THE UNIVERSITY of ADELAIDE

SUB CRUCE LUMEN

DOCTORAL THESIS

New Algorithmic Developments in Maximum Consensus Robust Fitting

Author:
Huu Minh LE

Supervisors:
Assoc. Prof. Tat-Jun CHIN
Prof. David SUTER

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the

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

June 2018

Declaration

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

The author acknowledges that copyright of published works contained within this thesis resides with the copyright holder(s) of those works.

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

Signed: ____________

Date: 5 - 1 - 2018

Abstract

In many computer vision applications, the task of robustly estimating the set of parameters of a geometric model is a fundamental problem. Despite the longstanding research efforts on robust model fitting, there remains significant scope for investigation. For a large number of geometric estimation tasks in computer vision, maximum consensus is the most popular robust fitting criterion. This thesis makes several contributions in the algorithms for consensus maximization.

Randomized hypothesize-and-verify algorithms are arguably the most widely used class of techniques for robust estimation thanks to their simplicity. Though efficient, these randomized heuristic methods do not guarantee finding good maximum consensus estimates. To improve the randomize algorithms, guided sampling approaches have been developed. These methods take advantage of additional domain information, such as descriptor matching scores, to guide the sampling process. Subsets of the data that are more likely to result in good estimates are prioritized for consideration. However, these guided sampling approaches are ineffective when good domain information is not available. This thesis tackles this shortcoming by proposing a new guided sampling algorithm, which is based on the class of LP-type problems and Monte Carlo Tree Search (MCTS). The proposed algorithm relies on a fundamental geometric arrangement of the data to guide the sampling process. Specifically, we take advantage of the underlying tree structure of the maximum consensus problem and apply MCTS to efficiently search the tree. Empirical results show that the new guided sampling strategy outperforms traditional randomized methods.

Consensus maximization also plays a key role in robust point set registration. A special case is the registration of deformable shapes. If the surfaces have the same intrinsic shapes, their deformations can be described accurately by a conformal model. The uniformization theorem allows the shapes to be conformally mapped onto a canonical domain, wherein the shapes can be aligned using a Möbius transformation. The problem of correspondence-free Möbius alignment of two sets of noisy and partially overlapping point sets can be tackled as a maximum consensus problem. Solving for the Möbius transformation can be approached by randomized voting-type methods which offers no guarantee of optimality. Local methods such as Iterative Closest Point can be applied, but with the assumption that a good initialization is given or these techniques may converge to a bad local minima. When a globally optimal solution is required, the literature has so far considered only brute-force search. This thesis contributes a new branch-and-bound algorithm that solves for the globally optimal Möbius transformation much more efficiently.

So far, the consensus maximization problems are approached mainly by randomized algorithms, which are efficient but offer no analytical convergence guarantee. On the other hand, there exist exact algorithms that can solve the problem up to global optimality. The global methods, however, are intractable in general due to the NP-hardness of the consensus maximization.

To fill the gap between the two extremes, this thesis contributes two novel deterministic algorithms to approximately optimize the maximum consensus criterion. The first method is based on non-smooth penalization supported by a Frank-Wolfe-style optimization scheme, and another algorithm is based on Alternating Direction Method of Multipliers (ADMM). Both of the proposed methods are capable of handling the non-linear geometric residuals commonly used in computer vision. As will be demonstrated, our proposed methods consistently outperform other heuristics and approximate methods.

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Associate Professor Tat-Jun Chin and Professor David Suter for their guidance, support, and encouragements during my research. The wisdom that I received from my supervisors is the most precious posession that I am thankful to have throughout the course of my PhD.

My appreciation is also extended to Dr. Anders Eriksson for giving me valuable discussions and inspiring me with his excellent knowledge on mathematics during my visit at Queensland University of Technology.

I would also like to thank Dr. Alvaro Parra Bustos, Mr. Toan Tran and Dr. Thanh-Toan Do for providing me with many resources and giving me insightful comments on my work. My sincere thanks also goes Dr. Trung Pham and Dr. Quoc-Huy Tran for their advise on research methods during the early days of my research career.

My gratitude also goes to The University of Adelaide for awarding me with the The Beacon of Enlightenment scholarship. I also thank the Australian Centre for Robotic Vision (ACRV), the ARC grants (DP160103490) for funding support.

Last but not least, I would like to thank my family for their unconditional support and encouragement during my research.

Contents

List of Figures

Publications

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

- **Huu Le**, Tat-Jun Chin and David Suter: Conformal Surface Alignment With Optimal Mobius Search. In Computer Vision and Pattern Recognition (CVPR) 2016.
- **Huu Le**, Tat-Jun Chin and David Suter: An Exact Penalty Method for Locally Convergent Maximum Consensus. In Computer Vision and Pattern Recognition (CVPR) 2017.
- **Huu Le**, Tat-Jun Chin and David Suter: RATSAC-Random Tree Sampling for Consensus Maximization. In Digital Image Computing: Techniques and Applications (DICTA) 2017 - (Oral presentation) - **Winner of the DSTG award**
- **Huu Le**, Tat-Jun Chin, Anders Eriksson and David Suter: Deterministic Approximate Methods for Maximum Consensus Robust Fitting. Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) - arxiv: https://arxiv.org/abs/1710.10003

To my parents.

