \cos(\epsilon) &= 0 \quad , \; i = 1, 2, 3 \\ \|q\|^2 - 1 &= 0. \end{cases} \tag{3.21}$$

Which is quadratic in q, four equations for four unknowns and x_i, $\tilde{x}_i$ vectors to map. Comparison results for 100 such examples are shown in figure 3.4. The accuracy is essentially unchanged. The problem size was reduced from 280 equations and 210 monomials to 184 equations and 130 monomials. Speedup is a factor 1.7 .

3.5 Conclusions and Remarks

In this chapter we have shown methods for easily reducing problem sizes for whole classes of polynomial systems. We show significant speed increases on both real problems and synthetic equation sets, without sacrificing numerical stability. In addition the specifics of implementation has changed barely at all, and consists of only a modification in how systems are expanded, the final size of the action matrix, and some extra steps in extracting the final solution from the eigenvectors.

As it turns out, several of the problems studied in subsequent chapters have polynomials that are p-fold symmetric, and in such cases solvers utilizing these techniques have been derived.

Part II

Robust and Tractable Model Fitting

Chapter 4

Outlier-Inlier Partitions on the Model Parameter Space

Many important problems in computer vision, such as structure from motion and image registration, involve the estimation of model parameters given measured point coordinates. As described in Chapter 2 matching such points is prone to outliers and data noise. In this chapter we present important results on the connection between the model parameter space and the possible inlier-outlier partitions. In particular we will show that, for many problems, all possible inlier-outlier partitions can be found in polynomial time. While we leave the application of this theory to later chapters, we give a motivating example for why this understanding is important.

Figure 4.1 shows two consecutive slices of prostate tissue with different stainings routinely used to diagnose cancer. To utilize the information obtained from the different stainings, the two images first have to be registered. As the images originate from different tissue slices, the local structure may look quite different in different images. Therefore, in order to obtain at least some correct correspondences, the acceptance threshold for the feature detector must be generously set. The downside is that this produces a considerable amount of false correspondences. The example in Figure 4.1 is the most difficult case in a database of 88 image pairs. More than 97% of the matches are outliers. There is little hope of finding a solution using a standard RANSAC approach. And as we will see in Chapter 5 and 6 that even if a solution is found, it is often not optimal.

The results are quite general and can be applied to a diversity of model fitting problems. From a practical perspective, our algorithms are not computationally attractive for high-dimensional model estimation. Even though the worst-time computational complexity is polynomial, one should bear in mind that a high-degree polynomial grows quickly and for such problems, one is forced to rely on

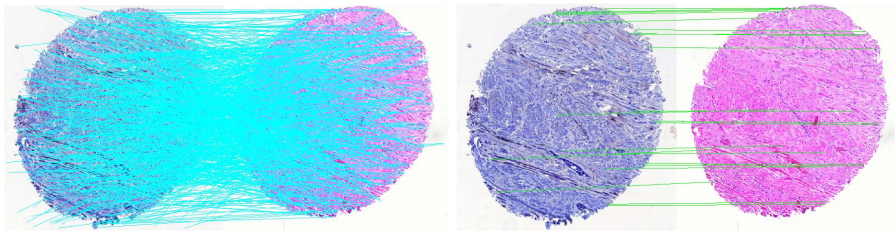


Figure 4.1: (Left) Images of two different stainings from a prostate biopsy with 791 hypothetical correspondences (cyan) obtained from SIFT. (Right) The same images, with 21 inlier correspondences (green) of the optimal truncated L_2-fit. The running time of our MATLAB implementation is 3s. See text for details.

more heuristic approaches.

4.1 Related work

The most frequently used methods for dealing with outliers are based on RANSAC [28]. The basic assumption is that all minimal inlier-samples will produce an acceptable solution. However, it has been empirically observed that this is not valid in practice [46] and therefore numerous improvements of the basic technique have been proposed. For example, guided sampling methods have been proposed in [64, 17] and a strategy for local model refinement was developed in [46]. RANSAC and its variants have also been applied to multi-model estimation [70, 63]. Other robust loss functions have also been investigated, for example, M-estimators in [50] where an iterative algorithm is given. Still, the fact remains that the estimators have no guarantee of finding the optimal solution.

Recent work have focused on computing an optimal estimate based on branch-and-bound. In [24], a robust estimator for camera pose is proposed and in [48], a formulation based on mixed integer programming is given. Similar ideas are presented in [5] for line clustering and vanishing point detection. One of the first approaches in this line of research was given by [8] for low-dimensional matching problems. These methods do not depend on initialization and converge to a global optimum. However, as they are based on branch-and-bound, the run times of the algorithms are unpredictable.

A different paradigm that has turned out to be useful in the context of multiple-

view geometry is the L_∞-framework [39]. In [40], a heuristic relaxation method is proposed to remove outliers. In [58], it is shown how to detect outliers but the method tends to remove a lot of inliers as well. In [47], a triangulation method is presented, but it is only practical for a few outliers due to its high computational complexity. Further extensions have been explored in [54, 69]. The approach works well for large-scale problems with few outliers, but cannot handle large rates of outliers.

The most similar works to ours are [14, 53] where the aim is to develop algorithms which provably minimize the number of outliers. In [14], registration problems dealing mainly with 2D transformations are considered. In the case of linear constraints on the transformation space (for example, when optimizing over translations), it is concluded that in order to obtain the optimal solution, one only needs to examine the intersections of the constraints. This is equivalent to what our approach boils down for this specific setting. However, in the case of non-linear constraints (more specifically, in the case of rotations), specialized solutions are proposed which do not apply to the general setting that we consider. For quasiconvex residual functions, an $\mathcal{O}(n^{d+2})$ algorithm is presented in [53], where n is the number of points and d the dimension of the model. We improve on this result by showing it is possible to solve the same problem in $\mathcal{O}(n^{d+1})$. Further, our result holds for a larger set of residual functions and we can handle other cost functions than the cardinality of the inlier set.

In the case of multiple models, the above methods (optimal or not) for estimating a single model can be applied sequentially. The sequential (or greedy) approach removes all correspondences that are deemed inliers for the most dominant model, and then the process is repeated. This, however, will not produce the optimal solution [70]. The approach may even produce "phantom" solutions [68]. Several heuristics have been proposed to overcome such artefacts, though there is no guarantee of optimality.

4.2 Preliminaries

We will work with a set of n residual functions

$$S = \{r_i : D \to \mathbb{R}_+\}_{i=1}^n \tag{4.1}$$

defined on a d-dimensional manifold, D. These functions describe how some model fitting error depends on model parameters $\theta \in D$. In general we will look

for model parameters that minimize some robust loss function of the residuals; cf. Problem 4.2.1.

Problem 4.2.1 Given a set of residual functions $S : D \to \mathbb{R}$ and a loss function ℓ, estimate a model $\theta \in D$ such that

$$\sum_{r_i \in S} \ell(r_i(\theta)) \tag{4.2}$$

is minimized.

The simplest choice of a loss function is the number of outliers, also known as the *zero-one* loss function, since it can be written

$$\ell(r) = \begin{cases} 0 & \text{if } r \leq \epsilon, \\ 1 & \text{otherwise.} \end{cases} \tag{4.3}$$

Later on we will consider other loss functions as well, but even so the notion of inliers and outliers will be crucial.

Definition 4.15. *Given model parameters θ and a threshold ϵ, an* outlier *is a residual function such that $r_i(\theta) > \epsilon$ and an* inlier *is one such that $r_i(\theta) \leq \epsilon$.*

Example. Given two sets of points in $\mathbb{R}^2$, $\{x_i\}$ and $\{y_i\}$, we want to find a rotation

$$R = \begin{pmatrix} \theta_1 & -\theta_2 \\ \theta_2 & \theta_1 \end{pmatrix} \text{ and a translation } t = \begin{pmatrix} \theta_3 \\ \theta_4 \end{pmatrix} \tag{4.4}$$

mapping one point set to the other. Here D is a 3-dimensional manifold that we embed in $\mathbb{R}^4$ using the equality constraint,

$$h(\theta) = \theta_1^2 + \theta_2^2 - 1 = 0. \tag{4.5}$$

The residuals are simply the Euclidean distances

$$r_i(\theta) = ||R(\theta)x_i + t(\theta) - y_i||. \tag{4.6}$$

Through Definition 4.15, each parameter vector, θ, induces a partition of the set of residuals into inliers, I, and outliers, O. The key to solving many robust fitting problems will be a method to enumerate all feasible partitions into inliers and outliers. One way to approach this is to find a finite set of θ's which is guaranteed to induce all feasible partitions, and this is the path we will pursue. The following definition will be useful.

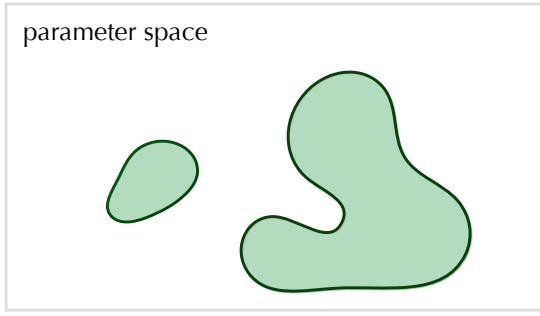


Figure 4.2: Each residual divides the parameter space into two parts. One consisting of θ's such that this residual is an inlier and the complement making it an outlier.

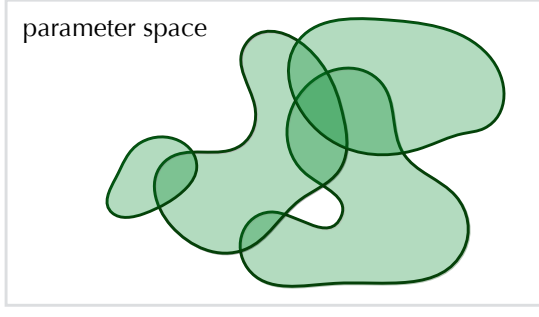


Figure 4.3: If we consider all residuals, then the parameter space is turned into a complex map. To be sure to find all feasible inlier-outlier partitions, we want to find one θ from each section of this map.

Definition 4.16. *Given two subsets, I and O, that partition the residual functions, let $D(I, O)$ denote the set of parameter values $\theta \in D$ such that the residuals in I are inliers and the residuals in O are outliers.*

In terms of Definition 4.16 we want a finite set of θ's to contain at least one point from each non-empty $D(I, O)$. Figures 4.2 and 4.3 illustrate the idea.

Consider a feasible partition, that is, a partition such that $D(I, O)$ is non-empty. In order to specify one θ in this set, we construct and solve a dummy

optimization problem, see the definition below. This problem and the goal function, f, are introduced merely as an analytical tool.

Definition 4.17. *Given a differentiable f, and a partition (I, O) of the residuals in S, we define* DUMMY(I, O) *as the optimization problem:*

$$\min_{\theta} f(\theta) \quad s.t. \quad \theta \in \overline{D}(I, O), \tag{4.7}$$

where the $\overline{D}(I, O)$ refers to the closure of this set.

4.3 Main theorem

We assume that the following technical conditions are fulfilled.

Condition 1. *(a) The domain D is a d-dimensional differentiable manifold embedded in $\mathbb{R}^m$ using $m - d$ polynomial constraints, $h_j(\theta) = 0$. (b) Constraints $r_i(\theta) \leq \epsilon$ can be written as $g_i(\theta) \leq 0$ with g_i being a polynomial defined on $\mathbb{R}^m$. (c) The goal function, f, is a polynomial s.t. $f \to \infty$ as $|\theta| \to \infty$.*

Note that in the subsequent analysis we will be working with the polynomials g_i instead of the residual functions r_i as they may not be originally stated as polynomial functions. The following example enlights why this is necessary.

Example. Recall, for the rigid registration problem we have

$$r_i(\theta) = ||R(\theta)x_i + t(\theta) - y_i||. \tag{4.8}$$

This is, however, not a polynomial in the unknowns, but writing

$$g_i(\theta) = r_i^2(\theta) - \epsilon^2 \leq 0 \tag{4.9}$$

we get the desired polynomial inequality.

As a consequence of Condition 1(b), $r_i(\theta) \geq \epsilon$ can be written $-g_i(\theta) \leq 0$. In the sequel, we will work with g_i instead of r_i. To simplify notation we introduce

$$s_i = \begin{cases} 1 & \text{if } i \in \text{ind}(I) \\ -1 & \text{otherwise,} \end{cases}$$

where $\text{ind}(I)$ denotes the index set of the residual set I. Now we can rewrite $\text{DUMMY}(I, O)$ as

$$\begin{aligned} &\min_{\theta} f(\theta) \\ s_i g_i(\theta) &\leq 0 \quad i = 1, \ldots, n \\ h_j(\theta) &= 0 \quad j = 1, \ldots, m - d. \end{aligned} \tag{4.10}$$

We will show that under mild conditions, the solution to this problem can also be found by analyzing a class of much smaller subproblems. The idea is illustrated in Figure 4.4.

Definition 4.18. *A parameter vector $\theta \in D$ is* critical *to a set of residuals $B \subset S$ if*

$$g_i(\theta) = 0 \quad \forall\, i \in \text{ind}(B) \tag{4.11}$$

and

$$\{\nabla g_i(\theta) : i \in \text{ind}(B)\} \cup \{\nabla f(\theta), \nabla h_1(\theta), \ldots\} \tag{4.12}$$

is linearly dependent and no proper subset of B has this property.

Lemma 4.19. *If $\theta \in D$ is critical to a set of residuals B, then $|B| \leq d$.*

Proof. Assume contrary, that B contains more than d residuals. Then, 4.12 contains $m + 2$ vectors. If we remove one it still contains $m + 1$ vectors in $\mathbb{R}^m$ so it is still linearly dependent. Hence θ was not critical. □

The set B of residuals may be thought of as a *base* set as there is no proper subset to which θ is critical.

An FJ*-point* to an optimization problem is a point that satisfies the Fritz-John conditions for local optimality, see [4]. The conditions are closely related to the more well-known Karush-Kuhn-Tucker (KKT) conditions. For the optimization problem 4.10, a feasible point θ is an FJ-point if there is a non-trivial solution to

$$\mu_0 \nabla f(\theta) + \sum \mu_i s_i \nabla g_i(\theta) + \sum \lambda_j \nabla h_j(\theta) = 0 \tag{4.13}$$

with $\mu_i \geq 0$ and $\mu_i g_i(\theta) = 0$ for all i.

Theorem 4.20. *Let (I, O) be a feasible partition of the residuals and suppose that Conditions 1 are satisfied. Then, (i) $\text{DUMMY}(I, O)$ has at least one* FJ*-point, and (ii) this point is critical to a residual set B of size $\leq d$.*

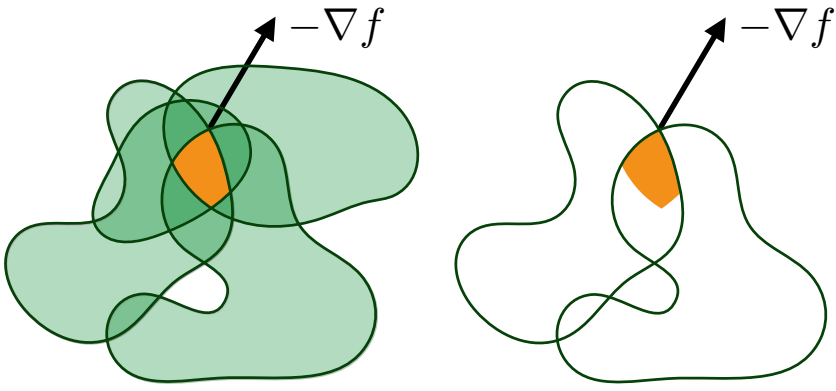


Figure 4.4: The residual functions trace out regions in D and for all θ in such a region, the inlier-outlier partition is constant. In order to find *one* point in the orange region (left), it is enough to consider a subproblem with only $d = 2$ residuals (right).

Proof. Pick any θ_0 in $D(I, O)$. As $f \to \infty$ when $|\theta| \to \infty$, there exists a ρ such that $f(\theta) > f(\theta_0)$ whenever $|\theta| > \rho$. Hence we can restrict the minimization in $\text{DUMMY}(I, O)$ to

$$\overline{D}(I, O) \cap \{\theta : |\theta| \leq \rho\}. \tag{4.14}$$

This set is compact so a minimizer, θ^*, exists. As the involved functions are polynomials, they are continuously differentiable, and by Theorem 4.3.2 in [4], θ^* will satisfy the Fritz-John conditions, which shows the first assertion. It remains to show that θ^* is critical.

As θ^* is an FJ-point, it satisfies

$$\mu_0 \nabla f(\theta^*) + \sum \mu_i s_i \nabla g_i(\theta^*) + \sum \lambda_j \nabla h_j(\theta^*) = 0, \tag{4.15}$$

where $\mu_i \geq 0$ and $\mu_i g_i(\theta^*) = 0$, $i = 1, \ldots, n$. As the h_j's embed a d-dimensional manifold in R^m, the set $\{\nabla h_j\}$ will span a $(m - d)$-dimensional subspace perpendicular to the manifold tangent space at θ^*. Let P be the projection operator onto this tangent space. By projecting 4.15 we get

$$\mu_0 P \nabla f(\theta^*) + \sum \mu_i s_i P \nabla g_i(\theta^*) = 0. \tag{4.16}$$

Elementary linear algebra tells us that we only have to use at most d of the projected vectors $P\nabla g_i(\theta^*)$ which have non-zero μ-coefficients. Let $\text{ind}(B)$ be a smallest such index set (and corresponding residual set B) so that we can write

$$\tilde{\mu}_0 P\nabla f(\theta^*) + \sum_{i\in\text{ind}(B)} \tilde{\mu}_i P\nabla g_i(\theta^*) = 0, \tag{4.17}$$

where the coefficients $\tilde{\mu}_i$'s include the s_i's as well, thus they may be negative. It follows that

$$\tilde{\mu}_0 \nabla f(\theta^*) + \sum_{i\in\text{ind}(B)} \tilde{\mu}_i \nabla g_i(\theta^*) \tag{4.18}$$

is perpendicular to the tangent space of D at θ^* and hence there exist $\tilde{\lambda}_j$'s such that

$$\tilde{\mu}_0 \nabla f(\theta^*) + \sum_{i\in\text{ind}(B)} \tilde{\mu}_i \nabla g_i(\theta^*) + \sum \tilde{\lambda}_j \nabla h_j(\theta^*) = 0. \tag{4.19}$$

From the construction of B, it also follows that $g_i(\theta^*) = 0$ for all $i \in \text{ind}(B)$, and we can conclude that θ^* is critical to B. □

Note that requirement that f, g_i and h_j are polynomials can be replaced with a smoothness constraint, see [4] for details. However, it is only for polynomials that we will have a simple way to find the critical points.

4.4 Finding all critical points

The previous sections establish a link between critical points and the possible inlier-outlier partitions. This is to little use unless the number of critical points to a given set of residuals is small, and we have means to compute them. According to Theorem 4.20, to find the critical points we can go through all subsets, B, of the residuals with $|B| \leq d$. The number of such sets is

$$\sum_{k=0}^{d} \binom{n}{k} = \mathcal{O}(n^d), \tag{4.20}$$

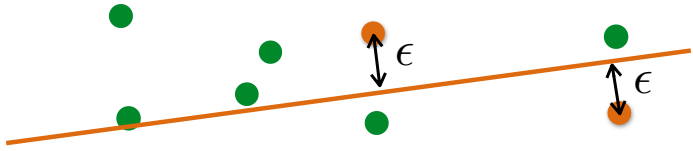


Figure 4.5: A line which is a critical point as two residuals equal ϵ.

where d is the dimension of the parameter manifold. If $|B| = d$, we can find the critical points by solving the system of polynomial equations

$$\begin{aligned} g_i(\theta) &= 0 \quad \forall\, i \in B \\ h_j(\theta) &= 0 \quad j = 1, \ldots, m - d. \end{aligned} \tag{4.21}$$

Bezout's theorem tells us that the number of solutions is either infinite or bounded by the product of the degrees of the polynomials. If the polynomials are in *general position*, then the latter is true. If residuals are in some sense random, then the probability of degeneracy is negligible, but more about this soon.

Example. Given a set of points $\{x_i\}$ in the plane $\mathbb{R}^2$, we want to fit a line to these points. The line can be parameterized by $\theta \in \mathbb{R}^3$ with the embedding constraint

$$h(\theta) = \theta_1^2 + \theta_2^2 - 1 = 0. \tag{4.22}$$

Then, the point to line distance is given by

$$r_i(\theta) = |x_{i,1}\theta_1 + x_{i,2}\theta_2 + \theta_3|. \tag{4.23}$$

This can be turned into a polynomial inequality constraint by setting $g_i(\theta) = r_i(\theta)^2 - \epsilon^2 \leq 0$. A typical critical point θ^* is illustrated in Figure 4.5 for a base set of size two.

There can also be less than d active constraints. In that case, we need to add the constraint that

$$\{\nabla g_i(\theta) : i \in \text{ind}(B)\} \cup \{\nabla f(\theta), \nabla h_1(\theta), \ldots\} \tag{4.24}$$

is a linearly dependent set. This can be enforced using determinants which will also result in polynomial equations. Once again, the number of solutions is either infinite or bounded by the product of the degrees. In this case we should choose f to avoid infinite number of solutions. We state this as an assumption.

Condition 2. *For any subset of $\leq d$ residuals, the set of critical points is finite.*

In order to compute the critical points for a given subset B with $|B| \leq d$, we need to solve a (small) system of polynomial equations. We are only interested in the real solutions of this system, so any complex solutions can be discarded. This means that for certain subsets B there may be no real solutions, but from Theorem 4.20, we know that there will be at least one critical point in one of the subsets.

Note that the number of polynomials in the system is independent of the number of residual functions n (only dependent on d and m), so the computational complexity is $\mathcal{O}(1)$ as both m and d are regarded as fixed constants. In the same way, also note that any valid choice of goal function f will not affect the complexity.

Example. A detailed analysis of Condition 2 is a rather delicate matter, similarly to the analysis of critical configurations in structure from motion estimation [38]. In practice, a probabilistic understanding of the risk of degeneracy is normally sufficient. Once again taking the registration problem as an example, consider three pairs of corresponding points (x_1, y_1), (x_2, y_2) and (x_3, y_3). We expect degeneracy if the pairwise distances are exactly equal in the two images,

$$|x_i - x_j| = |y_i - y_j|. \tag{4.25}$$

If the process used to generate points has some special structure, for example, if all coordinates are integers, then the probability of this occuring is non-zero and our analysis is not sufficient to prove optimality. But in the standard case of interpolated feature point positions, the probability of degeneracy is neglible.

A Speed-Up. The critical points are the real-valued solutions to a set of polynomial equations. Sometimes it might be possible to easily determine that the system has no real-valued solutions. Hence we can avoid running the (costly) algorithm for computing the solutions. The following example illustrates the idea.

Example. Recall the problem of registering two point sets, $\{x_i\}$ and $\{y_i\}$. Assume we want to find the critical points to

$$B = \{r_1, r_2, r_3\}. \tag{4.26}$$

If, for example,

$$||x_1 - x_3|| - \epsilon > ||y_1 - y_3|| + \epsilon \tag{4.27}$$

then there will not be any real-valued solutions to the system of equations, $r_1(\theta) = r_2(\theta) = r_3(\theta) = \epsilon$. That is, there are no critical points to this set.

More generally, we can use the following simple result.

Lemma 4.21. *If there are no critical points to B, then there are no critical points to $\bar{B} \supset B$.*

Proof. Consider the set

$$\{\theta : r_i(\theta) \leq \epsilon\,, \quad \forall\, r_i \in B\}. \tag{4.28}$$

Using the argument from Theorem 4.20, f will have a minimizer in this set and this minimizer is a critical point. Hence for B not to have any critical points this set has to be empty. Clearly the corresponding set for $\bar{B}$ is empty as well so $\bar{B}$ cannot have any critical points. □

To find all critical points we need to go through all subsets B of the residuals in S with $|B| \leq d$. The above lemma shows that we should start with the smallest B's and note whenever a real-valued solution is missing.

4.5 Enumerating all partitions

Our main theorem links each feasible inlier-outlier partition to a critical point, but it is still not clear how to find the partitions linked to a critical point. As usual in computational geometry special cases arise if the data is not in *general position*. In our setting, general position refers to the FJ-points.

Condition 3. *(a) If θ is a critical point to a set of residuals B, then there are no active residuals outside B. (b) If θ is a critical point, then the gradients of the active constraints are linearly independent.*

For most problems there exist degenerate cases with respect to these conditions. The task for the user is to verify that these configurations have neglible probability. Even the rare degenerate cases are solvable, but in order to avoid a too detailed and technical analysis, these cases are deferred to Appendix 4.7.

As shown previously, for a feasible partition of the residuals (I, O), there exists an FJ-point which is critical to a subset $B \subset S$ with $|B| \leq d$. Now, we are interested in the reverse task: Given some θ^* which is critical to a set B, find all partitions (I, O) such that θ^* is also an FJ-point to $\text{DUMMY}(I, O)$. If Condition 3 holds, there will be a unique partition satisfying this.

Step 1. For θ^* to be an FJ-point it has to be feasible, i.e., it has to lie in $\overline{D}(I, O)$. Hence we compute $g_i(\theta)$ for all i. If $g_i(\theta^*) < 0$ then we know that residual i belongs in I and if $g_i(\theta^*) > 0$ then it belongs in O. By Condition 3 all residuals outside B will have $g_i(\theta^*) \neq 0$ and hence this step will label all residuals outside B as either inliers or outliers.

Step 2. As θ^* is critical

$$\gamma_0 \nabla f(\theta^*) + \sum_{i \in \text{ind}(B)} \gamma_i \nabla g_i(\theta^*) + \sum_{j=1}^{m-d} \lambda_j \nabla h_j(\theta^*) = 0, \tag{4.29}$$

where both γ_i's and λ_j's might be negative. Condition 3(b) says that the constraint gradients are linearly independent so $\gamma_0 \neq 0$. Hence after rescaling

$$\nabla f(\theta^*) + \sum_{i \in \text{ind}(B)} \gamma_i \nabla g_i(\theta^*) + \sum_{j=1}^{m-d} \lambda_j \nabla h_j(\theta^*) = 0. \tag{4.30}$$

By Definition 4.18, removing one of the ∇g_i's will create a linearly independent set. This means that all γ_i's must be non-zero. Moreover, linear independence means that all coefficients are unique. Therefore, we can compute the gradients and then uniquely solve for γ_i's and λ_j's.

Step 3. Now, 4.30 looks like the constraint on an FJ-point, except that some of the γ_i's might be negative. In fact the sign of the γ_i's tells us whether to put residual i in I or in O. If γ_i is negative, we add residual i to the outlier set O and set $s_i = -1$. If γ_i is positive then we add residual i to the inlier set I and set $s_i = 1$. Thus

$$\nabla f(\theta^*) + \sum |\gamma_i| s_i \nabla \tilde{g}_i(\theta^*) + \sum \lambda_j \nabla h_j(\theta^*) = 0. \tag{4.31}$$

Comparing with 4.13, we see that θ^* is a FJ-point for $\text{DUMMY}(I, O)$.

Algorithm 1 gives an overview on how to compute all feasible partitions.

Theorem 4.22. *If Conditions 1-3 hold, then Algorithm 1 finds all partitions in $\mathcal{O}(n^{d+1})$-time.*

Algorithm 1 Enumerating all partitions

For each subset B of the residuals with $|B| \leq d$
  Compute all critical points to B (see Section 4.4).
  For each critical point θ^*
    Set $I = \emptyset$.
    For each $g_i \notin B$
      If $g_i(\theta^*) < 0$, add g_i to I.
    Compute the γ_i's of 4.30.
    For each $g_i \in B$
      If $\gamma_i > 0$, add g_i to I.
  Store the partition (I, O), where $O = S \setminus I$.

Proof. The number of subsets of size $\leq d$ is $\mathcal{O}(n^d)$. For a given problem the number of critical points to a given S is constant w.r.t. n, so the total number of critical points is also $\mathcal{O}(n^d)$. For each of these we need to compute $g_i(\theta^*)$ for $i = 1, \ldots, n$. Hence the total complexity is $\mathcal{O}(n^{d+1})$. □

As a consequence, the complexity of the outlier minimization will be $\mathcal{O}(n^{d+1})$. Note that for a given problem, d is normally fixed and the complexity is polynomial. In theoretical computer science, such problems are called *fixed-parameter tractable* with respect to the dimension of the parameter space. In contrast, when both n and d are regarded as input, the above complexity bound yields worst-case exponential running times. In fact, with this viewpoint, the class of problems is NP-hard as it includes, for example, the densest hemisphere problem, known to be NP-complete [37]. When d is regarded as a constant (as in our setting), the densest hemisphere problem is fixed-parameter tractable since it can be solved in $\mathcal{O}(n^d \log n)$ [37].

From a practical point of view, note that we are solving a fixed number of identical subproblems. This means that the exact running time can be predicted and that the algorithm can be easily parallelized by distributing the outer for-loop of Algorithm 1 over multiple cores.

4.6 Choice of goal function f

In practice, we want to choose f that yields simple polynomial equations. Typically, with parameter manifold embedded in R^m the best we can do is to choose f as a linear function on $\mathbb{R}^m$. However, a linear goal function does not satisfy Condition 1(c). To work around this we need harder constraints on the residual functions.

Condition 4. *If I is non-empty, then $D(I, O)$ is bounded.*

Example. It is not hard to show that this condition holds for registration. Given a corresponding point pair (x_i, y_i), assume that the translation $||t|| > ||x_i|| + ||y_i|| + \epsilon$. By the triangle inequality

$$||Rx_i + t - y_i|| \geq ||t|| - ||Rx_i|| - ||y_i|| > \epsilon. \tag{4.32}$$

Hence the translation is bounded as long as there is at least one inlier, and as the rotation part has unit length this shows that $D(I, O)$ is bounded.

We needed Condition 1(c) at the start of the proof of Theorem 4.20 to ensure that there exists a minimizer for f over the closure of $D(I, O)$. With the stronger constraint on the residuals, it follows that $D(I, O)$ is bounded and hence any continuous f will have a minimizer.

4.7 Handling degeneracies

Essentially, Condition 3 states that constraints are in some sense random. For example, it will normally not happen that more than d residuals will be exactly equal to ϵ at the same θ—and numerical accuracy will make it impossible to securely detect it. Having said that, assume that the problem has some special structure which makes degeneracies possible (and which makes it possible for us to detect them). With a little extra work we can still compute all inlier-outlier partitions. As we are not aware of any applications with this type of degeneracy we will only briefly sketch the solution.

Consider a critical point, θ_c, such that Condition 3 does not hold. We want to find all partitions (I, O) that such that $\theta_c \in \overline{D}(I, O)$. Place a ball around θ_c such that all other critical points are outside this ball. We will now show that this $\overline{D}(I, O)$ intersects this ball for all the desired partitions.

Assume the contrary, that there exists a partition (I, O) such that $\overline{D}(I, O)$ (or at least a subset of it) is completely contained in this ball. Then that set is compact and hence f has both a maximizer and a minimizer on this set. These two cannot coincide and they will both be critical points. Hence there is another critical point inside the specified ball, which is a contradiction.

A strategy to find the partitions at θ_c is to enumerate all the partitions on the specified ball. If necessary this process can be iterated.

Chapter 5

Optimal Model Fitting

The main contribution of the previous chapter was an algorithm that allowed all possible inlier-outlier partitions to be enumerated. In this chapter we will show how this result can be used to

- Solve the model fitting problem in strict inlier-outlier sense.
- Solve multiple model fitting in strict inlier outlier sense.
- Solve the model fitting problem under the truncated L_2 loss function for a large class of problems.
- Solve the model fitting problem approximately under any loss function, not limited to but including truncated L_2.

It is widely accepted that the truncated L_2-norm is a good way to model noise and outliers [7], but its use has been hindered by the difficulty in solving the resulting optimization problem.

All algorithms and techniques introduced are extensively evaluated on real problems in computer vision and image registration. In particular we show that the proven theoretical guarantees on the optimality of our solution translates to better solutions in practice. To better understand the tradeoff in regards to time complexity we analyze the practical running times, and compare the performance of our minimal solvers to standard exact solvers in a RANSAC scheme.

5.1 Outlier minimization

Explicitly enumerating all possible inlier-outlier partitions allows us to find the partition which minimizes the number of outliers. In fact, a slightly simplified algorithm can be applied, see Algorithm 2.

Algorithm 2 Outlier minimization

For each subset B of the residuals with $|B| \leq d$
  Compute all critical points to B.
  For each critical point θ^*
    Compute all $g_i(\theta^*)$.
    Count the outliers $g_i(\theta^*) > 0$.
    If this is the lowest number so far, store θ^*.

Theorem 5.23. *If Conditions 1-3 hold, then Algorithm 2 finds an optimal solution θ^* that minimizes the number of outliers in $\mathcal{O}(n^{d+1})$ time.*

In conclusion, we have solved Problem 4.2.1 with the zero-one loss function. Let us turn our attention to another loss function.

5.2 Multiple models

In many cases the data is well-described by a small set of low-dimensional models. Perhaps the best example application is motion segmentation. It is clear that this setting is significantly harder than single-model fitting. Therefore we will only consider the zero-one loss

$$\ell(r) = \begin{cases} 0 & \text{if } r \leq \epsilon, \\ 1 & \text{otherwise.} \end{cases} \tag{5.1}$$

Problem 5.2.1 defines the problem more precisely.

Problem 5.2.1 Let D be a d-dimensional differentiable manifold, embedded in $\mathbb{R}^m$ ($m \geq d$). Given a set of residual functions S, estimate a set of k models $\{\theta_1, \ldots, \theta_k\}$ with $\theta_j \in D$, such that

$$\sum_{i=1}^{n} \min_{j=1,\ldots,k} \ell(r_i(\theta_j)) \tag{5.2}$$

is minimized.

This corresponds to minimizing the number of outliers. Let us first note that the previous results also hold in the multi-model case, as fitting k d-dimensional models can be viewed as fitting one kd-dimensional model. However, we get a more practical approach by using the following result.

Theorem 5.24. *Assume that we would like to estimate k models and that Conditions 1-3 are satisfied. Then, an optimal solution to Problem 5.2.1 consists of k critical points to residual sets with size $\leq d$.*

This means that among the *same* critical points as for the single-model case, we can form an optimal solution to the multi-model case by picking a subset of k such critical points.

Proof. As the goal function in (5.2) can only attain a finite number of values, a minimizer is guaranteed to exist. Consider such a minimizer, $\{\theta_1^*, \ldots, \theta_k^*\}$. We will study θ_1^* more closely, but note that the discussion holds for any θ_j^*. Let I_1 and O_1 be the set of inliers and outliers, respectively, to θ_1^*. Note that a residual can be an inlier to more than one model, but this will not matter. Consider the set $D(I_1, O_1)$ as in Definition 4.16. Clearly this set is non-empty since it contains θ_1^*. Moreover, for any θ in this set, $\{\theta, \theta_2^* \ldots, \theta_k^*\}$ is a solution to Problem 5.2.1. By Theorem 4.20 at least one point in $D(I_1, O_1)$ is critical to a set of $\leq d$ residuals. This argument can be repeated for indices $2, \ldots, k$. □

The maximum k-cover problem. Assume that we have computed all critical points to residual subsets of size $\leq d$. Recall that the number of such points is $\mathcal{O}(n^d)$. The remaining problem is to choose k of these hypotheses such that Problem 5.2.1 is solved. Each hypothesis (or critical point) can be represented by its set of inliers I_i and we want to find a maximum k-cover,

$$\max_{|C|=k} \Big| \bigcup_{i \in C} I_i \Big|. \tag{5.3}$$

This is a well-known NP-hard problem, but it is easy to see that it is fixed-parameter tractable with respect to the number of models. More precisely, let H be the set of hypotheses, and assume that we want to fit k models. Since there are only $\binom{|H|}{k}$ possible choices, an exhaustive search can be done in polynomial time as long as k is fixed. Normally, a more efficient solution is to formulate the max k-cover problem as an integer linear program, and use standard solvers for this type of problem.

In summary, in order to optimally estimate a set of k models (Problem 5.2.1), one can use Algorithm 3. The worst-case running time is $\mathcal{O}(n^{kd+1})$.

Algorithm 3 Multi-model estimation

For each base set B of at most d residuals.
  Compute all critical points to B.
  For each critical point θ^*,
    Save the index set $\{ i \,|\, r_i(\theta^*) \leq \epsilon \}$.
Solve MAX k-COVER for the family of index sets.

Maximal Sets. A set of inliers is maximal with respect to an instance of a fitting problem if it is not a true subset of any other inlier set to that instance. It is easy to see, that only maximal inlier sets need to be considered in the maximum covering problem. This also means that the number of maximal sets is crucial to the complexity of multi-model fitting rather than the total number of inlier sets. One might wonder if this is a significant difference. From a computational complexity perspective, the answer is no.

5.3 Model Estimation under Truncated L_2-norm

5.3.1 Noise Modelling.

Minimizing the number of outliers is a simple approach that generally yields good results, but it does have its limitations, see [33]. One problem is that the method might be sensitive to the choice of ϵ, but also that the distribution of the inlier errors is not modelled. In [7] a more refined loss function is proposed. The assumption is that inlier residuals have a bell-shaped error distribution similar to the Gaussian distribution, whereas outlier residuals have approximately uniformly distributed errors. These assumptions lead to the loss function

$$l(r) = -\log\left(c + \exp\left(-r^2\right)\right) \tag{5.4}$$

where r is the residual error, see Figure 5.1. It is also noted that a good approximation can be obtained by truncating the ordinary squared error.

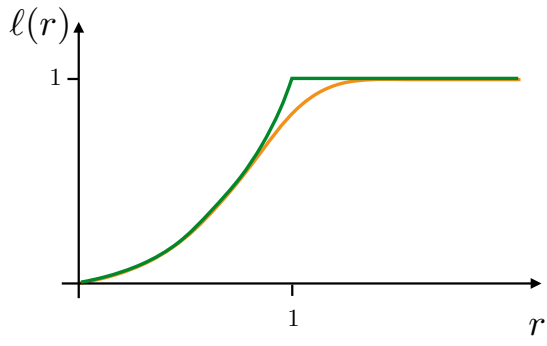


Figure 5.1: A robust loss function (orange) as suggested in [7], and the truncated L_2-error (green) which can be optimized using the proposed framework; in this case truncated at $\epsilon = 1$.

5.3.2 Truncated L_2-norm.

For a residual r we compute the loss as

$$\ell(r) = \begin{cases} r^2 & \text{if } r \leq \epsilon, \\ \epsilon^2 & \text{otherwise.} \end{cases} \tag{5.5}$$

Lemma 5.25. *A minimizer θ^* to Problem 4.2.1 under the truncated L_2-loss function (5.5) is also a global minimum to*

$$\min_{\theta} \sum_{r_i \in I} r_i^2(\theta), \tag{5.6}$$

where $I = \{r_i : r_i(\theta^*) \leq \epsilon\}$.

Proof. Assume to the contrary there exists a θ' such that

$$\sum_{r_i \in I} r_i^2(\theta^*) > \sum_{r_i \in I} r_i^2(\theta'). \tag{5.7}$$

Let O denote the set of residuals that are not in I. If we add $\epsilon^2|O|$ to the left hand

side, then we get the total loss at θ^*,

$$\sum_{i=1}^{n} \ell(r_i(\theta^*)) > \sum_{r_i \in I} r_i^2(\theta') + \epsilon^2 |O| \geq$$

$$\sum_{r_i \in I} \ell(r_i(\theta')) + \sum_{r_i \in O} \ell(r_i(\theta')) = \sum_{i=1}^{n} \ell(r_i(\theta')), \tag{5.8}$$

which is a contradiction. □

This lemma shows that minimizing the truncated L_2-loss is closely linked to minimizing the standard L_2-loss. If the standard L_2-loss is easy to minimize we can also handle the truncated loss. Simply compute all possible inlier-outlier partitions and for each inlier set, compute the optimal L_2-solution(s). Therefore, we can apply Algorithm 1 in order to enumerate all partitions.

5.3.3 Using approximated norms

The previous section shows how to optimize the truncated L_2-norm, but only given an efficient algorithm for optimizing the standard L_2-norm. In many cases no such algorithm exists. Then it is not possible to optimize the truncated L_2-norm exactly but we can approximate that or other loss functions with a piecewise constant function (see Figure 5.2) and optimize this instead. As for standard outlier minimization, the loss only changes value when one of the residuals r_i passes a threshold ϵ_k, see Figure 5.3.

To see how we can optimize a piecewise constant loss, consider Figure 5.4. It shows how the parameter space is divided in a case with two residual functions and a loss function with three discontinuities. Just like before, we want to sample each of the different regions in this plot. We can use Algorithm 1 for this, if we simply treat each combination of residual function and a threshold as a separate function. More precisely, for each residual function r_i, $i = 1, \ldots, n$ and threshold ϵ_k, $k = 1, \ldots, \kappa$, we introduce a new function $r_{i,k} = r_i - \epsilon_k$. Then we have a set of residual functions and we want to sample all feasible inlier-outlier partitions for these functions. This can be done with Algorithm 1. As the number of input functions is $n\kappa$, the number of critical points that are generated in Algorithm 1 is $\mathcal{O}((n\kappa)^d)$. To optimize the piecewise constant loss, we simply evaluate it for each of these critical points. The total complexity of this approach is $\mathcal{O}((n\kappa)^{d+1})$, where κ is the number of discontinuities of the loss function.

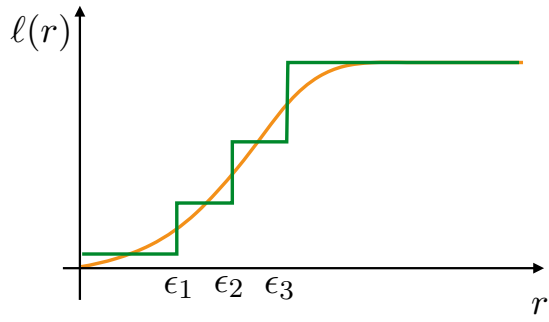


Figure 5.2: A general loss function (orange) is approximated with a piece-wise constant loss function (green).