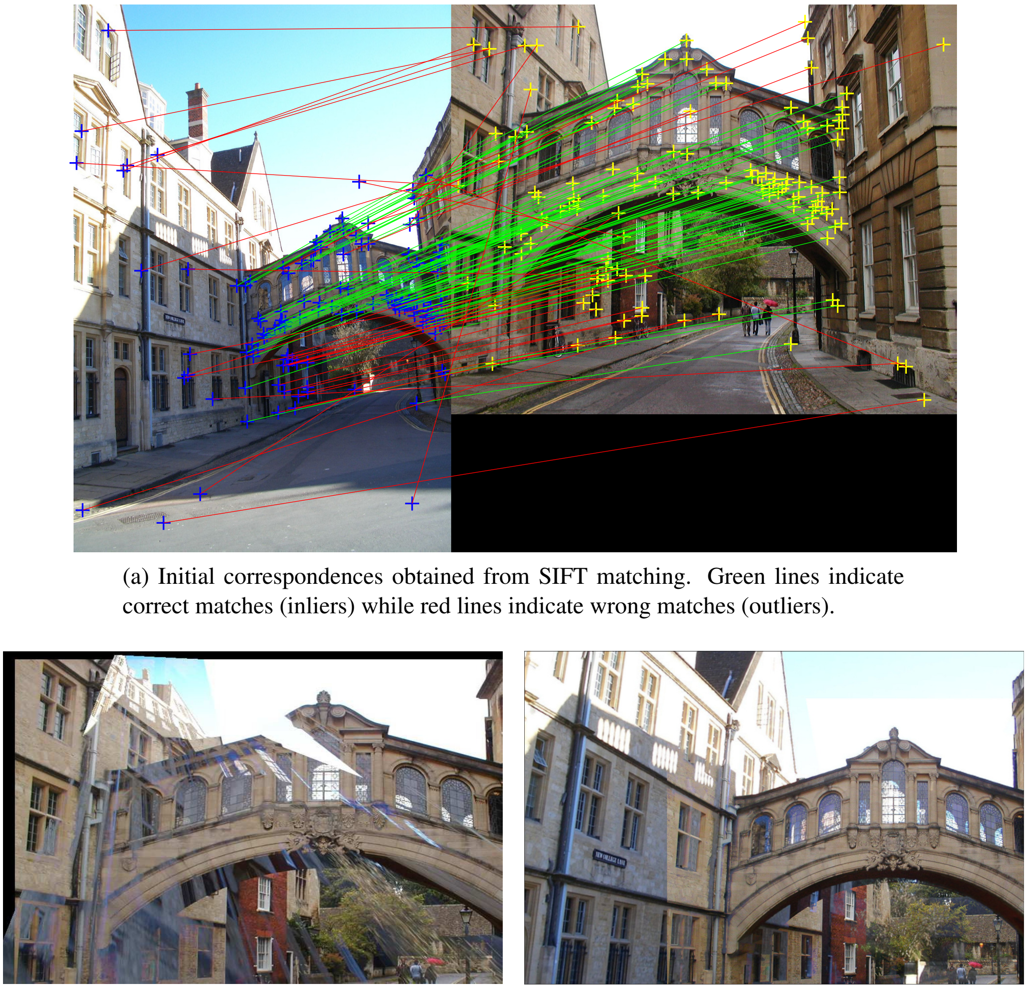
Chapter 1

Introduction

The goal of computer vision research is to build algorithms that enable machines to see and understand the world at the level of a human. This task, however, is immensely difficult due to many factors. Among them, the noise and outliers in the measurements obtained from imperfect data acquisition devices and preprocessing algorithms are major factors that can impede the performance of a computer vision algorithm. Therefore, the task of removing outliers and selecting the most relevant subset of data for further processing is one of the most important research topics in computer vision; this task is the focus of this thesis.

Given a set of measurements (observations), the goal of robust model fitting is to estimate a model that is consistent with as many of the relevant data points – so called the inliers – as possible. If the dataset contains no outliers, traditional least squares estimation is usually sufficient. For outlier-contaminated data, however, a robust estimator must be employed so that the final results are not affected by the outliers. To demonstrate the importance of robust estimators, consider the simple application of two-view image stitching. The input data consists of two overlapping images – see Fig. 1.1. Assuming that the poses of the two cameras differ by only a rotation, the images can be aligned by a homography transformation. As preprocessing, SIFT [64] feature points on each image are extracted and a set of putative matches between the images are produced. Fig. 1.1a plots this initial set of correspondences. As can be observed, SIFT matching invariably makes mistakes; the green lines indicate correct matches while wrong pairs are plotted in red. The incorrect matches behave as outliers to the estimation of the homography. Fig. 1.1b shows the stitching result using the least squares estimate of the homography, while Fig. 1.1c shows the stitching result obtained by employing a robust estimator (RANSAC [31]). Observe that that the estimate returned by least squares is much worse than the RANSAC estimate.

Another example problem that requires the use of robust model fitting is three-dimensional (3D) shape registration. In such applications, the input data is commonly represented in the



(a) Initial correspondences obtained from SIFT matching. Green lines indicate correct matches (inliers) while red lines indicate wrong matches (outliers).

(b) Stitching result obtained by applying least squares estimation on the initial correspondences.

(c) Stitching result obtained by using a robust estimator (RANSAC).

FIGURE 1.1: Example of two-view image stitching.

form of 3D point clouds, which are usually acquired from depth sensors or a 3D reconstruction algorithm. A preprocessing algorithm may be applied to extract the keypoints for each input shape to reduce the size of the point clouds. The goal of the point cloud registration problem is to estimate a transformation that best aligns the two point sets. In the case of rigid shapes, the transformation consists of a rotation matrix and a translation vector. For non-rigid shapes, the transformation could be a conformal mapping. Unlike the homography estimation example discussed above, putative correspondences are not normally used in 3D point cloud registration, since 3D keypoint matching methods are much less accurate. The transformation, therefore, needs to be jointly estimated with the set of correspondences. In practice, the input shapes may only partially overlap. Therefore, the input data contains outliers and a robust estimator needs to be used. An example of rigid 3D point cloud registration is shown in Fig. 1.2 and Fig. 1.3 shows the registration of two faces, which are examples of non-rigid shapes.

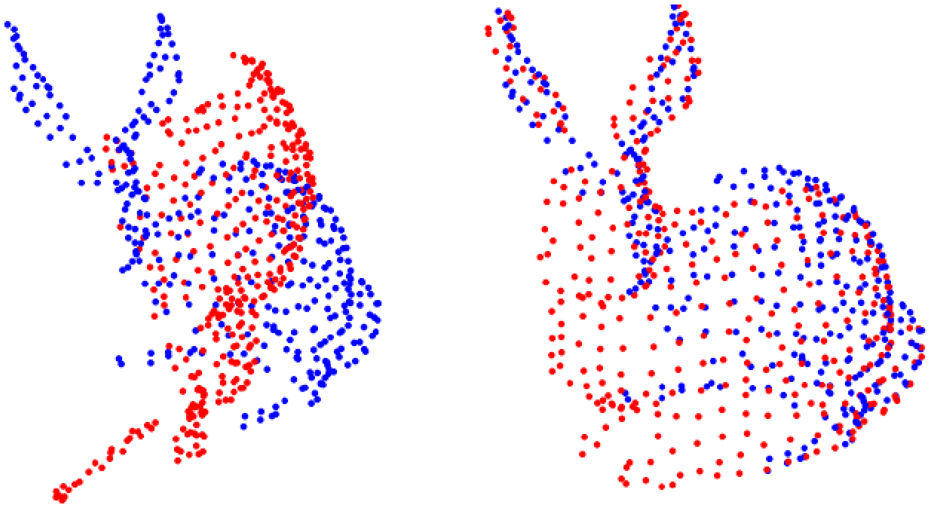


FIGURE 1.2: Registration of two 3D rigid shapes. Left: Two partially overlapping point clouds with arbitrary poses. Right: Point clouds aligned using a rigid transformation estimated by a robust estimation algorithm. Note that the putative correspondence are not available in advance. Points that have no corresponding matches are the outliers in this case.