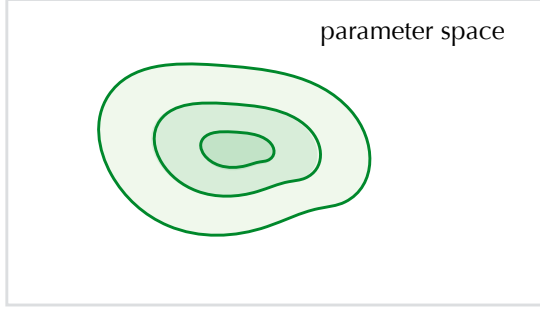


Figure 5.3: With only one residual and three discontinuities in the loss function, the parameter space is divided into four regions. In each part the loss is constant.

5.4 Fast outlier rejection

It would naturally be a great advantage if outliers could be discarded early and not even considered as input to Algorithm 1, and later Algorithm 2. This section will present such a method for fast outlier rejection which works both for registration and stitching. Our presentation is focused on the zero-one, but it can easily be adapted to the truncated L_2-norm as well. Note that only correspondences that can be shown to not be part of an optimal inlier set will be discarded.

The technique iterates through the correspondences, or in our terminology,

Figure 5.4: With two residuals and three discontinuities in the loss function, the parameter space is divided into several regions, but in each of these the loss is still constant.

the residuals. For each residual function $r_i \in S$, we obtain a bound of the following type: *If r_i is an inlier to the (unknown) optimal solution θ^*, then the total loss $l(\theta^*)$ is larger than some number, say l_{bound}.* If this bound l_{bound} is higher than the best solution found so far, we get a contradiction and we can draw the conclusion that r_i must be an outlier. The residual r_i can then be permanently removed from the discussion (that is, removed from S).

Let us assume that residual r_i is below the threshold at optimum, that is, the ith correspondence is an inlier. Under this assumption, we will produce a bound on the optimal solution.

Proposition 5.26. *Suppose that for a set of corresponding points there exists a transformation T represented with parameters θ (for* registration *or* stitching*) such that all residuals are less than ϵ, that is, $r_j(\theta) \leq \epsilon$ for all j. Then there exists another transformation T' with parameters θ' such that $r_i(\theta') = 0$ and all other residuals are less than 2ϵ, that is, $r_j(\theta') \leq 2\epsilon$ for all j but i.*

Proof. (*Registration*) Set $t' = y_i - Rx_i$ and $R' = R$. Then

$$r_i(\theta') = ||R'x_i + t' - y_i|| = 0.$$

Further,

$$||t - t'|| = ||t + Rx_i - y_i|| \leq \epsilon$$

and hence for any j,

$$\begin{aligned} r_j(\theta') &= ||R'x_j + t' - y_j|| \\ &\leq ||Rx_j + t - y_j|| + ||t - t'|| \leq 2\epsilon. \end{aligned}$$

□

Proof. (*Stitching*) Let $\alpha = \angle(y_i, Rx_i)$. Then a rotation R_α about $Rx_i \times y_i$ will map Rx_i exactly to y_i and $||R_\alpha|| \leq \epsilon$. So, set $R' = R_\alpha R$ and hence

$$r_i(\theta') = ||R'x_i - y_i|| = ||R_\alpha Rx_i - y_i|| = 0$$

and for any j,

$$\begin{aligned} ||R'x_j - y_j|| &\leq ||R_\alpha Rx_j - Rx_j|| + ||Rx_j - y_j|| \\ &\leq ||R_\alpha|| + \epsilon \leq 2\epsilon. \end{aligned}$$

□

This means that a bound for the number of 2ϵ-inliers given that $r_i = 0$ is also a bound for the number of ϵ-inliers given that $r_i \leq \epsilon$. For both registration and stitching, computing the former bound is fairly easy as the constraint $r_i = 0$ fixes the transformation up to a one-dimensional rotation using correspondence (x_i, y_i). We can parameterize this rotation with an angle α. Each of the remaining correspondences yields an interval constraint on this α (which should be interpreted modulo 2π) for which it is an inlier. To get a bound on the number of inliers we need to find a point that lies in as many of these intervals as possible. This can be done by sorting the intervals and going through the sorted list. The computationally most costly part here is the sorting and hence the cost of this algorithm is $\mathcal{O}(n \log n)$. If we do this for each correspondence we get a cost of $\mathcal{O}(n^2 \log n)$, which is significantly cheaper than the optimal algorithm.

5.5 Application I: Registration

We return to the image registration problem introduced in Section 4.2. Given two sets of 2-vectors, $\{x_i\}$ and $\{y_i\}$, find a rotation R and a translation t that minimizes the truncated L_2-norm. An example of two image pairs that we would like to align are given in Figure 4.1.

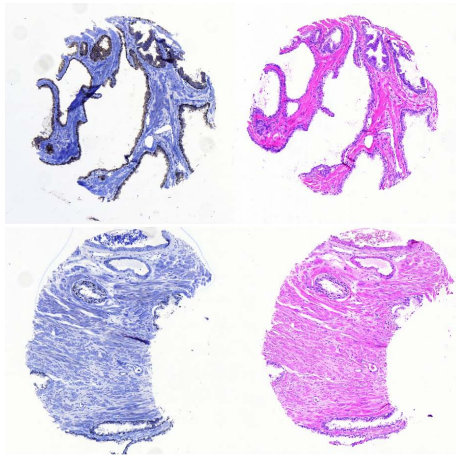


Figure 5.5: Image pairs of prostate tissues in two different stainings.

As per previous discussions we need to find all critical points for subsets B of the residuals. We write the residual constraints g_i as

$$g_i(\theta) = (R(\theta)x_i + t(\theta) - y_i)^T(R(\theta)x_i + t(\theta) - y_i) - \epsilon^2. \qquad (5.9)$$

Clearly these are differentiable. As goal function we choose a linear function, $f(\theta) = \theta_1$.

We will need different polynomial solvers depending on the size of B. With $|B| = 3$ we get three equations of type 5.9 and the embedding $h(\theta) = \theta_1^2 + \theta_2^2 - 1 = 0$. This amounts to four quadratic polynomial equations in four unknowns. Using methods presented in Chapter 3 we have implemented a minimal solver in MATLAB that runs in about 0.6 ms on a standard computer (Intel I5).

For problems of size < 3 we also need to consider the constraint posed on the gradients of g_i, h and f in 4.13. This constraint implies linear dependence of the gradients. These gradients are easy to calculate and using

$$\det\begin{pmatrix} \nabla f & \nabla g_1 & \nabla g_2 & \nabla h \end{pmatrix} = 0\,, \qquad (5.10)$$

we obtain the necessary extra equations.

5.5.1 Experiments

For the registration experiments the problem of matching two differently stained tissue slices of a prostate biopsy is addressed. Examples of such images are shown in Figure 5.5(a) out of a database of 88 pairs. These images vary greatly in the number of reliable matches, and can have high ratios of outliers (recall the example in Figure 4.1).

The outlier rejection method of 5.4 is applied as a preprocessing step to Algorithm 2. Normally 90% to 98% of the outliers are eliminated in this step, but in rare examples this ratio drops and in the two worst examples only 0.2% and 11% are eliminated, respectively. Truncation level is set at $\epsilon = 3$ pixels. Compared with standard RANSAC which optimizes the size of the inlier set and in the end, computes a least-squares fit on this set, there were eight pairs with no difference. The improvement for the other 80 pairs is shown in the upper, left of Figure 5.6. Compared with RANSAC that uses the same truncated quadratic loss, 42 pairs gave no difference. The improvement for the remaining 46 pairs is shown in the upper, right of Figure 5.6. The scaling of the x-axis is normalized by the squared threshold ϵ. This allows one to interpret the result as number of additional outliers. It is clear from the histograms that the optimal method significantly outperforms the best possible obtainable result from a standard inlier optimizing RANSAC approach. Even though the truncated L_2-RANSAC performs considerably better, it is still not optimal in half of the cases.

Runtimes and time complexity. By randomly selecting subproblems of different sizes from the registration data, we examined the running times of the fast outlier rejection and Algorithm 2 as a function of problem size. The results are shown in Figure 5.7. As most of the outliers are removed by the outlier rejection step and the runtime of the second step depends mainly on the number of inliers. This means that the total execution time of our algorithm varies from a couple of seconds to slightly over three hours even if the number of input points is fixed. In practice though, there would be no need to run Algorithm 2 exhaustively in cases with hundreds of inliers.

The practical complexity for Algorithm 2 is cubic rather than the theoretical $\mathcal{O}(n^4)$. It shows that for this problem size, the dominant cost is that of computing the critical points. The runtime of RANSAC is about 1 s for all experiments.

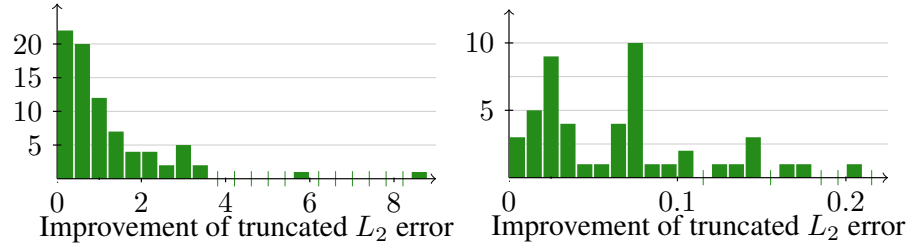


Figure 5.6: Histogram of performance differences between our optimal method and standard RANSAC (*left*) and RANSAC which evaluates the truncated L_2-error for each sample (*right*). The x-axis shows the difference in truncated L_2 loss normalized with a factor $1/\epsilon^2$ such that 1 corresponds to the cost of an extra outlier.

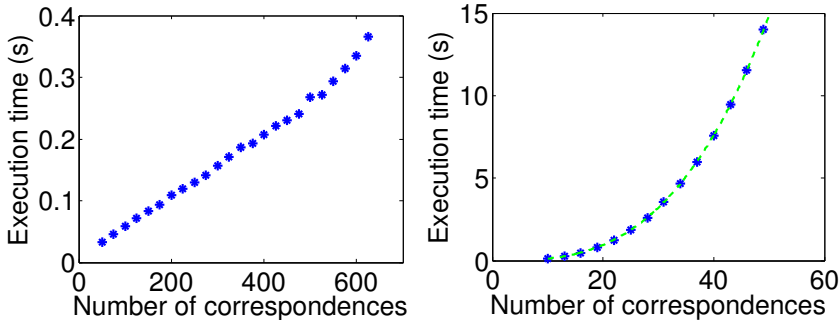


Figure 5.7: Timing experiments for registration. (Left) Runtime as a function of the number of correspondences for the fast rejection method. (Right) Runtime for Algorithm 2 as a function of problem size ($\approx$ the number of inliers). The dashed line is a $y = cx^3$-curve.

5.5.2 Random sampling

Algorithms 1 and 2 involve an exhaustive search over a high number of minimal subsets. For large problems this will make the entire approach infeasible. In such scenarios the RANSAC family of algorithms, testing only a limited number of minimal subsets, are still feasible. What happens if we use our new polynomial

solvers that compute critical points in order to generate hypotheses in the inner-loop of RANSAC?

We illustrate the relative performance with respect to RANSAC through synthetic experiments on registration. In this case we use up to three correspondences to generate hypotheses while the standard solvers need only two. Data is generated by randomly selecting a rotation, translation and point set of 100 points as well as a fixed threshold ϵ. The point set is then transformed and Gaussian distributed random noise with standard deviation ϵ is applied to all points to model potential inliers. Additional random noise with standard deviation 5ϵ was added to 40 of the points to model potential outliers.

After applying the fast rejection algorithm described in section 5.4 we ran RANSAC using both our and standard solvers. As the standard solver produces a single hypothesis for each random subset and our solver generates up to 6 solutions, the standard solver was allowed six times as many iterations. The process is repeated 500 times and the average results are presented in Figure 5.8. Somewhat surprisingly the new solvers outperform the standard ones already after a few hundred iterations. This indicates that it is better to compute critical points with $r_i(\theta) = \epsilon$ as hypotheses in RANSAC rather than the standard error-free hypotheses from $r_i(\theta) = 0$.

The difference is small, but we believe it is an important observation. By just modifying the standard RANSAC procedure to compute critical points instead, one can achieve better results. Moreover, it has a sounder theoretical basis as the optimum solution will eventually be obtained, which is not guaranteed for RANSAC.

5.6 Application II: Triangulation

Given n cameras with known camera matrices we wish to determine the coordinates of a scene point $\theta \in \mathbb{R}^3$. For this case no efficient algorithm for optimizing the standard L_2 loss is known. Hence we restrict ourselves to the zero-one loss, i.e., to minimizing the number of outliers.

The reprojection residual functions are given by

$$r_i(\theta) = \left\| \left(x_i - \frac{a_i^T\theta + \bar{a}_i}{c_i^T\theta + \bar{c}_i}, y_i - \frac{b_i^T\theta + \bar{b}_i}{c_i^T\theta + \bar{c}_i} \right) \right\|, \tag{5.11}$$

provided $c_i^T\theta + \bar{c}_i \geq 0$ where $(a_i^T, \bar{a}_i)$, $(b_i^T, \bar{b}_i)$ and $(c_i^T, \bar{c}_i)$ denote the first,

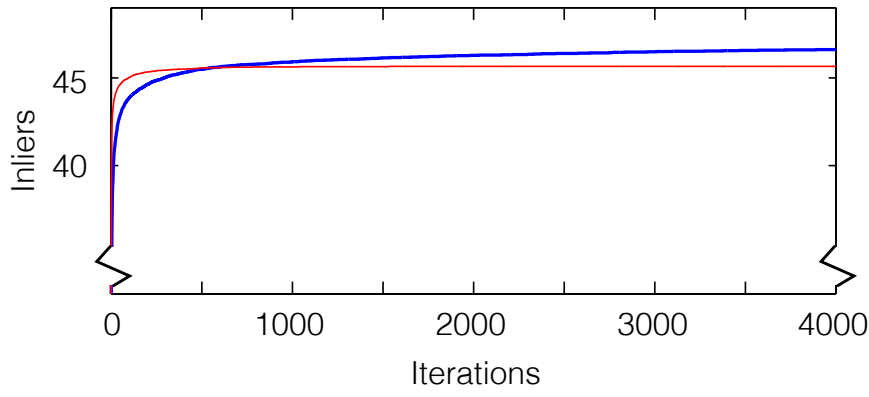


Figure 5.8: A comparison of performance of RANSAC using standard solvers (which assume zero noise) and our solvers based on critical points. The figure shows an average over 500 synthetic experiments. The x-axis shows the number of iterations of our solver. To be fair, the standard solver was allowed six times as many iterations.

second and third rows, respectively, of P_i. Note that all points that satisfy $r_i(\theta) \leq \epsilon$ form a convex cone.

To solve for the critical points we need to enforce constraints of the type $r_i(\theta) \leq \epsilon$ which is a convex cone in $\mathbb{R}^3$. Since the intersection of convex sets is convex, the optimization in DUMMY(I, O) is over a convex set. Any convex set has a unique point with minimal norm, so we choose the auxiliary goal function to be $f = ||\theta||^2$. This ensures that a minimizer exists and in practice ensures that Condition 2 is satisfied.

To find the critical points we need to consider the active constraints $r_i(\theta) = \epsilon_{j_i}$, which can be rewritten as $g_i(\theta) = 0$, with quadratic function

$$\begin{aligned} g_i(\theta) = &\left(\theta^T(x_i c_i - a_i) + x_i \bar{c}_i - \bar{a}_i\right)^2 \\ &+ \left(\theta^T(y_i c_i - b_i) + y_i \bar{c}_i - \bar{b}_i\right)^2 - \epsilon^2 \left(\theta^T c_i + \bar{c}_i\right)^2. \end{aligned} \tag{5.12}$$

We have to consider subsets of up to three residuals. If there are exactly three, then we get a system of three quadratic equations as in 5.12. As predicted by Bezout's theorem there will be up eight solutions. If there are only two active

constraints, then we add an equation by forcing the gradients to be linearly dependent,

$$0 = \det \begin{pmatrix} \nabla g_i & \nabla g_j & \nabla f \end{pmatrix}.$$

which is a cubic equation. This also yields a reasonable system with up to 12 solutions.

Basically the case of just one active constraint can be handled analogously. Note that $\nabla g_i = 0$ at the camera centre for camera i, so the camera centre itself is a critical point. Finally we need to check $\theta = 0$, since then $\nabla f = 0$.

5.6.1 Experimental validation

Algorithm 2 was implemented in MATLAB. The polynomial equations were solved using a generic polynomial solver based on [11]. Experiments were run using the step loss function to minimize the number of outliers on the well-known Dinosaur turn-table sequence with automatically tracked points in 36 images. An image point is tracked in up to 10 consecutive views. We compare our results with a standard minimal solver. We used the optimal two-view triangulation from [32]. To get a fair comparison that solver was tried for all possible choices of two views. Hence the result is the best possible that could be achieved with RANSAC. Figure 5.9 shows a comparison to the proposed method. For most of the triangulated points in the Dinosaur sequence there are no outliers - only cases with at least one outlier are reported.

	Error threshold	
	1 pixel	2 pixels
Same result	108	51
1 outlier less	156	82
2 outliers less	10	2

Figure 5.9: Comparison between the proposed method and the best possible result using a standard solver. For two different thresholds, the number of times that a certain improvement was achieved is presented. For example, with a threshold of 1 pixel, our method had one outlier less in 156 cases.

The running times for the generic solver were around 0.1 s per minimal case on a desktop computer. A specialized solver can solve the same problem in a few milliseconds [11].

5.7 Application III: Stitching

We formulate the problem as Horn *et. al.* [34]. Given two sets of unit 3-vectors, $\{x_i\}$ and $\{y_i\}$, find a rotation R such that

$$\sum_{i=1}^{n} \ell(||Rx_i - y_i||) \tag{5.13}$$

is minimized. As before, $\ell(r) = \min\{r^2, \epsilon^2\}$. Unless the threshold ϵ is large, this is basically equivalent to minimizing squared angular errors.

The relevant inequality constraints for this problem are $g_i(R) = y_i^T R x_i + \epsilon^2/2 - 1 \geq 0$. The parameter space is 3-dimensional, one angle around each axis. We avoid trigonometry in the unknowns and embed the rotation matrix using a unit quaternion q. This does not alter the form of $g_i(R(q)) \geq 0$ but introduces an equality constraint $h(q) = ||q||^2 - 1 = 0$. Here $R(q)$ merely denotes a rotation matrix parameterized by the quaternion q.

For problems defined by three active constraints we again have a system of four equations in four unknowns. Using methods presented in 3 a polynomial solver exploiting the symmetry of the quaternion representation was constructed. For problems where only two inequality constraints are active we can no longer formulate a solvable system. Again we need 4.13 to obtain a fourth equation necessary to solve for the four unknowns in q. Just as for the case of registration we use the implied linear dependence and $\det(\nabla f \ \nabla g_1 \ \nabla g_2 \ \nabla h) = 0$ to obtain this equation. To simplify this expression we select $f(q) = q_1$, giving us $\nabla f = (1, 0, 0, 0)^T$ reducing the above determinant calculation to a subdeterminant calculation. Single active constraints can be handled similarly. Our MATLAB implementation of the polynomial solvers runs in about 10 ms.

5.7.1 Experiments

A stitching example of a pair of images taken by different photographers at different times, downloaded from FLICKR with keywords "view from the Eiffel tower" is given in Figure 5.10(b). In total, a database of 53 image pairs were tested. The images were of size 640×480 and had very narrow overlap. The truncation was set to $\epsilon = 0.001$ corresponding roughly to 1.5 pixels. Compared with standard RANSAC, there were five pairs with no difference. The other 48 pairs are summarized in the lower, left of Figure 5.11. Similarly with the truncated L_2-RANSAC,

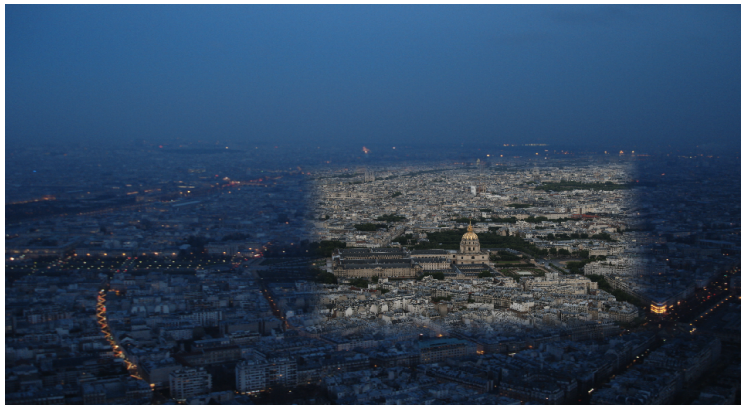


Figure 5.10: Stitching example from FLICKR showing the view from the Eiffel Tower. This challenging example consists of two images taken under very different illumination conditions and resolutions, resulting in poor matching.

there were 38 pairs with no difference while 15 pairs obtained improved results compared with the optimal method, see the lower, right of Figure 5.11.

The relative performance between our optimal method and RANSAC follows the same pattern as for the registration experiment. There is a significant difference compared to standard RANSAC while for the truncated L_2-RANSAC the performance difference is smaller. These findings are consistent with [46] where a truncated quadratic loss is also found to perform better than simply counting inliers in the RANSAC-loop.