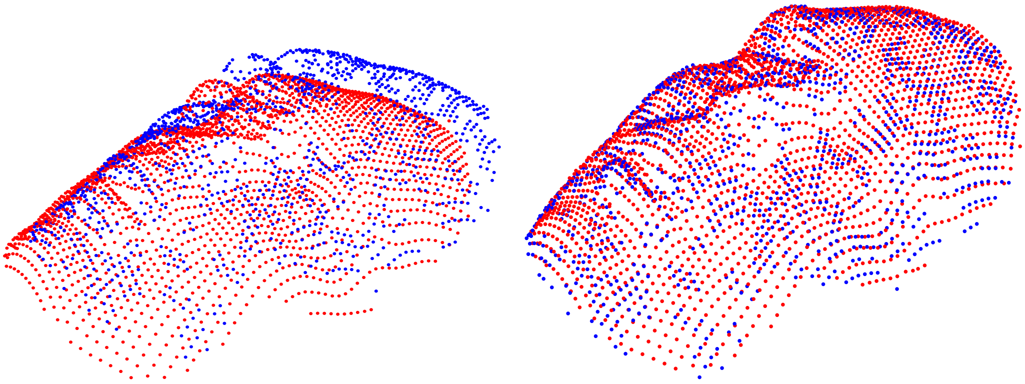


FIGURE 1.3: Example of non-rigid alignment of two faces. Left: Initial poses of the two input faces. Right: Registration result using a robust estimation algorithm. Similar to the rigid case, the putative correspondences are not available beforehand. Note that due to non-overlapping, not all points have corresponding matches, which generates outliers for the data.

1.1 Maximum consensus

In computer vision, consensus maximization is a robust fitting framework that has been used extensively. Given a set of measurements, the maximum consensus approach aims to estimate a model $\boldsymbol{\theta}$ that is consistent with as many of the measurements as possible, where consistency is up to a pre-defined error threshold. Mathematically, this can be summarized by the following optimization problem

$$\begin{aligned} \max_{\boldsymbol{\theta}, \mathcal{I} \in \mathcal{P}(N)} \quad & |\mathcal{I}| \\ \text{subject to} \quad & r_i(\boldsymbol{\theta}) \leq \epsilon \;\; \forall i \in \mathcal{I}, \end{aligned} \tag{1.1}$$

where $r_i(\boldsymbol{\theta})$ is the residual function for the i-th observation with respect to the model estimate $\boldsymbol{\theta}$, N is the number of measurements and $\mathcal{P}(N)$ is the powerset (the set of all subsets) of the measurement index set $\{1, 2, \dots, N\}$. The parameter ϵ is the pre-defined inlier threshold which decides whether a measurement is inlying with respect to the estimate $\boldsymbol{\theta}$. The set of measurements, the residual functions $r_i(\boldsymbol{\theta})$ and the inlier threshold ϵ depend on the application. In the following, specific instances of problem (1.1) are listed.

Robust Linear Fitting: Given a set of measurements $\{\mathbf{a}_i, b_i\}_{i=1}^{N}$ where $\mathbf{a}_i \in \mathbb{R}^d$ and $b_i \in \mathbb{R}$, the model that needs to be estimated is a hyperplane $\boldsymbol{\theta} \in \mathbb{R}^d$. In the context of (1.1), the robust linear fitting problem can be written as

$$\begin{aligned} \max_{\boldsymbol{\theta}, \mathcal{I} \in \mathcal{P}(N)} \quad & |\mathcal{I}| \\ \text{subject to} \quad & |\mathbf{a}_i^T \boldsymbol{\theta} - b_i| \leq \epsilon \;\; \forall i \in \mathcal{I}. \end{aligned} \tag{1.2}$$

Consider an example of a robust line fitting, which is shown in Fig. 1.4. Given a set of points $\{x_i, y_i\}_{i=1}^{N}$ on the two-dimensional plane, the goal is to estimate the line represented by the equation $y = mx + c$ (m and c are the parameters) that agrees with as many of the points as possible. With $\mathbf{a}_i = [x_i \;\, 1]^T$, $b_i = y_i$ and $\boldsymbol{\theta} = [m \;\, c]^T$, the robust line fitting problem can be put in the form of (1.2). Fig. 1.4 plots the fitting results obtained from a robust fitting algorithm (RANSAC) and the least squares estimator. The blue points are the inliers and the red points are the outliers with respect to the RANSAC solution. Observe that the outliers have greatly biased the least squares estimate, while RANSAC was able to return a solution that better describes the underlying structure. In computer vision, robust linear regression plays a major role because it is often employed in many estimation tasks.

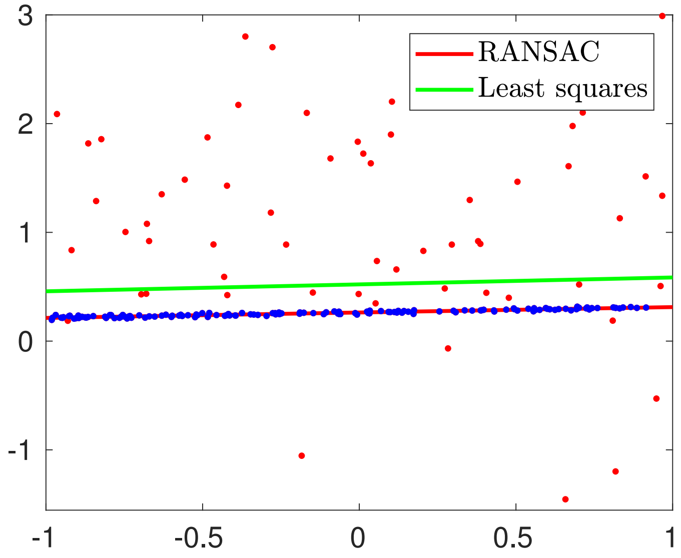


FIGURE 1.4: Example of a 2D line fitting problem. Blue points are the inliers and red points are the outliers. The estimate returned by RANSAC was not affected by the outliers, while the least squares estimate was biased by the outliers and was not able to fit the underlying structure.

Robust Homography Estimation: From the set of correspondences $\{\mathbf{u}_i, \mathbf{v}_i\}_{i=1}^N$ (obtained from automatic feature dectection and matching – see Fig. 1.1) the task is to estimate a 3×3 homography matrix $\boldsymbol{\theta}$. With the residual function defined to be the transfer error, the problem can be written as consensus maximization as follows

$$\begin{aligned} \max_{\boldsymbol{\theta}, \mathcal{I} \in \mathcal{P}(N)} \quad & |\mathcal{I}|, \\ \text{subject to} \quad & \left\| \mathbf{v}_i - \frac{\boldsymbol{\theta}_{1:2}\bar{\mathbf{u}}_i}{\boldsymbol{\theta}_3\bar{\mathbf{u}}_i} \right\| \leq \epsilon \ \ \forall i \in \mathcal{I}, \end{aligned} \tag{1.3}$$

where $\boldsymbol{\theta}_{1:2}$ and $\boldsymbol{\theta}_3$ are respectively the first two rows and the third row of the homography matrix, $\bar{\mathbf{u}}_i = [\mathbf{u}^T \ 1]^T$ and $\|.\|$ is the Euclidean norm in $\mathbb{R}^2$. The residual function in (1.3) measures the Euclidean distance between a point in the first view and its correspondence in the second view after the transformation $\boldsymbol{\theta}$ is applied, while the inlier threshold ϵ specifies the acceptance error (in pixels) for a pair of corresponding feature to be counted as inlier. The task of estimating a homography between two or multiple views is a curcial step in the larger pipeline of many computer vision applications such as image stitching, 3D reconstruction, and structure from motion.

Robust point set alignment: Given two point sets $\mathbf{M} = \{\mathbf{m}_i\}_{i=1}^{N_1}$ and $\mathbf{B} = \{\mathbf{b}_j\}_{j=1}^{N_2}$, where $\mathbf{m}_i \in \mathbb{R}^3$ and $\mathbf{b}_j \in \mathbb{R}^3$ are three-dimensional points, the task is to find a mapping $\Phi : \mathbb{R}^3 \to \mathbb{R}^3$ that best aligns the two point sets $\mathbf{M}$ and $\mathbf{B}$. In the context of consensus maximization, this problem can be formulated as

$$\begin{aligned} \max_{\Phi, \mathcal{I} \in \mathcal{P}(N_1)} \quad & |\mathcal{I}| \\ \text{subject to} \quad & \exists \mathbf{b}_j \in \mathbf{B}, \|\Phi(\mathbf{m}_i) - \mathbf{b}_j\| \leq \epsilon \ \ \forall i \in \mathcal{I}. \end{aligned} \tag{1.4}$$

where $\|.\|$ is the Euclidean norm. Note that in this problem, $\mathbf{M}$ and $\mathbf{B}$ are two separate sets and their putative correspondences are not available beforehand. Therefore, in order for a point $\mathbf{m}_i$ to be counted as inlier, there must exist a point $\mathbf{b}_j \in \mathbf{B}$ such that the distance between $\Phi(\mathbf{m}_i)$ and $\mathbf{b}_j$ (measured by the residual function $\|\Phi(\mathbf{m}_i) - \mathbf{b}_j\|$) is not greater than the error threshold ϵ. Typically, Φ is chosen from a specific class of transformations suitable for the problem at hand.

In the case of rigid transformation, Φ consists of a rotation matrix $\mathbf{R} \in SO(3)$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$

$$\Phi(\mathbf{m}_i|\mathbf{R}, \mathbf{t}) = \mathbf{R}\mathbf{m}_i + \mathbf{t}. \tag{1.5}$$

If the shape undergoes a non-rigid transformation, the function Φ must be chosen to take into account the deformations of the shape. For many biological objects, the conformal mapping, which transfers the shapes onto some canonical domains such as the hyperbolic discs or spheres, is commonly used as a pre-processing technique. In the special case of isometric surfaces, their conformal maps can be related by a Möbius transformation, which can be defined as

$$\Phi_M(\mathbf{m}_i|\mathbf{z}, \theta) = e^{i\theta}\frac{\mathbf{m}_i - \mathbf{z}}{1 - \mathbf{m}_i\bar{\mathbf{z}}}, \tag{1.6}$$

where $\mathbf{m}_i$ and $\mathbf{z}$ are complex numbers corresponding to points on the hyperbolic disc $\mathbb{D}$, $\bar{\mathbf{z}}$ denotes the complex conjugate of $\mathbf{z}$ and $\theta \in [-\pi; \pi]$. By substituting Φ_M for Φ in (1.4), the problem of robust non-rigid shape alignment can be casted as an instance of consensus maximization which can be performed on the cannonical domain. Fig. 1.5 shows an example of a face alignment problem using conformal mapping, where the faces are mapped onto the hyperbolic discs and aligned by estimating the Möbius transformation Φ_M. More details on shape registration using conformal mapping are discussed in Chapter 4.