5.8 Application IV: Multi-model registration

For the experiments on simultaneous multiple model registration we emulate structures as the one displayed in Figure 5.12(c). The image shows a lung biopsy with three components, matched to adjacent layers of the same tissues. Due to the lack of availability of such data we use our prostate data to generate multi-model cases with close to the same number of inliers. For the cases with two models we used 20 inliers per model, and in total 40 outliers. For the three model case we used models of different dominance and set the inlier sizes to 10, 20 and 30. The number of outliers was set to match the total number of inliers. The results for

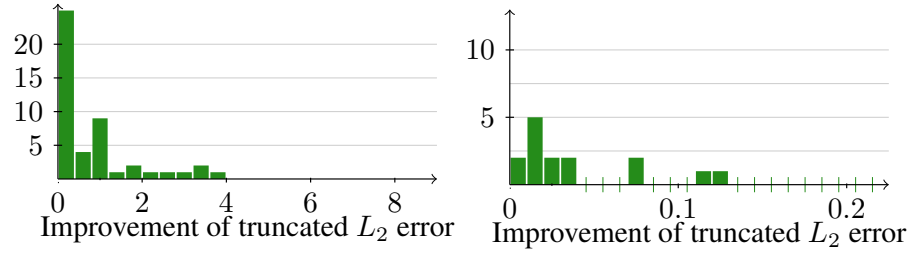


Figure 5.11: Histogram of performance differences between our optimal method and standard RANSAC (*left*) and RANSAC which evaluates the truncated L_2-error for each sample (*right*). The x-axis is scaled such that 1 unit corresponds to the cost of 1 outlier.

both methods, compared to using a sequential RANSAC is displayed in Table 5.1. Average running times were 18 s and 90 s for the two-model examples and the three-model examples, respectively.

For multi-model registration, we use Algorithm 3 with CPLEX for max k-cover, but before running the polynomial solver we use a simple test as discussed in Section 4.4 to determine if there will be any real-valued solutions.

	difference in # inliers										
# models	0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+13
2	2	7	13	9	8	6	2	1	1	1	0
3	1	3	4	8	5	10	5	4	5	4	1

Table 5.1: The improvement of our method compared to sequential RANSAC. Note that RANSAC hardly ever finds the optimal solution even though the sampling of minimal sets is exhaustive.

5.9 Conclusions

We have shown how to perform model estimation under robust loss functions that model both inlier and outlier noise. Our experiments demonstrate that this

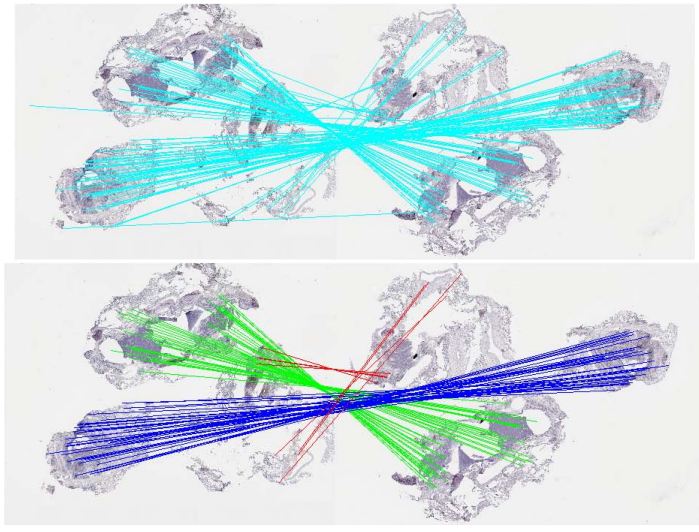


Figure 5.12: (c) Image pair of a lung tissue biopsy which has three components. (Top) Hypothetical correspondences (cyan) obtained from SIFT. (Bottom) Inlier correspondences (green, blue and red) from the three estimated motions.

yields a practical approach when combined with a fast outlier rejection scheme. One weakness is that for large number of inliers, it becomes infeasible to compute the optimal estimate. On the other hand, in such a case, it is not crucial to find all inliers in order to get an accurate estimate. Instead one can use our solvers in a random sampling framework. As experimentally demonstrated in the case of registration, this gives better results compared to applying standard minimal solvers.

The proposed work opens up the possibility to develop new feature detectors. Current detectors such as SIFT are optimized to find a good inlier/outlier ratio, but since we are able to handle large amounts of outliers, one can design new detectors that are optimized to find many inliers.

Chapter 6

Tractability and Fidelity of Optimal Fitting: A Case Study

Image registration is a classical problem in computer vision and it appears as a subroutine for many imaging tasks. For example, it is a prerequisite for shape analysis and modelling [18] and for automated analysis of multi-modal microscopy images [44]. It is also an important component in image guided surgery where often fiducial markers are used for estimating the transformation [30, 21]. In this chapter, we will evaluate the techniques introduced in the previous chapter on a real medical application. Instead of evaluating the improvement using the same goal function that we have theoretical guarantees of optimizing, we will compare to manually obtained ground truth. We will demonstrate that in this setting our methods frequently outperforms standard approaches. In addition we will derive additional optimal solvers, based on observations on the properties of rigid registration. More specifically cost functions based on the truncated and non-truncated L_1-norms are utilized.

The particular type of problem we are interested in is estimating rigid image transformations under less controlled situations where there may be a substantial number of mismatches and where it is important to obtain reliable results. For example, the method should not be dependent on a good initialization.

Naturally, the registration problem has been studied in depth. When choosing the method of preference, one is often faced with the following dilemma. Using a simplified, mathematical model of the problem enables efficient computations, but sacrifices realism. While using a more realistic model incurs the computational cost of hard inference. As an example, consider the case of feature-based registration under the assumption that measurement noise in the target image can be modeled by independently distributed Gaussian noise. This is in fact the standard Procrustes problem which can be solved in closed form. However, the model

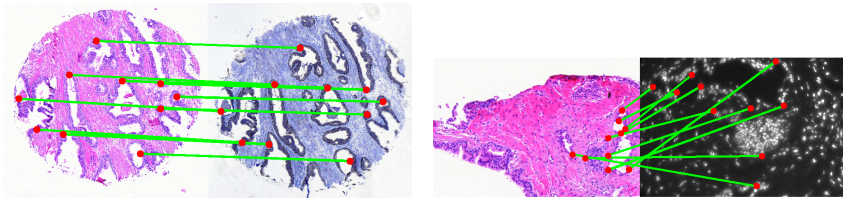


Figure 6.1: Examples from our two benchmarks with 10 manually marked correspondences. *Left:* Prostate tissue stained with H&E and p63/AMACR. *Right:* Prostate tissue stained H&E and TRF (fluorescent). The goal is to find a rigid transformation that aligns the two images using features from an automated method such as SIFT.

is not so realistic as there are typically erroneous measurements - *outliers* - among the feature correspondences. This makes the registration estimates very unreliable. On the other hand, modelling outliers leads to a much more complicated optimization problem and solving this problem exactly is sometimes dismissed as infeasible. Heuristic methods based on random sampling and expectation maximization dominate the field. We show that one can achieve a method which is both efficient (in terms of speed) and reliable (with respect to outliers).

In our first setting, the objective is to perform histologic analysis of biopsies. Prostate cancer is the second most common cancer in men worldwide [36] and whose gold standard of diagnosis and prognosis is based on histologic assessment of tumours in images stained with Hematoxylin and Eosin (H&E). Several automatic pattern recognition prototypes exist [51, 22]. In order to improve the accuracy in clinical practice, considerable research efforts have been directed to complement the analysis with additional types of stainings and imaging modalities [44]. One example is given in the left of Fig. 6.1 where two adjacent tissue sections have been stained with H&E and antibodies directed against p63/AMACR, respectively. Another example is given in the right of Fig. 6.1 with one H&E staining and one Time Resolved Fluorescence (TRF) image measuring the Androgen Receptor (AR) obtained from the same section. This type of images is quite challenging for any automated approach because reliable feature correspondences are hard to obtain and there are image degradations due to imperfect acquisition.

In our second setting, we are dealing with images of the human brain and the goal is to study the perfusion of blood flow through small vessels, so-called

capillaries in the white and gray matter regions of the brain. This is important for patients with hydrocephalus which are treated by placing a drainage tube (shunt) between the brain ventricles and the abdominal cavity to eliminate the high intracranial pressure, see `wikipedia/hydrocephalus` [67]. To capture the anatomy of the region of interest, MR-Flair images have been obtained. The perfusion data is obtained via contrast-enhanced CT images taken at one second apart during a two-minute session. To acquire good temporal resolution, only a couple of slices can be captured at each time instant. The challenge here is to register single slices from the CT image to the full 3D volume of the MR image. As the head of the patient is in an upright position, the mapping from one CT slice to the corresponding (but unknown) slice in the MR-Flair volume is well described by a rigid 2D transformation after having adjusted for known scale differences.

The methods are compared and extensively evaluated on The focus of our evaluation is on two important desiderata that a satisfactory solution should possess, namely *tractability* and *reliability*. The first term refers to the computational complexity. We investigate both the performance in practice and derive theoretical complexity bounds as a function of the number of feature correspondences. The second one concerns the reliability of the estimate. We are interested in methods that produce provably optimal estimates under a robust loss function. If the registration fails, then it can either due to lack of good correspondences or the algorithm's inability to find a good solution. In our approach, the latter source of error is removed from the process.

Note that the set of algorithms we propose is restricted to rigid point set registration in the plane, and other settings are not considered.

6.1 Choice of Loss Function

It is a common and reasonable assumption that there exist *correct* but noisy point correspondences as well as complete mismatches or outliers. The errors in the positioning of correct correspondences follow approximately a normal distribution, whereas the outliers are uniformly spread over the image. In [7] it is shown that in order to find a maximum likelihood estimate, a sum of loss functions of the following type

$$\ell(r) = -\log\left(c_1 + \exp\left(-r^2/c_2\right)\right) \tag{6.1}$$

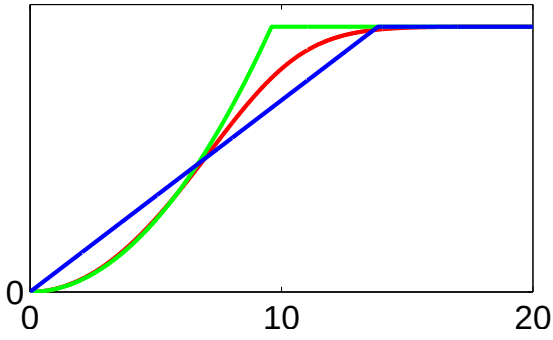


Figure 6.2: The robust loss function (red) suggested in [7], the truncated L_2-error (green) and the truncated L_1-error (blue) that can be optimized using the proposed framework.

should be minimized, where r is the residual error for one correspondence and the constants depend on the amount of inlier noise as well as on the rate of outliers; see Fig. 6.2. In the previous chapter an approximation wass obtained by truncating the squared error. However, the quality of this approximation depends heavily on the rate of outliers in data. At higher rates the loss function levels out much more slowly. In this case a truncated L_1-loss can be a more appropriate choice. All these loss functions lead to a non-convex problem with many local minima. One may even wrongly conclude that the problem is intractable to solve optimally, that is, that no polynomial-time algorithm exists. The possibility to solve for truncated L_2-norm has already been proven, and we will now look at the the truncated L_1 -loss.

6.2 Fast Optimization of the Truncated L_1-Norm

Given corresponding point coordinates in two images, $\mathbf{x}_i = (x_i, y_i)^T$ and $\mathbf{x}'_i = (x'_i, y'_i)^T$, $i = 1, \ldots, n$, consider the following problem

$$\min_{R,\mathbf{t}} \sum_{i=1}^{n} \ell(\|R\mathbf{x}_i + \mathbf{t} - \mathbf{x}'_i\|_1)\,, \tag{6.2}$$

where R is a 2×2 rotation matrix and $\mathbf{t}$ a translation vector, parameterized as

$$R(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \text{ and } \mathbf{t} = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}, \tag{6.3}$$

respectively and where ℓ is the loss function $\ell(r) = \min\{r, \epsilon\}$ for some given threshold ϵ, that is, the truncated L_1-norm.

The following observation allows us to simplify the problem.

Lemma 6.27. *For any fixed rotation R, consider the minimization of (6.2) over* $\mathbf{t}$

$$\min_{\mathbf{t}} \sum_{i=1}^{n} l\left(|x_i \cos\alpha - y_i \sin\alpha + t_1 - x_i'| + |x_i \sin\alpha + y_i \cos\alpha + t_2 - y_i'|\right). \tag{6.4}$$

Then there exist two indices j and k in $\{1, \ldots, n\}$ such that

$$\begin{aligned} t_1^* &= x_j' - x_j \cos\alpha + y_j \sin\alpha \\ t_2^* &= y_k' - x_k \sin\alpha - y_k \cos\alpha \end{aligned} \tag{6.5}$$

is an optimal choice of $\mathbf{t}$.

In order to get a geometric intuition why the above lemma is true, consider the graph of the loss function in (6.4). Note that it is piecewise linear in $\mathbf{t}$ and a global minimum can be found by examining all break points, that is, points which are non-differentiable in all directions. There are two different causes for non-differentiability in our objective function. One is due to truncation and one is due to taking absolute values. Our proof shows that break points that are also local minima are given by (6.5). This means that break points caused by truncation need not be examined since all local minima are due to taking absolute values.

Proof. The optimal $\mathbf{t}^*$ to the truncated L_1-loss, denoted $L(\mathbf{t}^*)$, is also a global minimizer to the L_1-loss on the set of optimal inlier correspondences (those that have residuals less than ϵ). To see this, let $L_{inliers}(\mathbf{t}^*)$ be the optimal loss on the inliers and $L_{outliers}(\mathbf{t}^*)$ the loss on the outliers. Assume that there exists a different solution $\mathbf{t}$ with

$$L_{inliers}(\mathbf{t}) < L_{inliers}(\mathbf{t}^*). \tag{6.6}$$

Clearly,

$$L_{outliers}(\mathbf{t}) \leq L_{outliers}(\mathbf{t}^*)\,, \tag{6.7}$$

as this is already maximal. Hence

$$L(\mathbf{t}) = L_{inliers}(\mathbf{t}) + L_{outliers}(\mathbf{t}) < L_{inliers}(\mathbf{t}^*) + L_{outliers}(\mathbf{t}^*) = L(\mathbf{t}^*) \tag{6.8}$$

which is a contradiction. This shows that an optimal $\mathbf{t}^*$ is a local optimum to the L_1-loss. The formula for the L_1-loss is given by

$$\sum_{i=1}^{n} |x_i \cos\alpha - y_i \sin\alpha + t_1 - x_i'| + |x_i \sin\alpha + y_i \cos\alpha + t_2 - y_i'|. \tag{6.9}$$

As no absolute value contains both t_1 and t_2 we can write this as a function of t_1 plus a function of t_2 and the minimization with respect to t_1 and t_2 can be analyzed separately. Consider the t_1-part. We have a piecewise linear function that tends to infinity as $|t_1|$ tends to infinity and thus a minimizer of this function is at a break point. The break points are due to the absolute values - there is a break point whenever one of the absolute values is exactly zero. Hence a minimizer exists for which at least one absolute value is zero, so $t_1^* = x_j' - x_j \cos\alpha + y_j \sin\alpha$ for some j as stated in the lemma. The same argument for t_2 proves the lemma. □

The lemma shows that if the two indices j and k are given (for example, by exhaustively trying all possibilities), we can reduce the problem via substitution of $\mathbf{t}^*$ in (6.5) to a one-dimensional search over rotation angle α,

$$\min_{\alpha} \sum_{i=1}^{n} \ell(|\delta x_{ij} \cos\alpha - \delta y_{ij} \sin\alpha - \delta x_{ij}'| + |\delta x_{ik} \sin\alpha + \delta y_{ik} \cos\alpha - \delta y_{ik}'|), \tag{6.10}$$

where $\delta x_{ij} = x_i - x_j$, $\delta y_{ij} = y_i - y_j$, etc. Let us denote the resulting, piecewise smooth objective function in (6.10) by $L(\alpha)$, see Fig. 6.3 for an illustration. It has optimum either at a break point or at a stationary point. The break points are places where the derivative $L'(\alpha)$ is discontinuous and occur when an absolute value is exactly zero or the number in an input to ℓ is exactly ϵ. Hence the number

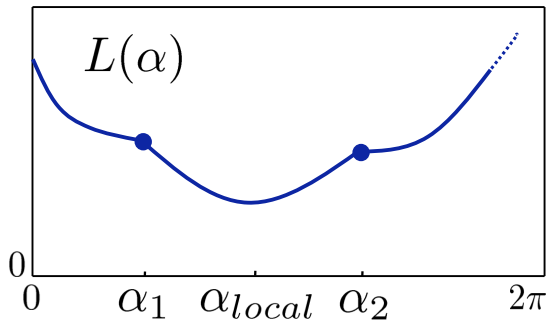


Figure 6.3: Sketch of the objective function in (6.10), denoted $L(\alpha)$, which is piecewise smooth.

of break points grows linearly with n. Given the break points $\alpha_1, \alpha_2, \ldots, \alpha_M$, consider an interval $[\alpha_i, \alpha_{i+1}]$ of $L(\alpha)$. It can be described by

$$L(\alpha) = w_1 \cos\alpha + w_2 \sin\alpha + w_3, \tag{6.11}$$

for some constants w_1, w_2 and w_3. By examining all intervals, we can compute the optimal rotation angle α^* using Algorithm 4.

Algorithm 4 Finding the rotation angle

Set $L^* := \infty$.
Compute all break points of $L(\alpha)$ for $\alpha \in [0, 2\pi)$.
Sort the break points α_1, α_2, ..., α_M.
for $i = 1, \ldots, M$
    Compute $L(\alpha_i)$ and compare with L^*.
    Compute w_1, w_2 and w_3 of (6.11) for $[\alpha_i, \alpha_{i+1}]$.
    Compute local minimum α_{local} of (6.11).
    if $\alpha_{local} \in [\alpha_i, \alpha_{i+1}]$,
        compute $L(\alpha_{local})$
        compare with L^*.

6.2.1 Complexity

There are two important things to note here. First, that each time we compute w_1, w_2 and w_3 for $[\alpha_i, \alpha_{i+1}]$ in (6.11), we can take advantage of the constants from the previous interval $[\alpha_{i-1}, \alpha_i]$. Only the coordinates $\mathbf{x}_i$ and $\mathbf{x}'_i$ that gave rise to α_i are required for computing the update. Second, that there is only one local minimum to

$$w_1 \cos\alpha + w_2 \sin\alpha + w_3,$$

being

$$(\cos\alpha, \sin\alpha) = \pm(w_1, w_2)/\sqrt{w_1^2 + w_2^2}, \tag{6.12}$$

given by the minus sign. Hence each step in the for-loop of Algorithm 4 is $\mathcal{O}(1)$ so the computationally heaviest step is the sorting. Given the indices j and k, we can find an optimal α^* in $\mathcal{O}(n \log n)$. If we consider all possible index pairs j and k exhaustively, the total complexity is $\mathcal{O}(n^3 \log n)$. Note that the most complex arithmetic operations in the algorithm consists of computing square roots.

6.2.2 Fast Outlier Rejection

To increase the speed even more we propose a fast outlier rejection step as preprocessing, analoguous to the approach from Chapter 5. For this we need a variant of Algorithm 4 that works with the zero-one loss (denoted by L_0), that is, counting the number of outliers rather than truncated L_1-norm. First note that the zero-one loss has the same break points as truncated L_1 and that the loss function only changes values at these break points. There, it either increases with one or decreases with one. Algorithm 5 lists the details.

We will use this algorithm together with the following observation.

- Assume that for the optimal transformation $(R^*, \mathbf{t}^*)$, correspondence k is an inlier and there are N outliers, i.e. residuals larger than ϵ. If we change the translation to $\mathbf{t}$ so that $r_k(R^*, \mathbf{t}) = 0$, then, since $\|\mathbf{t} - \mathbf{t}^*\| \leq \epsilon$, the error on inliers has increased with at most ϵ so there are at most N residuals larger than 2ϵ.

This means that we can use Algorithm 5 with threshold 2ϵ to produce a bound of the following kind: *If correspondence k is an inlier, then there are at least N outliers.* This also yields a bound on the truncated L_1 loss, as if N residuals are

Algorithm 5 Upper bound on inliers

Initialize best loss, $L_0^* = \infty$.
Compute all break points of $L_0(\alpha)$ for $\alpha \in [0, 2\pi)$.
Sort the break points $\alpha_1, \alpha_2, \ldots, \alpha_M$.
Compute $L_0(\alpha_1)$ and update L_0^*.
for $i = 2, \ldots, M$
    Depending on the type of α_i
    Set $L(\alpha_i) = L(\alpha_{i-1}) \pm 1$ and update L_0^*.

$> \epsilon$, then the truncated L_1 loss is at least $N\epsilon$. If this is a higher loss than one we have already found, we can discard correspondence k from further consideration.