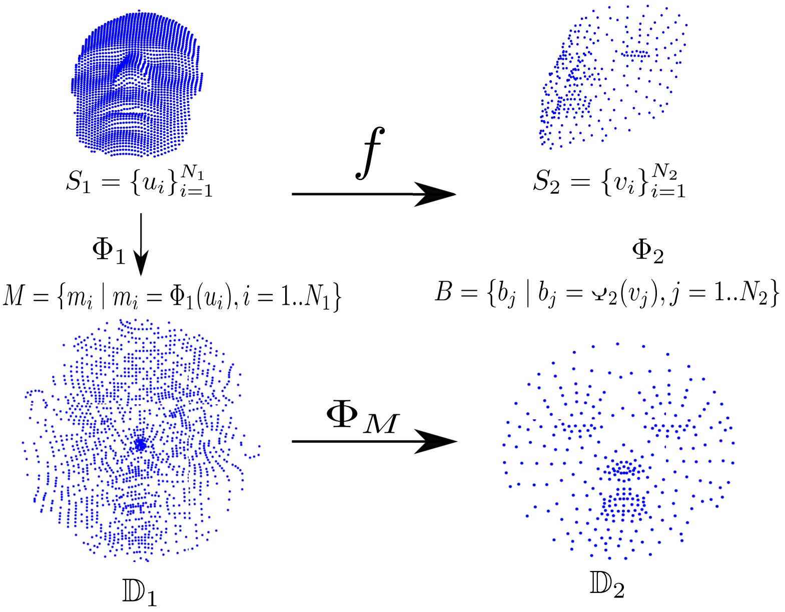

$S_1 = \{u_i\}_{i=1}^{N_1}$
f
$S_2 = \{v_i\}_{i=1}^{N_2}$
Φ_1
Φ_2
$M = \{m_i \mid m_i = \Phi_1(u_i), i = 1..N_1\}$
$B = \{b_j \mid b_j = \Upsilon_2(v_j), j = 1..N_2\}$
Φ_M
$\mathbb{D}_1$
$\mathbb{D}_2$

FIGURE 1.5: Illustration non-rigid shape alignment using conformal mapping. Two partially overlapping input shapes S_1 and S_2 are mapped onto hyperbolic discs using the two mapping functions Φ_1 and Φ_2 respectively, before robustly estimating the Möbius transformation Φ_M.

1.2 Why is maximum consensus hard?

The consensus maximization problem is computationally challenging due to the combinatorial nature of the problem. It has been established that this problem is NP-hard [19]. To solve the problem exactly, some form of exhaustive searching is inevitably required. Despite extensive efforts devoted to solving this problem globally, obtaining the globally optimal maximum consensus solution is not fast in most of the real life applications due to the large number of the measurements and high dimensionality. In practice, this problem is approached mostly by the class of randomized hypothesize-and-verify methods due to their simplicity and their ability to provide usually acceptable results. The following section briefly reviews the popular approaches, both approximate and exact, for the consensus maximization and describes their strengths and weaknesses.

1.3 Previous works

1.3.1 Approximate methods

1.3.1.1 Random Sample Consensus (RANSAC) and its variants

RANSAC [31] (**RAN**dom **SA**mpling **C**onsensus) is arguably one of the most widely used algorithms for consensus maximization. This method works by randomly sampling and fitting the model onto the minimal subsets, where a minimal subset is defined as the subset of data that contains the minimum of number data points required to estimate a model. For each fitted model, the number of inliers of the hypothesis is recorded. The sampling and verfication process is repeated until the probabilistic stopping criterion is satisfied. Finally, the model with the largest consensus size is returned. The idea behind RANSAC is that if one samples a sufficiently large number of minimal subsets, it is likely that a minimal subset with all inliers will be chosen, and the hypothesis of that subset provides a good estimate.

Despite its simplicity, RANSAC performs quite well in many applications. The popularity of RANSAC had inspired multiple variants. Most variants aim to achieve a higher quality solution and/or lower execution time. Among the RANSAC variants, LO-RANSAC (**L**ocally **O**ptimized RANSAC) [23, 56] is a notable example that aims to locally optimize the solution of RANSAC. Based on the observation that an uncontaminated minimal subset produces a hypotheis model that is close to the optimal solution, LO-RANSAC samples larger-than-minimal subsets, where the samples are drawn from the current best-so-far inlier set. This procedure is embedded into RANSAC's iterations and is triggered when RANSAC solution is updated. The method has been shown to achive good improvements over the original RANSAC algorithm.

Several RANSAC variants [22][84] use additional domain information to guide the sampling process, particularly in two-view geometry problems where the data are correspondences associated with matching scores. Correspondences with higher scores are favored for selection, under the assumption that they are more likely to lead to good hypotheses. Empirical results demonstrated that these guided sampling strategies achieved substantial improvements over standard RANSAC.

There exist fundamental shortcomings in the random sampling. Primarily, its randomized nature offers no guarantee on finding good estimates. Moreover, different runs may give different results. LO-RANSAC also suffers the same weakness as the inner sampling routine also employ randomized trial-and-error procedure instead of a deterministic search to improve the estimate. For guided sampling methods that require additional domain knowledge, they are ineffective on problem settings where useful/accurate domain knowledge is not available.

1.3.1.2 Methods employing convex optimization

Besides the class of randomized methods, approximate methods such as ℓ_1 approximation [74] have been developed to tackle the consensus maximization problem. This algorithm works by first reformulating the consensus maximization problem into an outlier minimization problem, where the outlier count is represented by the ℓ_0 norm of the slack variables (more details are discussed in Sec 2.3.2.1). To make the problem amenable to convex optimization, the objective function is relaxed by minimizing the ℓ_1 norm of the slack variables instead. In the context of linear regression, this relaxation results in an instance of linear programming (LP), which can be solved efficiently using mature LP algorithms.

Based on the observation that the maximum residual of all the data points that belong to any consensus set $\mathcal{I}$ is less than or equal to the inlier threshold ϵ, an algorithm based on ℓ_∞ minimization can be constructed to solve the maximum consensus problem approximately [79]. The ℓ_∞ minimization estimates the model that minimizes the maximum residuals (more details in Sec 2.1.2). The idea behind the outlier removal algorithm is to recursively solve the ℓ_∞ fitting problem and remove the data whose residuals equal the minimax value. The algorithm is stopped when the minimax value is not greater than the inlier threshold ϵ. Sec. 2.3.2.2 provides deeper discussion on this algorithm.

In general, the approximate methods based on convex optimization work well in moderately difficult instances, i.e., low outlier rate and/or the outliers are uniformly distributed. In the later chapters, it will be demonstrated that these methods may fail on many practical datasets.

1.3.2 Exact methods

1.3.2.1 Mixed Integer Program (MIP)

The maximum consensus problem is a special case of maximum feasible subsystem (MaxFS) problem [20]. Therefore, consensus maximization can be solved up to global optimality by converting it into an instance of mixed integer program (MIP). If the constraints are linear, it becomes a special case of mixed integer linear program (MILP) – more details are discussed in Section 2.4.1. The MIP re-formulation enables consensus maximization to be tackled by popular off-the-shelf solvers (Gurobi or IBM's CPLEX, etc.). In practice, as the problem is also intractable in general, solving MIPs exactly is computationally expensive.

Most state-of-the-art MIP solvers employ branch-and-bound (BnB), which is a global optimization technique that are frequently used to solve non-convex optimization problems [45], to search for the optimal solution. In computer vision, it has also been used extensively for the robust geometric matching problems [34, 75, 33]. BnB works by dividing the search space into smaller subdomains. For each subdomain, a bounding function is evaluated to decide whether that particular subdomain is pruned or further divided. The effectiveness of a BnB algorithm relies on the tighness of the bounding function. Branch-and-bound has exponential complexity in the worst case, thus renders it impractical for many real-time applications. More details on the mechanism behind BnB are discussed in Sec. 2.4.3.

1.3.2.2 Tree search

An interesting property of the maximum consensus problem is its underlying tree structure. By formulating consensus maximization as an instance of tree search, many works have proposed different tree traversal strategies. The standard bread-first-search approach was investigated in [67]. This strategy was then improved by introducing a heuristics and employ the A* search strategy [18], which significantly speed up the time needed to reach the optimal solution. However, these algorithms are still far from practical for problems with large input as it may take exceptionally long time for the algorithm to finish, especially for problems in high dimensionality.

1.4 Motivations and contributions

Despite longstanding research efforts in consensus maximization, there remain many avenues for research. The specific contributions of this thesis are summarized as follows:

- The first contribution of this thesis is a novel tree sampling algorithm to solve the maximum consensus problem. State-of-the-art guided sampling techniques require domain information such as keypoint matching scores, thus they cannot be easily extended. Taking advantage of the fact that the maximum consensus problem can be formulated as an instance of tree search, a randomized algorithm based on the Monte Carlo Tree Search (MTCS) is proposed. This method is a guided sampling strategy that makes use of the underlying geometric structure to accelerate the search. It is emperically demonstrated that the method outperforms traditional RANSAC and its variants.
- The second contribution of this thesis is a novel globally optimal algorithm for estimating Möbius transformations to robustly align surfaces with topological discs. As discussed previously, this enables the robust estimation of conformal maps. Current algorithms

for estimating Möbius transformations often cannot provide satisfactory alignment or are computationally too costly. Unlike previous methods, the proposed algorithm deterministically calculates the best transformation, without requiring good initializations. Further, the proposed algorithm is also much faster than previous techniques in practice.

- The third contribution of this thesis is to propose deterministic approximate algorithms for maximum consensus estimation. The proposed algorithms represent a new class of maximum consensus solvers, that sit between randomized and globally optimal approaches. One of the proposed algorithm is based on the non-smooth penalty method with a Frank-Wolfe style optimization scheme, the other is based on the Alternating Direction Method of Multipliers (ADMM). Both algorithms solve convex subproblems to efficiently compute the update step. It can be demonstrated that the algorithms can greatly improve a rough initial estimate, such as those obtained using least squares or a randomized maximum consensus heuristic.

1.5 Thesis outline

The rest of this thesis is structured as follows:

- Chapter 2 summaries the fundamental theory behind the maximum consensus problem, including the special case of robust non-rigid correspondence problem. Chapter 2 also reivews some of the previous works and discusses the theoretical and practical gaps that will be filled by the works developed in the following chapters of this thesis.
- Chapter 3 describes the random tree sampling algorithm (RATSAC) for maximum consensus.
- Chapter 4 describes the novel algorithm for optimal Möbius alignment.
- Chapter 5 and Chapter 6 describes the deterministics local refinement algorithms.
- Chapter 7 provides concluding remarks and discusses future work.

Chapter 2

Model fitting - the background

This chapter discusses the background of robust model fitting and the maximum consensus problem, and surveys in detail a number of commonly used algorithms and optimization techniques. The outline of this chapter is as follows:

- Section 2.1 discusses non-robust estimation, including least squares estimation, ℓ_∞ estimation, and the algorithms to solve these problems.
- Section 2.2 provides an overview of robust model fitting, with particular attention on the maximum consensus problem, which is the target problem of this thesis.
- The remaining sections review existing methods to solve the maximum consensus problem.

2.1 Non-robust estimation

2.1.1 Least squares estimation

As the name “least squares” suggests, in least squares estimation problem, the parameters are estimated by minimizing the sum of the squared residuals. Least squares estimation is usually sufficient if the dataset contains no outliers. If there are outliers, however, the least squares estimate will be heavily biased – see Fig. 1.4. Mathematically speaking, the model $\boldsymbol{\theta}$ is obtained by solving

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} r_i^2(\boldsymbol{\theta}), \tag{2.1}$$

where N is the number of data points and $r_i(\boldsymbol{\theta})$ is the residual of discrepancy between the $i-$th measurement and its expected values with respect to the model parameter $\boldsymbol{\theta}$.

2.1.1.1 Linear least squares

When the observations can be expressed linearly in terms of the parameters $\boldsymbol{\theta} \in \mathbb{R}^d$, each residual function $r_i(\boldsymbol{\theta})$ is a linear function of the form

$$r_i(\boldsymbol{\theta}) = \mathbf{x}_i^T \boldsymbol{\theta} - y_i, \tag{2.2}$$

where the set $\{\mathbf{x}_i, y_i\}_{i=1}^N$ (with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$) contains the measurements. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^T$, $\mathbf{Y} = [y_1, y_2, \ldots, y_n]^T$ and denote by F the objective function of (2.1), i.e., $F = \sum_{i=1}^N r_i^2(\boldsymbol{\theta})$. With linear residual (2.2), problem (2.1) can be expressed in matrix form as

$$\min_{\boldsymbol{\theta}} \|\mathbf{X}\boldsymbol{\theta} - \mathbf{Y}\|^2. \tag{2.3}$$

As (2.3) is a convex quadratic program, its global solution $\boldsymbol{\theta}^*$ of must satisfy the first-order optimality condition:

$$\frac{\nabla F}{\nabla \boldsymbol{\theta}}(\boldsymbol{\theta}^*) = \mathbf{0}, \tag{2.4}$$

which is equivalent to

$$2\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} - 2\mathbf{X}^T\mathbf{Y} = \mathbf{0}. \tag{2.5}$$

From (2.5), the solution $\boldsymbol{\theta}^*$ of (2.3) can be computed in closed form as

$$\boldsymbol{\theta}^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}. \tag{2.6}$$