Algorithm 6 Fast Outlier Rejection

Given an upper bound L_c on the optimal loss.
for $i = 1, \ldots, n$
    Set $\mathbf{t} = \mathbf{x}_i' - \mathbf{x}_i$
    Use Algorithm 5 with threshold 2ϵ to compute L_0^*
    (The output L_0^* is a bound on the number of outliers)
    if $L_0^*\epsilon > L_c$,
        discard correspondence i

A value for L_c can be found by running Algorithm 6 using ϵ in place of 2ϵ and simply storing the best loss function value rather than discarding points. As the dominating cost inside the loop is the sorting in Algorithm 5 running this scheme to remove outliers costs only $\mathcal{O}(n^2 \log n)$ and can be used as a preprocessing step while keeping guaranteed optimality.

6.3 Fast Optimization of the L_1-Norm

Optimizing the L_1-norm is a simpler problem compared to the truncated case. In fact, one can set $\epsilon := \infty$ and use the same algorithm, but we can do better. Lemma 6.27 still applies, so we can eliminate the translation and only consider

the rotation problem, which simplifies to

$$\min_{\alpha} \sum_{i=1}^{n} |\delta x_{ij} \cos\alpha - \delta y_{ij} \sin\alpha - \delta x'_{ij}| + |\delta x_{ik} \sin\alpha + \delta y_{ik} \cos\alpha - \delta y'_{ik}|. \tag{6.13}$$

An important difference here is that we can compute the break points for the first term and the second term *independently*. This means that we can precompute and sort all the break points for $j, k = 1, \ldots, n$ in $\mathcal{O}(n^2 \log(n))$ and then use the for-loop of Algorithm 1 to find the optimal α^*. Now, the heaviest part is no longer the sorting. The total time complexity is $\mathcal{O}(n^3)$ since the for-loop is $\mathcal{O}(n)$ and exhaustively trying all combinations of j and k is $\mathcal{O}(n^2)$.

6.4 Experiments

The proposed methods have been evaluated on two challenging registration tasks.

6.4.1 Registering Histology Sections

The first set of experiments is concerned with the registration of histology sections of prostate tissue, and also serves as a quantitative evaluation. We used one dataset with 88 image pairs of adjacent slides of prostate tissue, stained using H&E and p63/AMACR, respectively. Another dataset consists of 103 images of H&E stained slides, in which sub-parts are also analyzed with TRF. Examples can be seen in Fig. 6.1. The size of the stained images are on the order of 1100x1100, while the TRF images are 368x546.

We used SIFT features as the basis of our point-to-point correspondences. Matching was restricted to the same scale octave and we used Lowe's ratio criterion with a threshold at 0.9 to discard poor matches. This yielded 800-1500 matches for the first dataset, and, due to TRF images being smaller, 40-500 matches in the second dataset. The inlier rate varies from 1% to 40% with a 10% average for the H&E-p63/AMACR set and from 4% to 54% for the H&E-TRF set with a 28% average.

The proposed algorithms from this and the previous chapter were compared to standard L_2-minimization and RANSAC followed by L_2-minimization on the inlier set. For each problem instance, 10 correspondences were manually picked by an expert and used to compute an optimal transformation under the L_2-loss.

Reported results are compared to the rotation and the translation of this estimate. We have also selected two failure criteria based on these comparisons. The first being that the rotation error is larger than 5°, the second that the translation error is larger than 25 pixels. The percentage of results that fail according to these criteria are presented.

	RANSAC						Truncated norms					
ϵ	500 iter.			1000 iter.			L_1			L_2		
1p	8.38°	221p	27%	8.14°	105p	28%	2.72°	61p	11%	2.47°	58p	8%
5p	2.85°	53p	9%	1.94°	28p	5%	1.21°	7p	3%	0.43°	6.4p	2%
10p	1.23°	42p	6%	2.24°	43p	6%	0.29°	4.8p	1%	0.28°	4.6p	1%
20p	2.43°	34p	8%	0.91°	23p	3%	0.27°	4.0p	0%	0.26°	3.9p	0%
∞		-			-		2.43°	6.5p	5%	6.54°	94p	69%

Table 6.1: The results for the H&E - p63/AMACR benchmark. In the left column, the inlier threshold ϵ is varied. Then, for each of the methods (RANSAC with varying number of iterations, and the truncated L_1- and L_2-norms), three numbers are reported: average rotation error (degrees), average translation error (pixels) and failure rate. A failure case is one with error in rotation larger than 5° or in translation larger than 25 pixels. When $\epsilon = \infty$, no truncation takes place.

The experimental results on H&E-p63/AMACR are shown in Table 6.1. The most accurate results are obtained by the truncated L_2-method. Truncated L_1-norm performs poorly on the lowest threshold, but at more reasonable levels for this task performance is similar to truncated L_2. None of the methods based on RANSAC succeeds on all examples, although the accuracy is good at higher thresholds with 1000 iterations. We also note that regular L_1-norm (marked ∞) succeeds much more frequently than L_2-norm and with better accuracy than a majority of the RANSAC variants—on a dataset with only 10% inliers on average. For the highest threshold level (20p), we have also performed a test exhaustively trying all possible hypotheses ransac could generate. The accuracy is slightly worse but comparable to the L_1- and L_2- truncated methods. However, as the time complexity is $\mathcal{O}(n^3)$, close to the complexity of truncated L_1 with more expensive operations and no fast rejection method, it is in practice as slow or slower as the L_1- method while having no theoretical guarantees.

Results from the benchmark experiment on H&E-TRF registration are shown

	RANSAC						Truncated norms					
ϵ	100 iter.			500 iter.			L_1			L_2		
1p	2.53°	30.6p	6%	0.40°	6.2p	1%	0.34°	2.9p	0%	0.31°	2.6p	0%
2p	2.27°	31.2p	5%	0.29°	2.6p	0%	0.29°	2.6p	0%	0.28°	2.6p	0%
3p	1.75°	23.4p	4%	0.29°	2.7p	0%	0.28°	2.6p	0%	0.28°	2.6p	0%
4p	0.66°	8.5p	3%	0.28°	2.6p	0%	0.28°	2.6p	0%	0.27°	2.6p	0%
5p	1.14°	7.4p	2%	0.26°	2.5p	0%	0.27°	2.5p	0%	0.26°	2.6p	0%
10p	0.76°	7.8p	1%	0.27°	2.4p	0%	0.26°	2.5p	0%	0.26°	2.4p	0%
∞	-			-			15.6°	173p	57%	33.6°	341p	100%

Table 6.2: The results for the H&E - TRF benchmark. See Table 6.1 for explanation.

in Table 6.2. This dataset has significantly fewer matches per image pair and higher inlier ratios, making it more suitable for RANSAC. With 1000 iterations, RANSAC performs on par with truncated L_1-norm and truncated L_2-norm, but with fewer iterations there are still some failures. The poor results for regular L_1-norm and L_2-norm show that for this task, aligning a sub-image to a larger image, using truncated norms is essential.

We also tested the intensity-based Image Regsitration Toolkit [61], using normalized mutual information. For the first dataset, the toolkit failed to produce a correct registration (less than 5° rotation and 25 pixels translation error) in 86% of the experiments. For the second dataset it failed to produce any corrrect results. The poor results are not surprising as these methods often are sensitive to initialization and to outlier structures in the images.

6.4.2 Registering CT to MR-Flair

This experiment is a demonstration of the applicability of the method. For more quantitative results, see Sections 6.4.1 and 6.4.3. The dataset consists of 44 image slices captured using the MR-Flair methodology and 4 image slices from a CT-scan of one single subject. To correlate the information provided by the different modalities, one would like to register each of the CT slices to the MR-Flair volume. As the CT slices are roughly aligned with the slices of the MR-Flair volume, we try to register each of the CT slices to each of the MR-Flair slices and then try to find the sequence of four MR-Flair slices that best match the four CT slices.

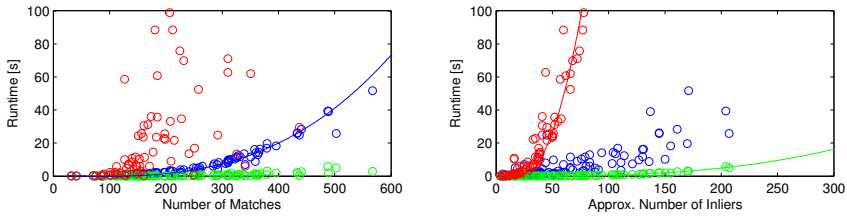


Figure 6.4: *Left:* Runtime as a function of number of matches is graphed for truncated L_1-norm (green), truncated L_2-norm (red) and regular L_1-norm. The L_1-method follows closely a $\mathcal{O}(n^3)$-curve (blue). *Right:* Runtime as a function of number of inlier matches is graphed. The truncated methods are more correlated to the number of inliers, see the $\mathcal{O}(n^4)$-curve (red) and the $\mathcal{O}(n^3 \log(n))$-curve (green), respectively. ($\epsilon = 10$ pixels.)

We use standard 2D SIFT to obtain correspondences. To improve the matching performance, all descriptors were extracted at a fixed scale instead of using the estimated scale from the difference of Gaussians detector. The motivation is that in very noisy images the scale estimation tends to be uncertain.

Due to the small dataset we only present qualitative results for our truncated L_1-approach. Different slice-matches have different number of potential inliers, making the truncated L_1-cost skewed. However this is easily rectified by using a modified cost c_i defined as $c_i = N_i\epsilon - l_i^*$ where N_i is the combined number of correspondences for subsequence i, $epsilon$ the truncation level, and l_i^* the combined optimal truncated L_1-solution. Using this criterion and $\epsilon = 10$ the best subsequence evaluated at $c_{19} = 397$, with closest runners up $c_{18} = 362$, $c_{17} = 367$. All other sequence-matchings had significantly lower score. We show the found matchings for the best matching in Fig. 6.5.

The frequently used intensity-based method called NIFTYREG [55] using mutual information was also tested, but without any reasonable registration results at all. Note that this method was also developed to cope with outlier structures by using robust estimation techniques.

6.4.3 Speed

The theoretical worst time complexities are stated in Table 6.3. In practice RANSAC is not run exhaustively but with a fixed number of k iterations, giving a complex-

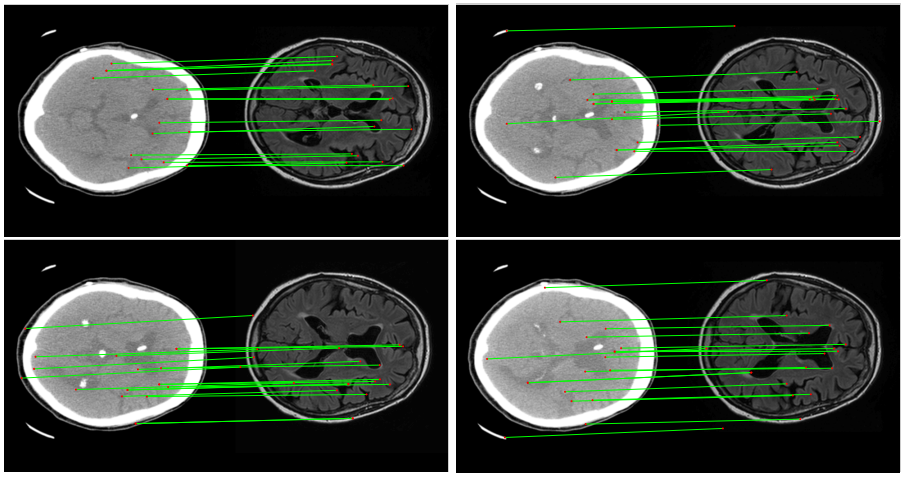


Figure 6.5: The found inliers for the best subsequence obtained using our truncated L_1 algorithm.

ity of $\mathcal{O}(nk)$. For average-size problems (280 matches) and $k = 1000$, RANSAC required 73 ms. The fastest (but worst-performing) method is the closed-form L_2-method with a typical runtime of 0.2 ms. For the remaining methods timing plots are shown in Fig. 6.4. Because of the fast outlier rejection scheme discussed in Section 6.2.2, runtimes of the truncated L_1-norm and L_2-norm depend mainly on the size of the inlier sets. The full L_1-method has no such advantage. These numbers clearly show the advantage in runtime for the truncated L_1-method over both the regular L_1-norm and the truncated L_2-norm. However, on datasets consisting of a majority of inliers, the lower complexity of the L_1-norm would give faster runtimes as all operations are identical apart from the sorting strategies. The timing statistics is from experiments on H&E-TRF, though the same analysis holds for H&E-p63/AMACR.

6.5 Discussion

So what is the right way to attack feature-based image registration in presence of outliers? The literature provides us with a vast amount of choices, but many of these are based on local optimization and require a reasonable starting solution,

Algorithm	*complexity*	*tractability*	*reliability*	*reference*
RANSAC	$\mathcal{O}(n^3)$	high	medium	[27]
Truncated L_1-norm	$\mathcal{O}(n^3 \log(n))$	high	high	Our
L_1-norm	$\mathcal{O}(n^3)$	high	medium	Our
Truncated L_2-norm	$\mathcal{O}(n^4)$	medium	high	Our
L_2-norm	$\mathcal{O}(n)$	high	low	[35]

Table 6.3: Characteristics of the algorithms presented or discussed in this and previous chapters. Note that the stated complexity for RANSAC is for exhaustive selection of all minimal subsets which can be thought of as a worst time complexity bound.

which means that the outlier problem is already more-or-less solved. To handle really difficult outlier problems, RANSAC-type algorithms are the standard against which others are measured. However, as our experiments show, they are sub optimal both in terms of accuracy and with respect to the risk of failure. Some of the failures could be avoided by increasing the number of iterations - even up to exhaustively searching all the minimal subsets. But, that will increase the complexity to $\mathcal{O}(n^3)$, being practically the same as the algorithms proposed here (Table 6.3). More importantly, even then there is no guarantee as to the solution quality (Fig. 6.6). Hence, we would only recommend RANSAC when the amount of outliers is known to be low and the available runtime is very limited.

This contrasts sharply to the typical setting for medical image registration

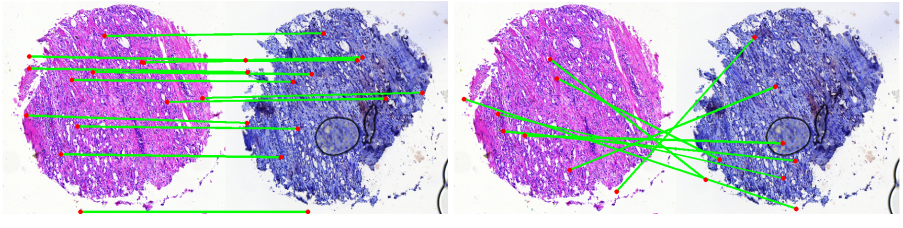


Figure 6.6: *Left:* 13 inliers among 1179 hypothetical SIFT matches of the truncated L_1-method (*success*). *Right:* 8 inliers of RANSAC with 1000 iterations (*failure*). This was the hardest case to register among all pairs. ($\epsilon = 20$ pixels.)

where the process is performed offline. With different image modalities, the rates of outliers are usually high. In these cases the increased reliability of optimizing a truncated norm is valuable and the L_1-based methods, although slower than RANSAC, should be efficient enough for most applications. Our experiments indicate only a small gain in accuracy for the truncated L_2-norm, so using the truncated L_1-norm would be the general recommendation.

In many applications, the actual improvement in terms of accuracy and failure rates of these methods might not be huge. This is compensated by the value of removing a possible error source and not having to tune the parameters of the algorithm. We believe that the choice between a tractable, reliable algorithm with guaranteed high-quality solutions and a fast algorithm with no guarantees whatsoever should be an easy one.

Part III

Sensor Network Calibration

Chapter 7

Difference in Dimension TDOA Measurement Calibration

Sound ranging or sound localization are used to determine the sound source using a number of microphones at known locations and measuring the time difference of arrival of sounds. Such techniques are used today with microphone arrays to enable beamforming and speaker tracking. Calibration of a sensor network using only TOA or TDOA measurements is a nonlinear optimization problem, for which proper initialization is essential. Several previous works rely on prior knowledge or extra assumptions of locations of the sensors to initialize the problem. In [6], the distances between pairs of microphones are manually measured and multi-dimensional scaling is used to compute microphone positions. Other options include using GPS [52] to get approximate locations, or using transmitter-receiver pairs (radio or audio) that are close to each other [23, 57, 20]. In [16] it is shown how to estimate additional microphones, once an initial estimate of the positions of some microphones are known. Another line of work focus on solving the initialization without any additional assumptions. Initialization of TOA networks has been studied in [59], where solutions to the minimal case of 3 transmitters and 3 receivers in the plane is given and in [43], where solutions to the minimal cases of $(4, 6)$, $(5, 5)$ and $(6, 4)$ receiver-transmitter combinations are presented. Initialization of TDOA networks is studied in [56], where solutions were given to non-minimal cases in 3D (10 receivers, 5 transmitters) for TDOA and in [9] where four cases of $(9, 5)$, $(7, 6)$ and $(6, 8)$ receiver-transmitter combinations are presented. However solvers for the minimal cases $(10, 5)$, $(7, 5)$, $(6, 6)$ and $(5, 9)$ are still open research problems. A related work that is based on iterative solvers and similar rank constraints as we use is [31].

In this chapter we study the initialization network calibration problem from only TDOA measurements for the case where there is a difference in dimension

between the spaces spanned by the receivers and by the transmitters. We combine the techniques developed in [43] and [9]. This makes it possible to solve for many (almost all) of the relevant minimal cases. Solving these cases is of theoretical importance. The solvers can also be used in RANSAC [29] schemes to remove outliers in noisy data. The methods are validated both on synthetic and real data. The node localization is cross-validated against computer vision based approaches.

7.1 Problem Formulation

Under the assumption that signals travel at constant speed measuring time of arrival (TOA) is equivalent to measuring distance. TOA requires synchronization between transmitters and receivers in the sense that both transmitting time and time of arrival is available for analysis. This is often not the case and only differences in time or distance is measurable, with either only synchronized transmitters or receivers. In the following discussions we will always assume that the *receivers* are synchronized.

Given a set $\{\mathbf{r}_i\}$ of receivers and a set of $\{\mathbf{s}_j\}$ of transmitters a TDOA measurement is

$$f_{ij} = ||\mathbf{r}_i - \mathbf{s}_j||_2 + o_j \ , \tag{7.1}$$

where o_j is an unknown offset, compensating for the lack of synchronization between transmitters and receivers.

If the size of the set $\{\mathbf{r}_i\}$ is k and the size of $\{\mathbf{s}_j\}$ is n, we have kn measurements $\{f_{ij}\}$. Assuming all positions are unknown, the basic TDOA problem is

Problem 7.1.1 Given all pairwise measurements $\{f_{ij}\}$ find all positions $\mathbf{r}_i \in \mathbb{R}^{\mathcal{D}_r}$ and all positions $\mathbf{s}_j \in \mathbb{R}^{\mathcal{D}_s}$.

Note that solving Problem 7.1.1 implicitly includes solving the unknown offsets o_j.

The topic of this chapter is to determine for what choices of k and n Problem 7.1.1 is solvable when transmitters and receivers live in spaces with different dimension and provide solvers for these cases. To do this we first need some understanding of how the difference in dimension limits the uniqueness of the solution. We look at an illustrative configuration of 3 receivers, $\mathbf{r}_i$, on a line and 1 transmitter placed arbitrarily in the plane; see Figure 7.1. We choose coordinate

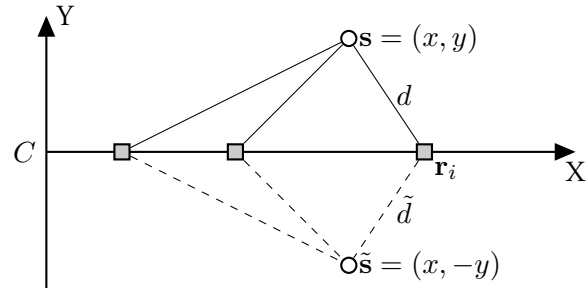


Figure 7.1: Ambiguity in the line-plane case. The point $\mathbf{s}$ and $\tilde{\mathbf{s}}$ are indistinguishable in regards to distances to $\mathbf{r}_i$.

system such that the receivers lie on the x-axis. It is clear that $\mathbf{s} = (x, y)$ results in the exact same measurements as $\mathbf{s} = (x, -y)$. As long as additional $\mathbf{r}$ sensors remain on the line all transmitters $\mathbf{s}_j$ will be individually ambiguous in this respect. We will see later that this is representative of all difference in dimension cases. In practive, we will handle this ambiguity by only solving up to a suitable halfspace, i.e. among the possible solutions we select one where e.g. all ambiguous coordinates are positive.

With these insights regarding solvability, we can define the subproblems

Problem 7.1.2 $\mathcal{D}_s - \mathcal{D}_r = 1$, and structure as in Problem 7.1.1.

Problem 7.1.3 $\mathcal{D}_r - \mathcal{D}_s = 1$, and structure as in Problem 7.1.1.

Here $\mathcal{D}_r$ is the dimension of measurements $\mathbf{r}$ and $\mathcal{D}_s$ the dimension of $\mathbf{s}$.

Since all obtained measurement in both the TOA and TDOA setting depend only on relative distances between points, subjecting all points in any given constellation to a common isometric transformation will not affect the measurements. This observation has two important implications, summarized in the following lemma.

Lemma 7.28. *Problems of type 7.1.1 with $\mathcal{D}_s - \mathcal{D}_r > 1$ or $\mathcal{D}_r - \mathcal{D}_s > 1$ can be reduced to Problem 7.1.2 and Problem 7.1.3 respectively.*

Proof. Since one set of points span a lower-dimensional space the transformation that allows us to express these points as $(\mathbf{x}^T, \mathbf{0}^T)$ exists. Assuming the higher

dimension is m and the lower is k de distance d_{ij} between points $\mathbf{x}_i$ and $\mathbf{y}_i$ fulfill

$$||(\mathbf{x}_i^T, \mathbf{0}^T)^T - \mathbf{y}_j)||^2 = \sum_{h=1}^{k}(x_h^{(i)} - y_h^{(j)})^2 + \sum_{h=k+1}^{m} y_h^{(j)2} = d_{ij}^2 \ .$$

Assume now that for any fixed j and arbitrary number of points $\mathbf{x}_i$ all true coordinates for $h = 1, \cdots, k$ are known, implying that the first sum is known in each equation, and we want to determine the remaining coordinates for $\mathbf{y_j}$, the above is then

$$\underbrace{\sum_{h=1}^{k}(x_h^{(i)} - y_h^{(j)})^2}_{e_{ij}} + \sum_{h=k+1}^{m} y_h^{(j)2} = d_{ij}^2 \ \Rightarrow$$

$$\sum_{h=k+1}^{m} y_h^{(j)2} = d_{ij}^2 - e_{ij} \ \forall \ i.$$

But since the left hand side only depend on j, we only have 1 independent equation (per j) regardless of how many points $\mathbf{x}_i$ we have. Since we don't measure distances between points $\mathbf{y}_j$, adding more such points does not help. Thus only the norm of the last $(m-k)$ coordinates can be computed, i.e., the distance from y to the lower dimensional subspace. As this does not depend on m, we only need a solver for the case $m = k + 1$. □

Naturally this also mean that for differences larger than 1, we get even more ambiguous solutions. If for example the lower dimension is 1 and the higher is 3 each higher-dimensional point can lie anywhere on a specified circle around the line.

7.2 Minimal Cases in Difference in Dimension

Each pair $(\mathbf{r}_i, \mathbf{s}_j)$ give a measurement f_{ij} and there are $\mathcal{D}_r$ unknowns for each $\mathbf{r}$ and $\mathcal{D}_s + 1$ unknowns for every $\mathbf{s}$. The number of unknowns per sensor is at most $\mathcal{D}_\vee = \max(\mathcal{D}_r, \mathcal{D}_s)$. If we set $\mathcal{D}_\wedge = \min(\mathcal{D}_r, \mathcal{D}_s)$ the following must hold for Problems 7.1.2 and 7.1.3 to be solvable

$$kn \geq \mathcal{D}_r k + (\mathcal{D}_s + 1)n - \frac{(\mathcal{D}_\wedge + 1)\mathcal{D}_\wedge}{2} \ . \tag{7.2}$$

The kn, $\mathcal{D}_r k$ and $\mathcal{D}_s$ terms are straightforward. The final term comes from the ambiguity in coordinate system and is as follows: Place the first lower dimensional coordinate at the origin, place the second along the first axis, the third in the plane spanned by the first and second axis, continue until the entire subspace is defined and place all remaining lower dimensional points in the subspace.

It is shown in [9] that the underlying TOA difference in dimension case requires

$$1 + \mathcal{D}_\wedge + \mathcal{D}_\wedge(\mathcal{D}_\wedge + 1)/2, \tag{7.3}$$

of sensors in the lower dimension to be solvable. It is straightforward to confirm that the generalization with offsets does not alleviate this requirement. This together with (7.2) give us the necessary requirements on k and n and all solvable cases for Problem 7.1.2 and 7.1.3 are summarized in Figure 7.2a and 7.2b respectively.

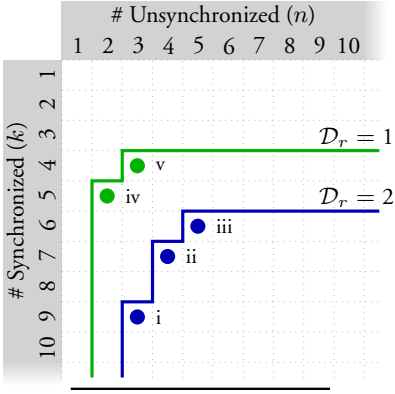
7.3 Solution

To derive solvers for feasible k and n we will employ rank constraint strategies introduced in [43] and [42] and modify them for the difference in dimension setting. In the cases where the offsets can be completely solved separately from the remaining unknowns we will use methods from [9] to solve for the remaining unknowns. For cases where the offset cannot be computed separately we will show how the rank constraints can be used in conjunction with other constraints to obtain the full solution. The implementations are based on techniques from [13].

7.3.1 Rank Constraints

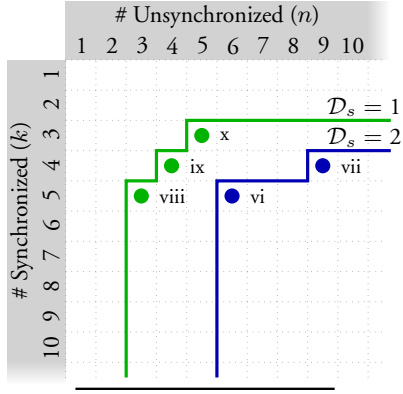
From now on, we will assume that $\mathbf{r}_i \in \mathbb{R}^{\mathcal{D}_\wedge}$, i.e., the receivers are placed in the lower dimension. As per Lemma 7.28, we restrict ourselves to $\mathcal{D}_\vee - \mathcal{D}_\wedge = 1$. The rank constraint strategy requires a reformulation of the measurement equations, as well as some observations on their relations to the locations of the sensors. Assuming the coordinates of the lower dimension is "zero-padded", i.e. $\mathbf{r}_i = [x_1, \ldots, x_{\mathcal{D}_r}, 0]^T$, as per the previous discussions we have

$$(f_{ij} - o_j)^2 = (\mathbf{r}_i - \mathbf{s}_j)^T(\mathbf{r}_i - \mathbf{s}_j). \tag{7.4}$$



Case	(k, n)	Solutions
i	$(9, 3)$	75
ii	$(7, 4)$	1
iii	$(6, 5)$	10
iv	$(5, 2)$	10
v	$(4, 3)$	6

(a) Solvable configurations for Problem 7.1.2.



Case	(k, n)	Solutions
vi	$(5, 6)$	5
vii	$(4, 9)$	NA
viii	$(5, 3)$	1
ix	$(4, 4)$	5
x	$(3, 5)$	16

(b) Solvable configurations for Problem 7.1.3.

Figure 7.2: Summary of all solvable cases for difference in dimension TDOA. All configurations (k, n) below the green curves mark solvable cases when the lower dimension is 1, all configurations below the blue lines mark solvable configurations when the lower dimension is 2. The table gives the properties of the configurations.

If we introduce vectors

$$\mathbf{R}_i = \begin{bmatrix} 1 & \mathbf{r}_i^T & \mathbf{r}_i^T\mathbf{r}_i \end{bmatrix}^T \text{ and } \mathbf{S}_j = \begin{bmatrix} \mathbf{s}_j^T\mathbf{s}_j - o_j^2 & \mathbf{s}_j^T & 1 \end{bmatrix}^T , \tag{7.5}$$

and collect the $\mathbf{R}_i$ vectors into an $(\mathcal{D}_\vee + 2) \times k$ matrix $\mathbf{R}$ and the $\mathbf{S}_j$ in an $n \times (\mathcal{D}_\vee + 2)$ matrix $\mathbf{S}$, we get

$$\mathbf{F} = \mathbf{R}^T\mathbf{S} , \tag{7.6}$$

where $\mathbf{F}$ is a matrix containing $\{f_{ij}^2 - 2f_{ij}o_j\}$. Due to the zeros in $\mathbf{r}_j$ we get a a row of zeros in $\mathbf{R}$. This shows that the rank of the matrix $\mathbf{F}$ is bounded by $(\mathcal{D}_\vee + 2)$, constraining the offsets o_j without involving the other unknowns. Using a trick from [42], it is possible to exploit the structure of $\mathbf{R}$ and $\mathbf{S}$ to obtain tighter rank constraints. We introduce two matrices $\mathbf{C}_R$ and $\mathbf{C}_S$ both on the form $[-\mathbf{1}\ \ \mathbf{I}]^T$ that by the operations $\hat{\mathbf{R}}^T = \mathbf{C}_R^T\mathbf{R}^T$ and $\hat{\mathbf{S}} = \mathbf{S}\mathbf{C}_S$ turns the rows of ones into zeros. In essence this operation will subtract the first column of $\mathbf{R}$ from the remaining columns, and the first column of $\mathbf{S}$ from the remaining columns of $\mathbf{S}$. By the construction of $\mathbf{R}$, the row of zeros caused by the 0 entries in $\mathbf{r}_i$ is not in the last, nor the first row of $\mathbf{R}$. We have then after the above operations two rows of zeros in $\hat{\mathbf{R}}$.

Applying the above matrices we get the final system

$$\mathbf{C}_R^T\mathbf{F}\mathbf{C}_S = \hat{\mathbf{R}}^T\hat{\mathbf{S}} , \tag{7.7}$$

As the last row in $\hat{\mathbf{R}}$ is multiplied with the row zeros in $\hat{\mathbf{S}}$ we can remove both without affecting the product. The same holds for the 2 rows of $\hat{\mathbf{S}}$ corresponding to the 2 rows of zeros in $\hat{\mathbf{R}}$ and all 4 can be removed. This leaves only $\mathcal{D}_\vee - 1 = \mathcal{D}_\wedge$ rows in $\hat{\mathbf{S}}$ and consequently $\hat{\mathbf{F}}$ has at most rank $\mathcal{D}_\wedge = \mathcal{D}_r$.

The entries of $\hat{\mathbf{F}}$ are of the form

$$\hat{f}_{ij} = g_{ij} - g_{0j} - g_{i0} + g_{00}, \tag{7.8}$$

where

$$g_{ij} = f_{i+1,j+1}^2 - 2f_{i+1,j+1}o_{j+1}. \tag{7.9}$$

It is straightforward to confirm that if $\mathbf{s}_j$ would lie in the lower dimension, a row of $\mathbf{S}$ would be zero, and would remain after reduction of $\hat{\mathbf{S}}$. This would again result in a matrix $\hat{\mathbf{F}}$ of rank $\mathcal{D}_\wedge$, but with $\mathcal{D}_\wedge = \mathcal{D}_s$.

We can enforce the rank constraint on the sub-matrices of $\hat{\mathbf{F}}$. Specifically all submatrices of size $(\mathcal{D}_\wedge+1)\times(\mathcal{D}_\wedge+1)$ will have rank $\mathcal{D}_\wedge$. This gives equivalently constraints on the determinants of the set of $(\mathcal{D}_\wedge + 1) \times (\mathcal{D}_\wedge + 1)$ sub-matrices $\Lambda_{\mathcal{D}_\wedge+1}$:

$$\det \mathbf{Q} = 0,\ \forall \mathbf{Q} \in \Lambda_{\mathcal{D}_\wedge+1}. \tag{7.10}$$

For a $(k-1) \times (n-1)$ matrix $\hat{\mathbf{F}}$, the number of constraints N_c is

$$N_c = |\Lambda_{\mathcal{D}_\wedge+1}| = \binom{k-1}{\mathcal{D}_\wedge+1}\binom{n-1}{\mathcal{D}_\wedge+1}.$$

As can be seen from (7.8) and (7.9) the entries of $\hat{\mathbf{F}}$ are linear in the unknown offsets $\{o_1, \ldots, o_n\}$, so each constraint of type (7.10) is a polynomial equation of degree $\mathcal{D}_\wedge + 1$.[1]

In general, for a case with k receivers and n transmitters, with the minimal affine span of the two as $\mathcal{D}_\wedge$, there are $N_o = (k-1-\mathcal{D}_\wedge)(n-1-\mathcal{D}_\wedge)$ linearly independent constraints of type (7.10), see [43, 42]. For cases where $n = N_o$, determining the offsets using only the rank constraints is minimal and well-defined. For $\mathcal{D}_\wedge = 1$ cases, these correspond to $(4,4), (5,3)$. And for the $\mathcal{D}_\wedge = 2$ cases, (7,4) and (5,6) are the two minimal problems for determining the offsets. Note that such properties are independent of $\mathcal{D}_s$ and $\mathcal{D}_r$.

For cases where $N_o > n$, the rank constraints are overdetermined for the offsets. There are two ways to estimate the offsets using these overdetermined set of equations. The first one is to utilize the fact that there exist a unique solution to the overdetermined system, using techniques from [56], the offsets can be solved linearly. The second scheme is to ignore a subset of constraints such that the remaining constraints render the problem minimal and well-defined. One possible drawback of this scheme is the possible existence of multiple solutions.

For the minimal TDOA cases where offsets can be determined using only the rank constraints, i.e. (4,4), (5,3), (7,4) and (5,6), the full problem can be solved by combining the corresponding linear difference in dimension TOA solver from [9]. Again accounting for the inherent ambiguity of the last coordinate in the high dimensional space, the linear solver is unique and the number of solutions is entirely dependent on the number of solutions of the offset equation. These

[1] Note that in the (5,2) case no such submatrices exist. We will deal with this case in Section 7.3.3.

are summarized in Figure 7.2. In a few cases there are multiple valid solutions to a given set of measurements, but in general the excess solutions are complex and can be directly discarded.

As for the cases where the rank constraints give under-determined systems, one needs to exploit other, often non-linear constraints.

7.3.2 Distance Equations

We here derive additional non-linear equations on the offsets. Just like before, we assume that the receivers lie in the lower-dimensional space. The opposite case can be handled analogously.

According to [43], each factorization of $\hat{\mathbf{F}} = \hat{\mathbf{R}}^T\hat{\mathbf{S}}$ provides the receiver and transmitter coordinates up to an coordinate change described by a full rank matrix $\mathbf{L}$ and translation $\mathbf{b}$. Let $\tilde{\mathbf{R}}$ be the first $\mathcal{D}_\wedge$ columns of the rank-$\mathcal{D}_\wedge$ matrix $\hat{\mathbf{F}}$ which is parameterized by the offsets $\mathbf{o}$. This then corresponds to a choice of factorization that has the identity matrix on the corresponding places in $\hat{\mathbf{S}}$. Based on this and the formulation in (7.7), we can write the positions of the receivers $\mathbf{r}_i = \mathbf{L}\tilde{\mathbf{r}}_i(\mathbf{o})$. Following the derivation in [43] this gives the following constraints on the unknown transformation $\mathbf{H}$ and translation $\mathbf{b}$ for $i = \{1, \ldots, m-1\}$,

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

where $d_{ij} = f_{ij} - o_j$, $\mathbf{H} = (\mathbf{L}^T\mathbf{L})^{-1} \in \mathbb{R}^{\mathcal{D}_\wedge \times \mathcal{D}_\wedge}$ and $\mathbf{b} \in \mathbb{R}^{\mathcal{D}_\wedge}$. Since the equations are linear in the entries in $\mathbf{H}$ and $\mathbf{b}$, the system can be rewritten as

$$\mathbf{W}\begin{bmatrix}\mathbf{h}\\ \mathbf{b}\\ 1\end{bmatrix} = 0, \tag{7.12}$$

where $\mathbf{W}$ is a $(m-1) \times k$ matrix parameterized by the offsets and $\mathbf{h}$ is the vector representation of the unknowns in $\mathbf{H}$. Here $k = \mathcal{D}_\wedge(\mathcal{D}_\wedge + 1)/2 + \mathcal{D}_\wedge + 1$. From (7.12), we know that all $k \times k$ sub-determinants of $\mathbf{W}$ are equal to 0. By forming these equations, we remove the unknowns $\mathbf{h}$ and $\mathbf{b}$ and reduce (7.11) to a polynomial system of only n unknowns. Combining these equations with the rank constraints, one arrives at a set of well-defined equations for the offsets. In principle, both the (3,5) (9,3) as well as the (4,9) cases can be solved using this formulation. Fast and stable solvers have been implemented based on Gröbner basis methods for (3,5) and (9,3) cases. Efficient solvers for the (4,9) case is still

difficult to derive due to the large number of unknowns (9 offsets) and high degree (degree 9). Such idea can also be extended to cases where $\mathcal{D}_\wedge = 3$. The number of solutions for these cases using the above solving strategy are presented in Figure 7.2.

7.3.3 The (5,2) Case

As mentioned above, the case (5,2) has no 2×2 submatrices. The reason is that the compacted version of its measurement matrix is of size 4×1. To work around this, we solve it using manipulations of its equations directly. Let the 5 $\mathbf{r}_i$ be placed on the x-axis with at $u_1 = 0$, u_{2-5}, and $\mathbf{s}_j$ at coordinates (x_j, y_j) Equation (7.4) is

$$f_{ij}^2 - 2f_{ij}o_j + o_j^2 = x_j^2 - 2x_j u_i + u_i^2 + y_j^2 \,. \tag{7.13}$$

Subtracting the first equation, $u_1 = 0$ this can be written in matrix form as

$$\begin{pmatrix} f_{2j} - f_{1j} & u_2 & u_2^2 - f_{2j}^2 + f_{1j}^2 \\ f_{3j} - f_{1j} & u_3 & u_3^2 - f_{3j}^2 + f_{1j}^2 \\ f_{4j} - f_{1j} & u_4 & u_4^2 - f_{4j}^2 + f_{1j}^2 \\ f_{5j} - f_{1j} & u_5 & u_5^2 - d_{5j}^2 + f_{1j}^2 \end{pmatrix} \begin{bmatrix} -2o_j \\ 2x_j \\ -1 \end{bmatrix} = \mathbf{0}, \tag{7.14}$$

i.e. each transmitter gives us a rank deficient matrix only dependent on the positions of the receivers. Using determinants of the submatrices we get equations only in u_{2-5}, that allows us to solve for these unknowns. The linear case (4,3) has a similar constraint, but with one less row for its matrices.

7.4 Summary

The classification of all cases is shown in Figure 7.2. Two cases were solved using direct manipulation of the distance equations in [1]. Some cases are overdetermined by one equation, however further reducing either k or n would make the system underdetermined and thus unsolvable. As described above this sometimes allows for linear solvers to be employed. In general the resulting systems have relatively low total degree and few solutions, with the exception of case (i). Case (vii) is even more complex and using the presented strategy we were unsuccesful in constructing a solver that displayed good numerics.

7.5 Experiments

We will present the numerical stability for all implemented solvers using generated examples. We further will present results on real data using a microphone setup in 2D, with sounds in 3D. The accuracy of the solution will be measured by comparing it to a 3D reconstruction from images. The visual reconstruction is obtained using standard techniques from computer vision.

7.5.1 Numerical Stability

Synthetic data is generated by randomly placing sensors in a $[0, 1]$ cube, meeting the requirements of dimesnionality assumed by the solvers. The solver for case (i) requires that the original equations are expanded to a 1400 by 500 coefficient matrix, and it has very poor stability even if no noise is added. Typical accuracy without noise is RMS on the order of 10^{-4}. All other solvers had consistent accuracy of the order 10^{-10} to 10^{-13} with the exception of (x) that on rare occasions had values of 10^{-2}, skewing its mean quite severely. We believe this is caused by close to degenerate configurations. This behavior is also visible in the presence of noise, as illustrated in Figure 7.3. The figure shows the mean over 200 cases for different levels of relative added gaussian noise, applied to the measurements. The RMS is calculated against the generated ground truth (GT). Again the poor performance of (x) is due to single events with substantially less accurate result.

7.5.2 Reconstruction of Microphone Array

A total of 8 microphones are placed on a floor (2D), see Figure 7.4, and sequences of distinct sounds generated from several locations in the room (3D). The sounds are far enough apart to be distinct in the matching, but due to echoes, disturbances exact time differences are unavailable, and in some cases the matches are bad enough to be considered outliers. We then use the (6,5) minimal solver in a RANSAC-like algorithm. As a final step the solution is locally optimized using all found inliers. The result is very promising with an RMS of 6.7cm in microphone positions between the visual and audio based reconstructions. The reconstructed path for the sound source is consistent with the dimensions of the room, and form a smooth track. The reconstructed layout is illustrated in Figure 7.5.

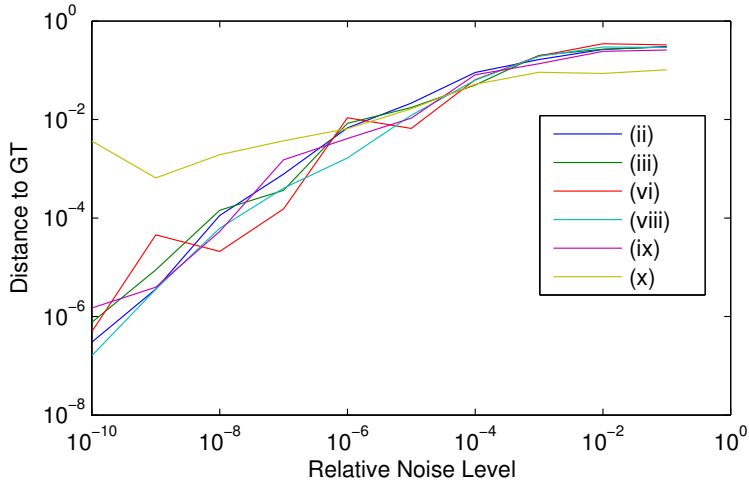


Figure 7.3: Stability of the derived solvers, except (i),(iv) and (v).