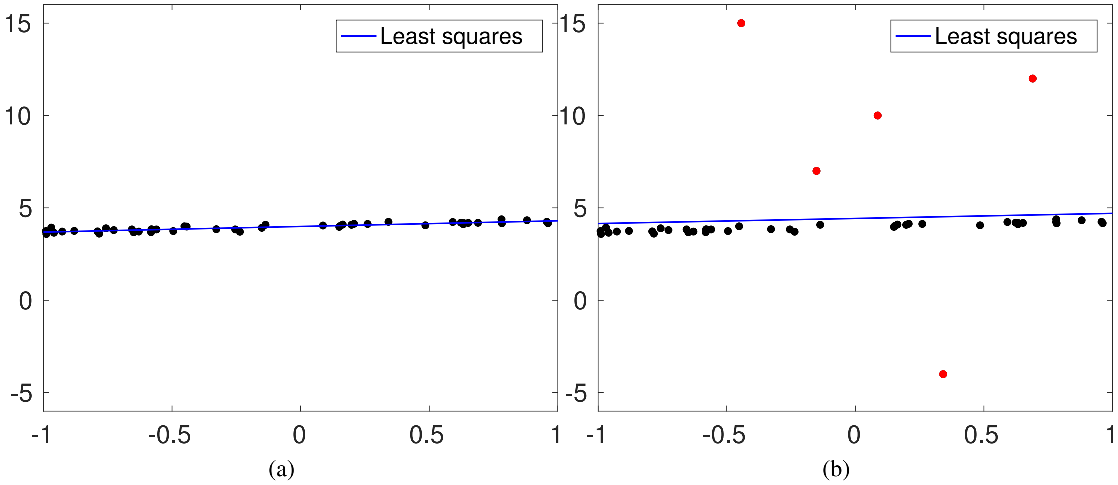
Fig. 2.1 shows the example of linear least squares on a line fitting problem for datasets with and without outliers.

2.1.1.2 Nonlinear least squares

On the other hand, if the residual functions are not linear, the problem becomes *nonlinear least squares*. Usually, the associated optimization problem becomes non-convex, and solving non-linear least squares globally is intractable in general. In practice, non-linear least squares are tackled by non-linear optimization approaches such as Gauss-Newton or Levenberg-Marquardt. These methods, however, require good initializations to prevent them from converging to bad local minima.

2.1.1.3 Iterative Closest Point (ICP)

For many geometric matching problems problems where putative correspondences are not available in advance, for example, the shape correspondence problems discussed above, iterative



(a)

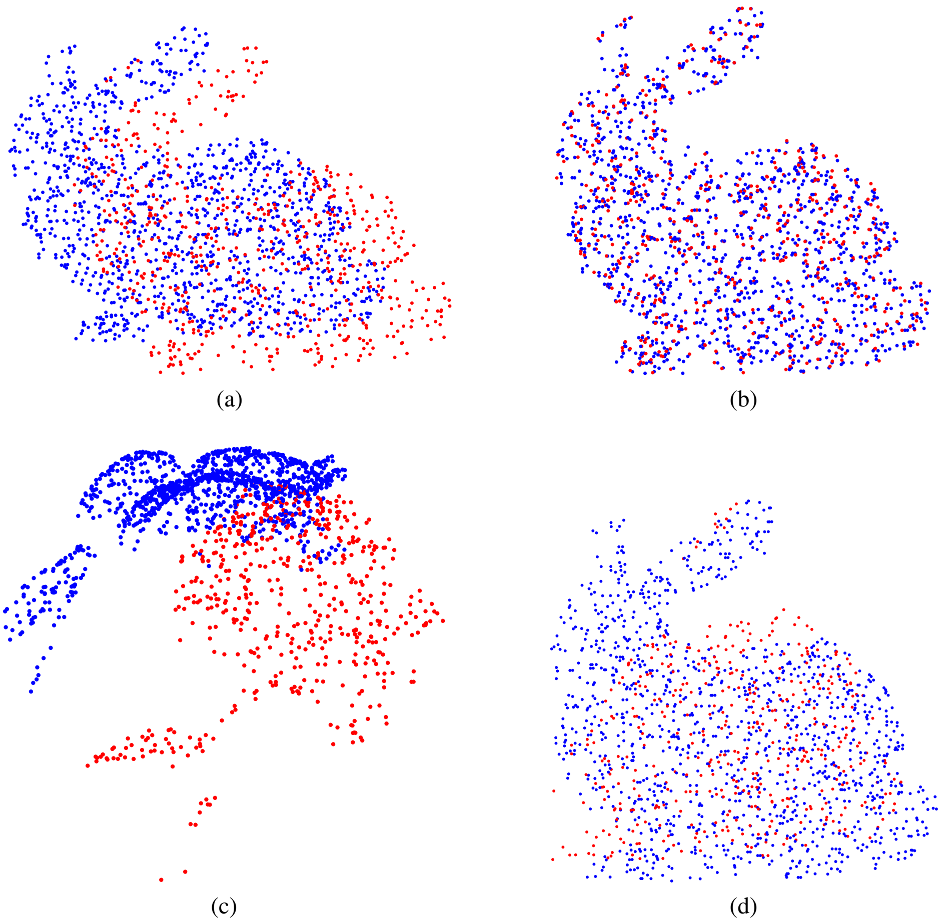
(b)

FIGURE 2.1: Example of linear least squares for a line fitting problem. (a) Least squares fit for a set of 2D points without outliers. (b) When five points are corrupted and become the outliers, the least squares estimate is biased by these outliers (plotted in red) and can no longer represent the underlying structure.

closest points (ICP) is frequently employed to jointly estimate the transformation and the correspondences. At its core, ICP can be viewed as solving a least squares problem since it minimizes the sum of the squared Euclidean distances between matching points

$$\min_{\Phi} \sum_{i=1}^{N_1} \|\Phi(\mathbf{m}_i) - \mathbf{b}_j\|^2, \tag{2.7}$$

where Φ is the transformation that needs to be estimated, $\mathbf{m}_i$ is a point on the source surface and $\mathbf{b}_j$ is the point on the target surface that is closest to the transformed $\mathbf{m}_i$. ICP works by alternating between two steps: point-wise correspondence assignment and geometric transformation estimation. At each iteration, every single point on the source surface is associated with its nearest neighbor on the target surface to form a pair of correspondence. Then, a transformation is estimated by minimizing the sum of squares of the Euclidean distances between the correspondences. The process is repeated until the algorithm converges, i.e., no more improvements can be made. It is a well-known fact that ICP requires good initialization for it to work properly as the method may easily converge to a bad alignment with a poor starting point. Moreover, as a least squares approach, ICP is easily biased by outliers. Here, outliers are points that do not have correct match due to non-overlapping. Fig. 2.2 shows examples of registering two point sets using ICP. Observe that with good initialization (Fig. 2.2a), ICP provides good alignment (Fig. 2.2b), while with bad initialization (Fig. 2.2c), ICP converges to a poor solution (Fig. 2.2d).