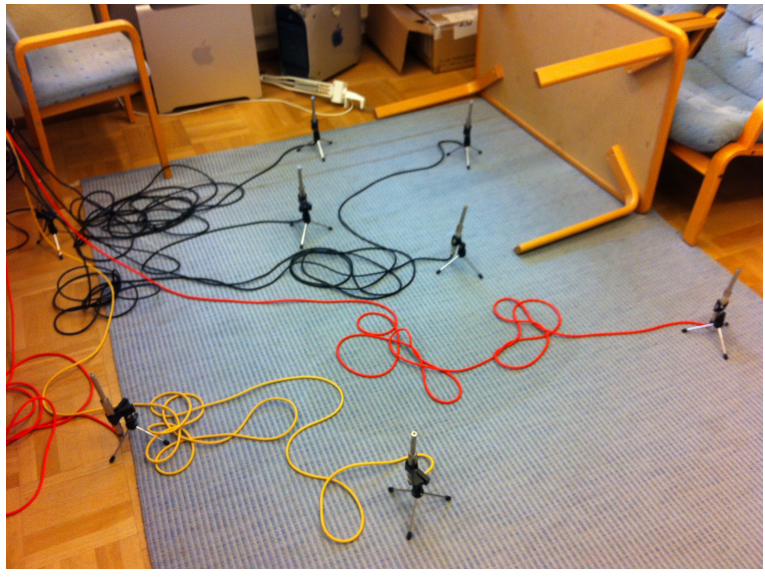


Figure 7.4: Microphones placed on office floor.

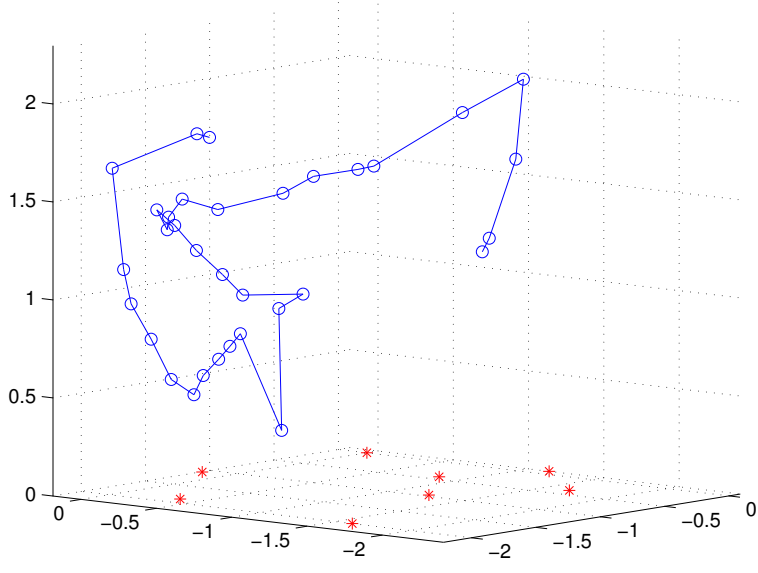


Figure 7.5: Reconstructed layout of microphone positions (red stars) and motion trajectory of sound sources (blue circles and line), all units in meter.

7.6 Conclusions

We have classified all solvable minimal cases in a difference in dimension TDOA setting. Further we have devised solution strategies and implemented solvers for most of these cases. With the exception of 2 solvers the overall performance is excellent, and one of the bad solvers still maintain a very high success rate.

Chapter 8

Far Field Approximation as Initialization for TDOA Based Calibration

In the previous chapter we used a limitiation in dimension of one set of sensors to reduce the number of unknowns and derive solvers for the true minimal cases. In this chapter we will study another specific setup that allows for the construction of efficient solvers, far field approximation. This approximation is used for initialization to the calibration problem. We use a similar factorization as [62] but in three dimensions, and show that far field approximation requires at least four measurement positions, i.e. three motions, and measurements to at least six real or virtual transmitters. Apart from this we also describe the failure modes of the algorithm and show what can be done when such configurations are present. We further propose two optimization strategies for more thorough calibration and evaluate them with regard to accuracy and convergence rate. Several test cases are simulated in which we validate the far field approximation and the performance of the proposed algorithms.

8.1 Determining pose

In the following treatment, we make no difference between real and virtual transmitters or receivers. Assume that the transmitters are stationary at positions $\mathbf{s}_j \in \mathbb{R}^3$ and that the receiver are moving with position $\mathbf{r}(t) \in \mathbb{R}^3$ at time t. As we are only interested in discrete time points we will use the notation $\mathbf{r}_i$. This makes a moving receiver directly analogous to the formulation in earlier chapters where several different transmitters and receivers at different fixed positions where used. By measuring the signal with known base band frequency one obtains

a complex constant, whose phase depends on the distance $d = |\mathbf{r} - \mathbf{s}|$ between the transmitter and the receiver. Furthermore if during measurements we have for any $\mathbf{r}$, $\mathbf{r}_0$ and $\mathbf{s}$ that

$$\|\mathbf{r} - \mathbf{r}_0\| \ll \|\mathbf{r} - \mathbf{s}\|, \tag{8.1}$$

it is reasonable to approximate the distance

$$\begin{aligned} d_{i,j} &= \|\mathbf{r}_i - \mathbf{s}_j\| \\ &\approx \|\mathbf{r}_0 - \mathbf{s}_j\| + (\mathbf{r}_i - \mathbf{r}_0)^T \mathbf{n}_j \\ &= \mathbf{r}_i^T \mathbf{n}_j + \underbrace{(\|\mathbf{r}_0 - \mathbf{s}_j\| - \mathbf{r}_0^T \mathbf{n}_j)}_{\bar{o}_j} \ . \end{aligned} \tag{8.2}$$

Here $\mathbf{r}_0$ is any choice of the receivers such that Equation (8.1) holds, and $\mathbf{n}_j$ is the direction from this receiver towards the transmitter, now assumed to be constant with unit length. As stated in previous chapters a TDOA measurement has the form

$$f_{i,j} = \|\mathbf{r}_i - \mathbf{s}_j\| + \tilde{o}_j \, , \tag{8.3}$$

and by setting $o_j = \tilde{o}_j + \bar{o}_j$ one obtains the far field approximation

$$f_{i,j} \approx \mathbf{r}_i^T \mathbf{n}_j + o_j. \tag{8.4}$$

The central problem is the following far field time difference of arrival (FFTDOA) type problem that arise from this approximate relative distance measurement.

Problem 8.1.1 Given measurements $F_{i,j}$, $i = 1, \ldots, m$ and $j = 1, \ldots, k$ from m receiver positions to k transmitters, determine both the positions $\mathbf{r}_1, \ldots, \mathbf{r}_m$ and the directions $\mathbf{n}_1 \ldots \mathbf{n}_k$ from the so that

$$\begin{aligned} F_{i,j} &= \mathbf{r}_i^T \mathbf{n}_j + o_j \\ ||\mathbf{n}_j||_2 &= 1, \end{aligned} \tag{8.5}$$

where o_j is a constant distance offset for each transmitter.

This is the far-field approximation of the full TDOA problem seen in previous chapters.

From the formulation above we see that unknowns offsets occur already in the basic distance equations, and hence we have offsets for both the far-field TDOA and far-field TOA problems. This leads us to the following lemma.

Lemma 8.29. *A problem with m receivers and k transmitters with unknown offset o_j can be converted to a problem with $m-1$ receivers and k transmitters with known offset.*

Proof. Note that because of the unknown constant o_j the problem does not change in character by modification $D_{i,j} = F_{i,j} - K_j$. For simplicity we set

$$D_{i,j} = F_{i,j} - F_{1,j} = \mathbf{r}_i^T \mathbf{n}_j + o_j - \mathbf{r}_1^T \mathbf{n}_j - o_j. \tag{8.6}$$

Since the coordinate system is arbitrary we can set $\mathbf{r}_1 = \mathbf{0}$, yielding

$$D_{i,j} = \mathbf{r}_i^T \mathbf{n}_j, \text{ for } i = 2, \ldots, m \tag{8.7}$$

□

For simplicity we will in the sequel assume that one receiver has been used to eliminate the offsets. Denote by D the resulting measurement matrix. This converts the FFTDOA problem into a FFTOA problem, i.e.

Problem 8.1.2 Given measurements $D_{i,j},\ i = 1, \ldots, (m-1), j = 1, \ldots, k$ from receivers at $m-1$ different positions to k transmitters, determine both the positions $\mathbf{r}_i$ of the receivers and the directions to the transmitters $\mathbf{n}_j$ so that

$$\begin{aligned} D_{i,j} &= \mathbf{r}_i^T \mathbf{n}_j \\ ||\mathbf{n}_j||_2 &= 1. \end{aligned} \tag{8.8}$$

Lemma 8.30. *If $\mathbf{r}_i$ and $\mathbf{s}_j$ lie in $\mathbb{R}^3$, then the matrix D with elements $D_{i,j}$ is of rank at most 3.*

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

$$Z = \begin{pmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \vdots \\ \mathbf{r}_m^T \end{pmatrix} \tag{8.9}$$

and

$$N = \begin{pmatrix} \mathbf{n}_1 & \mathbf{n}_2 & \ldots & \mathbf{n}_k \end{pmatrix} \tag{8.10}$$

we see that $D = ZN$. Both $Z \in \mathbb{R}^{(m-1)\times 3}$ and $N \in \mathbb{R}^{3\times k}$ have at most rank 3, therefore the same holds for D. □

Assuming that k and m are large enough and assuming that the motion $\mathbf{r}_i$ and the directions $\mathbf{n}_j$ are in general enough constellation the matrix D will have rank 3. If so it is possible to reconstruct both Z and N up to an unknown linear transformation. This can be done using singular value decomposition, $D = USV^T$. Even with noisy measurements, the closest rank 3 approximation in the L_2-norm can be found using the first 3 columns of U and V. By setting $\tilde{Z} = U_3$ and $\tilde{N} = S_3V_3^T$ we get all possible solutions by $N = A\tilde{N}$, with A a general full rank 3×3 matrix. Changing 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

$$\tilde{\mathbf{n}}_j^T A^T A \tilde{\mathbf{n}}_j = 1, \tag{8.11}$$

which after substitution $B = A^T A$ becomes linear

$$\tilde{\mathbf{n}}_j^T B \tilde{\mathbf{n}}_j = 1 \tag{8.12}$$

in the unknown elements of B. Since symmetric 3×3 matrices have 6 degrees of freedom we need at least 6 transmitters to determine the matrix uniquely. We have no additional requirements on m other than Z being a rank 3 matrix so atleast m=4 recievers are necessary. Once B has been determined A can be determined by Cholesky factorization. This gives the transformation A up to an unknown rotation and possible mirroring of the coordinate system. We summarize the above in the following theorem.

Theorem 8.31. *The minimal case for reconstructing m positions $\mathbf{r}_i$ and k orientations $\mathbf{n}_j$ from relative distance measurements $F_{i,j}$ as formulated in Problem 8.1.1 is $m = 4$ and $k = 6$.*

Accordingly a solver for the minimal case is given in Algorithm 7. Note that using minimal information $m = 4$ and $k = 6$ results in estimates that fulfill the measurements exactly (up to machine precision) even if the measurements are disturbed by noise.

8.1.1 Failure modes of the algorithm

It is interesting and enlightening to know the failure modes of the algorithm. This is captured by the following theorem.

Algorithm 7 Minimal case solver

Given the measurement matrix D of size 4×6.

1. Set $D_{i,j} = F_{i,j} - F_{1,j}$
2. Remove the first row of D
3. Calculate a singular value decomposition $D = USV^T$.
4. Set $\tilde{Z}$ to first 3 columns of U and $\tilde{N}$ to first three columns of SV^T.
5. Solve for the six unknowns in the symmetric matrix B using the 6 linear constraints $\tilde{n}_j^T B \tilde{n}_j = 1$.
6. Calculate A by Cholesky factorization of B, so that $A^T A = B$.
7. Transform motion according to $Z = \tilde{Z}A^{-1}$ and structure according to $N = A\tilde{N}$.

Theorem 8.32. *The minimal case for reconstructing k orientations $\mathbf{n}_j$ and m positions $\mathbf{r}_i$ from relative distance measurements $F_{i,j}$ as formulated in Problem 8.1.1 is for $m = 4$ and $k = 6$. 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 structure $\mathbf{n}_j$ and motion $\mathbf{r}_i$ up to an unknown translation, orientation and reflection of the coordinate system.*

Proof. The algorithm can fail if the transformed measurement matrix D has rank 2 or lower. This happens if Z or N in $D = ZN$ has rank 2 or lower. Let us start with the first case. The rows of Z contains $r_i - r_1$ for three different i, so it has full rank unless all four involved receivers lie in a plane. Similarly N fails to have full rank if all directions $\mathbf{n}_j$ are coplanar.

The algorithm can also fail if there are two solutions to the matrix B in $\mathbf{n}_j^T B \mathbf{n}_j = 1$. But then the difference $\Omega = B_1 - B$ of these two solutions is a three by three matrix for which

$$\mathbf{n}_j^T \Omega \mathbf{n}_j = 0, \tag{8.13}$$

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

Yet another type of failure mode of the algorithm is if the data is corrupted by noise or far field approximation is not valid, so that the matrix B obtained is not positive definite. Then the algorithm fails because there is no Cholesky factorization of B into $A^T A$. If B was uniquely determined, there are no real solution to the problem in this case.

8.1.2 Analysis of failure modes

If the rank of the matrix D 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 $Z = U_2 A$ and $N = A S_2 V_2^T$ up to an unknown 2×2 matrix A representing a choice of affine coordinate system. Here we do get inequality constraints that

$$\left| A \begin{pmatrix} n_{j,x} \\ n_{j,y} \end{pmatrix} \right| \leq 1. \tag{8.14}$$

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

$$n_{j,z} = \pm \sqrt{1 - n_{j,x}^2 - n_{j,y}^2}. \tag{8.15}$$

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 $\mathbf{n}_j^T A^T A \mathbf{n}_j = 1$ 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 $k = 3$. 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 the points $\mathbf{r}_i$.

If the rank is 1, this could be because the directions are parallel. In this case. 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 $D_{i,j}$ are such coordinates by definition.

If the rank is 1 because the points lie on a line, we obtain a one-parameter family of reconstructions based on $Z = U_1 a$ and $N = aS_1V_1^T$, where a is an unknown constant that has to fulfill $a \leq 1/l$, where $l = \max_i |S_1V_{1,i}|$. 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.

8.1.3 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 for the pose problem:

Problem 8.1.3 Given measurements $D_{i,j},\ j = 1, \ldots,\ (m-1)$ and $i = 1, \ldots, k$ from receivers at m different positions to k transmitters, determine both the positions $\mathbf{r}_i$ and the directions $\mathbf{n}_j$ so that

$$\begin{array}{cc} \min_{Z,N} & ||D - Z^T N||^2_{Frob} \\ s.t. & ||\mathbf{n}_j||_2 = 1,\ i = 1, \ldots, k. \end{array} \tag{8.16}$$

where $||.||_{Frob}$ denotes the Frobenius norm.

For the over-determined cases, that is $m > 4$ and $k \geq 6$ or $m \geq 4$ and $k > 6$, it is possible to modify Algorithm 7 to obtain an efficient but not necessarily optimal algorithm that finds a reconstruction that fits the data quite good 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 B in Step 6 can be performed in a least squares sense and

(iii) re-normalize the columns of 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 B not being positive semi-definite can be attacked by non-linear optimization. Here we try to optimize A so that $\sum_{i=1}^{k} (\tilde{\mathbf{n}}_j^T A^T A \tilde{\mathbf{n}}_j - 1)^2$ is minimized. This can be achieved e.g. by initializing with $A = I$ and then using non-linear 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 Z and N, and then it alternates between optimizing Z given N and vice versa. The latter is essentially a method combining Gauss-Newton algorithm and a gradient descent that improve the solution locally. 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 8.1.3, we can use a coordinate descent scheme. Specifically, we would like to iteratively optimize the cost function in Problem 8.1.3 with respect to Z given N, and then find the optimal feasible N with fixed Z. If we initialize N such that it satisfies the norm constraints, we can easily see that the alternating procedure is converging (Algorithm 8).

Algorithm 8 Alternating optimiziation

Given the measurement matrix D with $m > 4$ and $k \geq 6$ or $m \geq 4$ and $k > 6$,

1. Construct D and initialize Z and N as in Algorithm 7
2. Fix N, find optimal Z
3. Fix Z, solve the constrained minimization for each $\mathbf{n}_i$, $i = 1, ..., k$
4. Repeat (2) and (3) until convergence or predefined number of iterations is reached

To enable the alternating optimization, we need to solve two separate optimization problems. The first one is to find the optimal Z given N. This is the classic least squares problem and is known to be convex and can be solved efficiently. On the other hand, solving for optimal $\mathbf{n}_j$ given Z is not always convex due to the additional norm 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 the small constrained problems (3 variables each) independently with interior point method. Alternatively, we can solve the constrained optimization as solving poly-

nomial equations. This can be related to the fact that for a given Z, 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 N and Z simultaneously. The difficulty here is again the constraints on the direction vectors $\mathbf{n}_j$. The key idea here is to re-parameterize the orientation vectors. Given a direction vector $\mathbf{n}$ having unit length, any direction vectors can be represented by $\mathbf{n} \cdot exp(S)$, where 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) orientation $\mathbf{n}$ as axis direction, any local change of the orientation on the sphere can be easily parameterized via the exponential map. Therefore, the gradient of D_{ij} with respect to $\mathbf{n}_j$ can be expressed without any constraints. We can then construct the Jacobian for the Levenberg-Marquardt algorithm to compute the optimal descent direction. In the following, we use $\mathbf{y}$ to denote the vector formed by stacking variables in Z and N, $\bar{\mathbf{d}}$ is the vectorized version of D based on the ordering of g. The algorithm is given in Algorithm 9

8.2 Experimental Validation

In this section, we present comprehensive experimental results for simulated data. We focus on the performance of the minimal solver, verification of the far field approximation as well as the comparisons between solvers for overdetermined cases.

Algorithm 9 Levenberg-Marquardt

Given the measurement matrix D (over-determined), initialize $\mathbf{y}$ and construct $\bar{\mathbf{d}}$ as in Algorithm 7,

1. Compute the Jacobian of $\bar{\mathbf{d}}$ with respect to $\mathbf{y}$, $J = (\frac{\partial \bar{D}_{11}}{\partial \mathbf{y}}, \ldots, \frac{\partial \bar{D}_{ij}}{\partial \mathbf{y}}, \ldots, \frac{\partial \bar{D}_{(m-1)k}}{\partial \mathbf{y}})$
2. Calculate $\Delta\mathbf{y} = (J^T J + \lambda \cdot \text{diag}(J^T J))^{-1} J^T \Delta\bar{\mathbf{d}}$, where $\Delta\bar{\mathbf{d}}$ is the residue and λ a damping factor.
3. $\mathbf{y} = \mathbf{y} + \Delta\mathbf{y}$
4. repeat (1),(2) and (3) until convergence or predefined number of iterations is reached

8.2.1 Minimal Solver Accuracy

The numerical performance of the algorithm was evaluated by generating problems where the far field approximation is true and not degenerate. In essence this constitutes generating directions $\mathbf{n}_j$ and relative distance measurements $D_{i,j}$ and culling cases where the three largest singular values of the measurement matrix are below a threshold or the directions lie on a conic. The error is then evaluated as the average norm-difference of the estimated receiver positions. The receiver positions were selected as the corners of a tetrahedron with arc-length one. The average error of 10000 such tests was $6.8 \cdot 10^{-15}$, close to machine epsilon.

8.2.2 Far Field Approximation Accuracy

Minimal Case

To evaluate the performance of the assumption that transmitters can be viewed as having a single common direction to receivers, data was generated using 3D positions for both transmitters and receivers at different relative distances in-between receivers and transmitters to receivers. The constellation of receivers is again the tetrahedron and transmitters are randomly placed on a sphere surrounding it. A graph showing the error, as defined in Section 8.2.1, as a function of radius of the sphere (that is relative distance), as well as the failure rate of the solver is shown

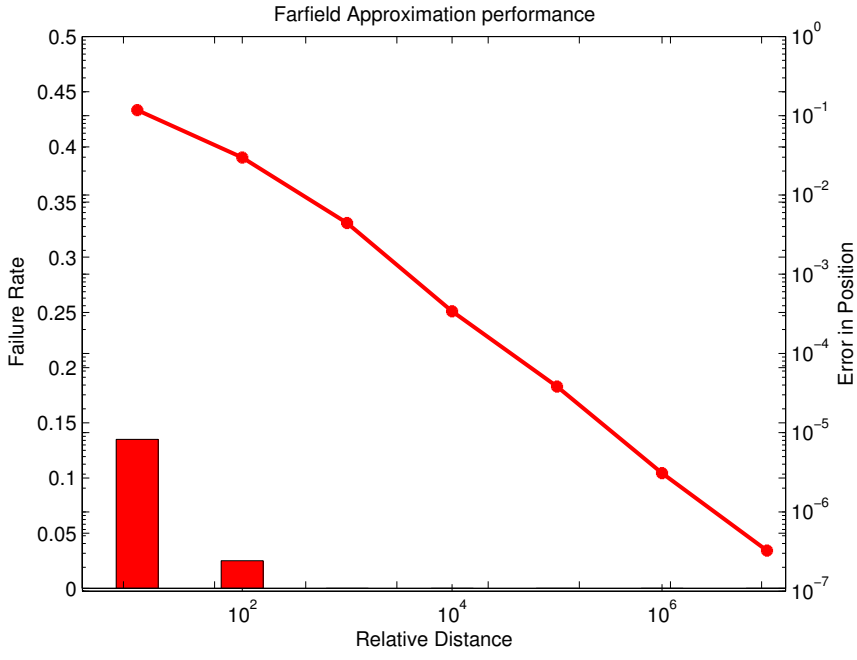


Figure 8.1: Performance on minimal case solver. Bars show failure rate (left y-axis) and curve shows the mean error in estimated position as a function of distance (right y-axis). Note that bar height has linear scale.

in Figure 8.1. A failure constitutes a case in which the B matrix in algorithm 7 is not positive definite. As can be seen this is infrequent even at small relative distances in when one would not expect a far field approximation to work. As can be expected the approximation gets better when the relative distance increases.

Initialization for Overdetermined Cases

As described in Section 8.1.3, Algorithm 7 can with some modifications be used on overdetermined cases without guarantees on optimality of the solution. In these situations the solutions serves as an initial guess of some other optimization method. The additional information should however give some numerical stability and it is interesting to evaluate the algorithm for initial guess estimates in overdetermined cases. To do this the synthetic dataset is augmented with ad-

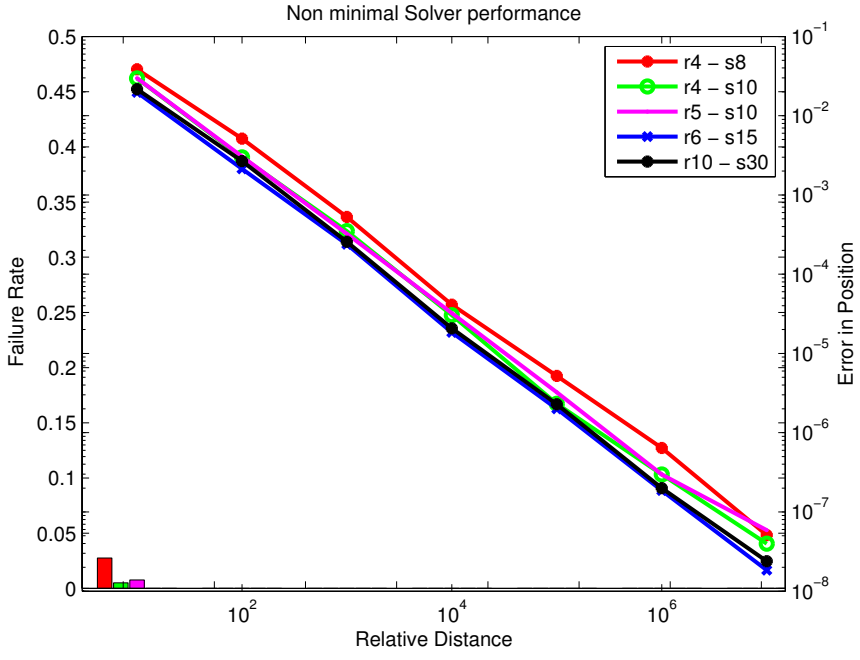


Figure 8.2: Performance on non-minimal cases. Bars show failure rates, line is error as a function of distance in loglog scale. Size of test-cases are noted in figure with rx-sy denoting x receivers and y transmitters. This plot is best viewed in color. Note the scale difference to the graph in Figure 8.1.

ditional randomly placed transmitters and receivers. The four first receivers are again the tetrahedron and the rest are randomly uniformly distributed within the unit cube. Senders are again placed on a sphere around the receivers. Results for different problem sizes are shown in Figure 8.2. One immediately notices that the failure ratio drops, in many cases to zero. One can also see that adding more data will (in general) result in smaller errors, for the cases shown here up to one order of magnitude smaller than a min case.

8.2.3 Overdetermined Cases

We also investigate the performance of the two schemes for over-determined cases. In all experiments below, we initialize both the alternating optimization and LMA

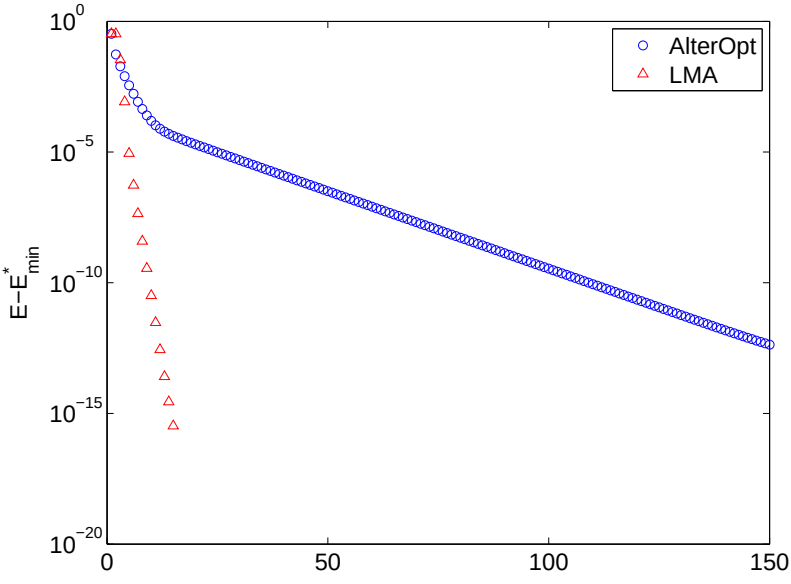


Figure 8.3: Convergence of alternating optimization and LMA on simulated TDOA measurements with gaussian white noise ($\sigma = 0.1$). Here $m = 10$ and $k = 10$.

based on the minimal solver modified for over-determined case. The simulated data is of a true far field approximation with gaussian white noise, i.e. measurements are simulated as $D_{i,j} = \mathbf{z}_i\mathbf{n}_j + \epsilon_{i,j}$ where $\epsilon_{i,j} \in N(0, \sigma)$ i.i.d. In Figure 8.3, we can see that alternating optimization and LMA all decrease the reconstruction errors compared to the minimal solver. On the other hand, from figure 8.3, LMA converges much faster than alternating scheme (20 vs. 150) and obtains relatively lower reconstruction errors. This verifies the superiority of LMA over coordinate descent. This observation is consistent over different m and k as well as a variety of noise levels. Note that here for all the experiments, we set the damping factor λ to 1.

It is also of interest to view the complete system when the measurements $D_{i,j}$ does not fulfill the far field approximation and when disturbed 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 $D_{i,j}$ are then simulated, perturbed with gaussian white

noise. Figure 8.4 shows the results. The pictures show that the initialization method is fairly good, but in many cases the LMA brings down the position error. The system is also fairly robust to noise.

8.3 Conclusions

In this chapter another simplification of the full general $3D$ TDOA problem has been analyzed. It was solved through reformulation into a TOA formulation, and solving of that problem. The proposed far field approximation to the calibration of TDOA and TOA sensor networks results in a factorization algorithm with constraints. The failure modes of the algorithm is studied and particular emphasis is made on what can be said when any of these failure conditions are met. The experimental validation gives a strong indication that a far field approximation is a feasible approach both for getting direct estimates as well as initial estimates for other solvers. Even considering that there are cases when the algorithm fails, obtained solutions are good even at small relative distances. This validation is done on 3D problems and confirms findings in [62] where evaluation was done in 2D.

Further two optimization schemes are analyzed and what difficulties may arise when employing them. Both of these schemes are experimentally evaluated and confirmed to successfully optimize the initial guess on a problem fulfilling the far field assumptions, although at quite different convergence rates. The faster of the two is also employed on cases when transmitters are given true locations and measurements are subject to noise with good results.

It would be interesting in future work to study to what extent it can be shown that the local optimum obtained to the problem can be proven to be global optimum. To integrate the solvers with robust norms is also worth studying to handle situations with outliers. It would also be interesting to verify the algorithms on real measured data and investigate the possibilities of using our algorithms in a RANSAC approach to remove potential outliers that may occur in real life settings.

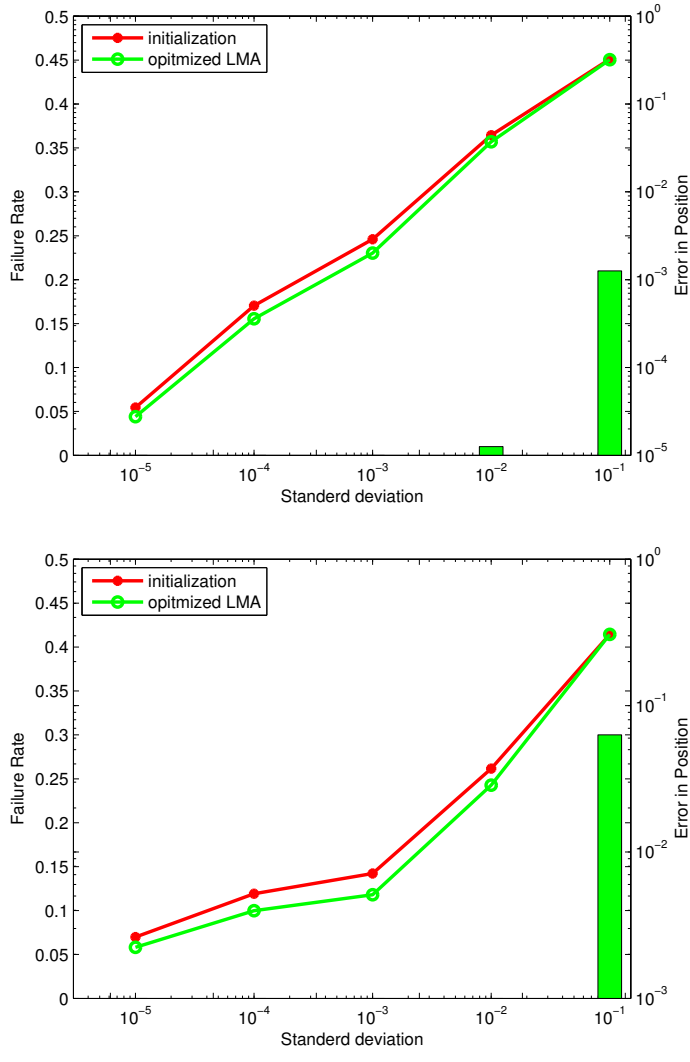


Figure 8.4: Performance on non-minimal cases with simulated TDOA measurements with gaussian white noise. The mean error in position of the receivers are plotted against the noise standard deviation. Here m=5 and k=10, and the relative distance to receivers and transmitters are 10^7 (top) and 10^2 (bottom). Failure rates for the initialization are also shown for completeness.

Bibliography

[1] Erik Ask, Simon Burgess, and Kalle Åström. Minimal structure and motion problems for toa and tdoa measurements with collinearity constraints. In *Proceedings of the 2nd International Conference on Pattern Recognition Applications*, pages 425–429, 2013. 106

[2] Erik Ask, Olof Enqvist, and Fredrik Kahl. Optimal geometric fitting under the truncated l2 -norm. In *CVPR 2013*, 2013.

[3] Erik Ask, Yubin Kuang, and Kalle Åström. Exploiting p-fold symmetries for faster polynomial equation solving. In *Proceedings of the 21st International Conference on Pattern Recognition*, 2012.

[4] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty. *Nonlinear Programming: Theory and Algorithms*. Wiley, 1993. 49, 50, 51

[5] J-C. Bazin, Y. Seo, C. Demonceaux, P. Vasseur, K. Ikeuchi, I. Kweon, and M. Pollefeys. Globally optimal line clustering and vanishing point estimation in manhattan world. In *Proc. Conf. Computer Vision and Pattern Recognition*, Providence, Rhode Island, USA, 2012. 44

[6] S. T. Birchfield and A. Subramanya. Microphone array position calibration by basis-point classical multidimensional scaling. *IEEE transactions on Speech and Audio Processing*, 13(5), 2005. 97

[7] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, USA, 1987. 59, 62, 63, 81, 82

[8] T.M. Breuel. Implementation techniques for geometric branch-and-bound matching methods. *Computer Vision and Image Understanding*, 2003. 44

[9] Simon Burgess, Yubin Kuang, and Kalle Åström. Toa sensor network calibration for receiver and transmitter spaces with difference in dimension. In *Proceeding of the 21st European Signal Processing Conference 2013*, 2013. 97, 98, 101, 104

[10] M. Byröd, M. Brown, and K. Åström. Minimal solutions for panoramic stitching with radial distortion. In *bmvc09*, 2009. 36

[11] M. Byröd, K. Josephson, and K. Åström. Fast and stable polynomial equation solving and its application to computer vision. *Int. Journal of Computer Vision*, 2009. 22, 24, 35, 73

[12] 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. 22

[13] 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. 101

[14] T.A. Cass. Polynomial-time geometric matching for object recognition. *Int. Journal of Computer Vision*, 1999. 45

[15] M. Chasles. Question 296. *Nouv. Ann. Math.*, 14(50), 1855. 29

[16] J. C. Chen, R. E. Hudson, and K. Yao. Maximum likelihood source localization and unknown sensor location estimation for wideband signals in the near-field. *IEEE transactions on Signal Processing*, 50, 2002. 97

[17] O. Chum and J. Matas. Optimal randomized ransac. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2008. 44

[18] T. F. Cootes, G. J. Edwards, and Taylor C. J. Active appearance models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001. 79

[19] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Springer, 2007. 18, 20

[20] 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. 97

[21] R.D. Datteri and B.M. Dawant. Estimation and reduction of target registration error. In *MICCAI*, pages 139–146, 2012. 79

[22] S. Doyle, A. Madabhushi, M. Feldman, and J. Tomaszeweski. A boosting cascade for automated detection of prostate cancer from digitized histology. In *Medical Image Computing and Computer-Assisted Intervention*, pages 504–511, 2006. 80

[23] E. Elnahrawy, Xl. Li, and R. Martin. The limits of localization using signal strength. In *SECON-04*, 2004. 97

[24] O. Enqvist and F. Kahl. Robust optimal pose estimation. In *Proc. European Conf. on Computer Vision*, pages 141–153, Marseille, France, 2008. 44

[25] Olof Enqvist, Erik Ask, Fredrik Kahl, and Kalle Åström. Robust fitting for multiple view geometry. In Andrew Fitzgibbon, editor, *Lecture Notes in Computer Science*, volume 7572, pages 738–751. Springer, Heidelberg, 2012.

[26] J.-C. Faugère. A new efficient algorithm for computing gröbner bases without reduction to zero (f5). In *ISSAC '02*, pages 75–83, New York, NY, USA, 2002. ACM Press. 22

[27] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting. *Commun. Assoc. Comp. Mach.*, 24:381–395, 1981. 9, 93

[28] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun. Assoc. Comp. Mach.*, 1981. 44

[29] 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. 29, 98

[30] J.M. Fitzpatrick, J.B. West, and C.R. Maurer. Predicting error in rigid-body point-based registration. 17(5):694–702, 1998. 79

[31] Nikolay D. Gaubitch, W. Bastiaan Kleijn, and Richard Heusdens. Auto-localization in ad-hoc microphone arrays. In *ICASSP*, pages 106–110. IEEE, 2013. 97

[32] R. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 1997. 73

[33] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. Second Edition. 13, 62

[34] B. Horn, H. M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using ortonormal matrices. *Journal of the Optical Society of America A*, 5, 1988. 74

[35] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternion. *Journal of the Optical Society of America A*, 4(4), 1987. 93

[36] A. Jemal, F. Bray, M.M. Center, J. Ferlay, E. Ward, and D. Forman. Global cancer statistics. *CA Cancer J Clin*, 61(2):69–90, 2011. 80

[37] D.S. Johnson and F.P. Preparata. The densest hemisphere problem. *"Theoretical Computer Science"*, 6(1):93–107, 1978. 56

[38] F. Kahl. *Geometry and Critical Configurations of Multiple Views*. PhD thesis, Lund Institute of Technology, Sweden, 2001. 53

[39] F. Kahl and R. Hartley. Multiple view geometry under the L_∞-norm. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2008. 45

[40] Q. Ke and T. Kanade. Quasiconvex optimization for robust geometric reconstruction. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2007. 45

[41] Yubin Kuang, Erik Ask, Simon Burgess, and Kalle Åström. Understanding toa and tdoa network calibration using far field approximation as initial estimate. In *International Conference on Pattern Recognition Applications and Methods*, 2012.

[42] Yubin Kuang and Kalle Åström. Stratified sensor network self-calibration from tdoa measurements. In *21st European Signal Processing Conference 2013*, 2013. 101, 103, 104

[43] Yubin Kuang, Simon Burgess, Anna Torstensson, and Kalle Åström. A complete characterization and solution to the microphone position self-calibration problem. In *The 38th International Conference on Acoustics, Speech, and Signal Processing*, 2013. 97, 98, 101, 104, 105

[44] J.T. Kwak, S.M. Hewitt, S. Sinha, and R. Bhargava. Multimodal microscopy for automated histologic analysis of prostate cancer. *BMC Cancer*, 11(62), 2011. 79, 80

[45] D. Lazard. Resolution des systemes d'equations algebriques. *Theor. Comput. Sci.*, 15:77–110, 1981. 21

[46] K. Lebeda, J. Matas, and O. Chum. Fixing the locally optimized ransac. In *Proc. British Machine Vision Conference*, 2012. 44, 75

[47] H. Li. A practical algorithm for L_∞ triangulation with outliers. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2007. 45

[48] H. Li. Consensus set maximization with guaranteed global optimality for robust geometry estimation. In *Proc. Int. Conf. on Computer Vision*, 2009. 44

[49] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 2004. 8

[50] S. Mittal, S. Anand, and P. Meer. Generalized projection based m-estimator: Theory and applications. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 34(12):2351–2364, 2012. 44

[51] K. Nguyen, B. Sabata, and A.K. Jain. Prostate cancer grading: Gland segmentation and structural features. *Pattern Recognition Letters*, 33(7):951–961, 2012. 80

[52] D. Niculescu and B. Nath. Ad hoc positioning system (aps). In *GLOBECOM-01*, 2001. 97

[53] C. Olsson, O. Enqvist, and F. Kahl. A polynomial-time bound for matching and registration with outliers. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2008. 45

[54] C. Olsson, A. Eriksson, and R. Hartley. Outlier removal using duality. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2010. 45

[55] S. Ourselin, A. Roche, G. Subsol, X. Pennec, and N. Ayache. Reconstructing a 3D structure from serial histological sections. *Image and Vision Computing*, 19:25–31, 2001. 91

[56] 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. 97, 104

[57] J. Sallai, G. Balogh, M. Maroti, and A. Ledeczi. Acoustic ranging in resource-constrained sensor networks. In *eCOTS-04*, 2004. 97

[58] K. Sim and R. Hartley. Removing outliers using the L_∞-norm. In *Proc. Conf. Computer Vision and Pattern Recognition*, 2006. 45

[59] H. Stewénius. *Gröbner Basis Methods for Minimal Problems in Computer Vision*. PhD thesis, Lund University, APR 2005. 97

[60] H. Stewénius, F. Kahl, D. Nistér, and F. Schaffalitzky. A minimal solution for relative pose with unknown focal length. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 789–794, San-Diego, USA, June 2005. Chapter 8 of my thesis. 29

[61] C. Studholme, D.L.G. Hill, and D.J. Hawkes. An overlap invariant entropy measure of 3D medical image alignment. *Pattern Recognition*, 32(1):71–86, 1999. 90

[62] S. Thrun. Affine structure from sound. In *Proceedings of Conference on Neural Information Processing Systems (NIPS)*, Cambridge, MA, 2005. MIT Press. 111, 124

[63] R. Toldo and A. Fusiello. Robust multiple structures estimation with J-linkage. In *Proc. European Conf. on Computer Vision*, Marseille, France, 2008. 44

[64] B. Tordoff and D. W. Murray. Guided sampling and consensus for motion estimation. In *Proc. European Conf. on Computer Vision*, 2002. 44

[65] P. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. *Image and Vision Computing*, 15(8):591–605, 1997. 29

[66] P. Torr and A. Zisserman. Robust computation and parametrization of multiple view relations. In *Proc. 6th Int. Conf. on Computer Vision, Mumbai, India*, pages 727–732, 1998. 29

[67] Wikipedia. www.wikipedia.org. 81

[68] J. Wills, S. Agarwal, and S. Belongie. A feature-based approach for dense segmentation and estimation of large disparity motion. *Int. Journal of Computer Vision*, 68(2):125–143, 2006. 45

[69] J. Yu, A. Eriksson, T.-J. Chin, and D. Suter. An adversarial optimization approach to efficient outlier removal. In *Proc. Int. Conf. on Computer Vision*, 2011. 45

[70] M. Zuliani, C.S. Kenney, and B.S. Manjunath. The multi-RANSAC algorithm and its application to detect planar homographies. In *International Conference on Image Processing*, Genoa, Italy, 2005. 44, 45