(a)

(b)

(c)

(d)

FIGURE 2.2: Example of ICP for two partially overlapping point clouds (bunny). Top row: Point clouds differ by a small transformation (a), ICP can provide relatively good alignment (b). Bottom row: Point clouds differ by a large transformation (c), ICP converged to a poor alignment (d).

2.1.2 ℓ_∞ estimation

Another popoular non-robust estimation approach is ℓ_∞ estimation, i.e., to minimize the largest residuals. Mathematically, this is the optimization problem:

$$\min_{\boldsymbol{\theta}} \max_{i} r_i(\boldsymbol{\theta}), \tag{2.8}$$

The problem (2.8) is also referred to as the "minimax" problem. The appeal of ℓ_∞ optimization lies in the fact that it pocesses one single minimum, which is also the global minimum as long as $r_i(\boldsymbol{\theta})$ is quasi-convex. This is illustrated in Fig 2.3 for a sample problem in one dimension, where $r_i(\boldsymbol{\theta})$ are the quasiconvex functions.

In computer vision, the advantage of ℓ_∞ estimation makes it more favorable than the traditional least squares method since a variety of geometric estimation problems have quasiconvex residuals. There have been much efforts in developing globally optimal algorithms for the ℓ_∞ problem [41, 49, 50, 60].

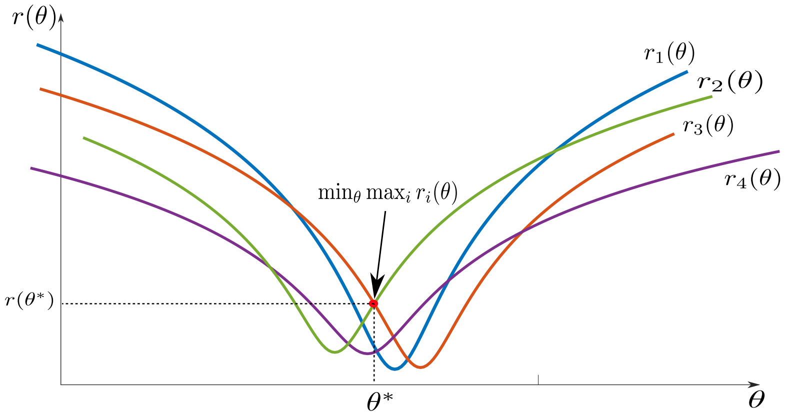


FIGURE 2.3: Example of a ℓ_∞ estimation problem for four quasiconvex residuals in one dimension. θ^* (plotted in red) is the only local minimum, which is also the global solution of the problem.

Consider the equivalent formulation for problem 2.8

$$\begin{aligned} \min_{\boldsymbol{\theta},\gamma} \quad & \gamma, \\ \text{subject to} \quad & r_i(\boldsymbol{\theta}) \le \gamma, \quad \forall i = 1 \dots N. \end{aligned} \tag{2.9}$$

Intuitively, by solving (2.9), one obtains the minimum $\gamma \in \mathbb{R}$ such that all the residuals are not greater than γ, which means the maximum residual is minimized.

Bisection is a simple approach to solve the ℓ_∞ estimation problem. Based on formulation (2.9), one can successively bisect (halve) γ to find the smallest γ such that all the constraints are satisfied. For each candidate $\hat{\gamma}$, the algorithm requires solving the feasibility problem

$$\begin{aligned} \text{Find} \quad & \boldsymbol{\theta} \\ \text{subject to} \quad & r_i(\boldsymbol{\theta}) \le \hat{\gamma} \quad \forall i \end{aligned} \tag{2.10}$$

If the residual functions are quasiconvex, solving the feasibility problem ammounts to solving a convex feasibility problem, which can be done efficiently by several solvers.

Although the ℓ_∞ estimation problem can be solved efficiently up to global optimality, like least squares, it is still heavily affected by the outliers. Fig.2.4 illustrates a line fitting problem with ℓ_∞ estimation and how the result is affected by the outliers.

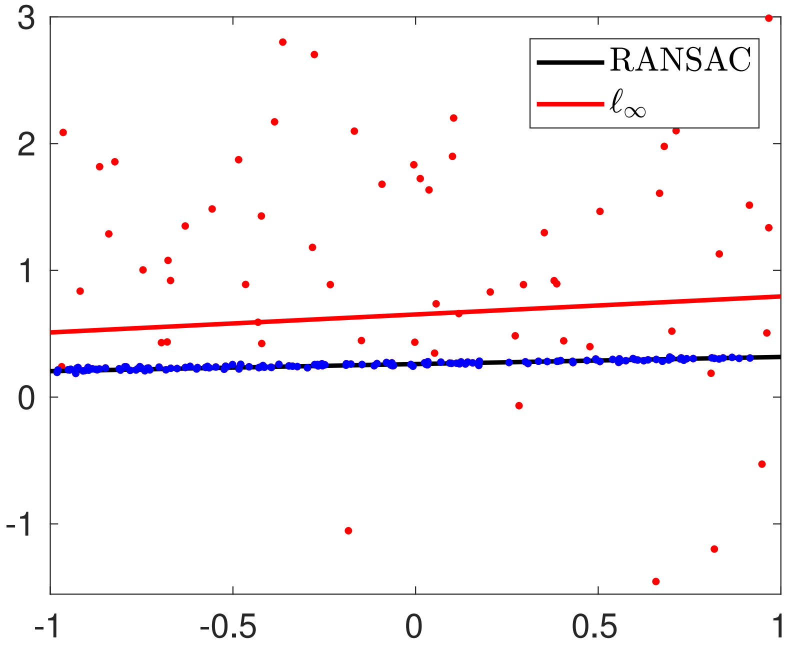


FIGURE 2.4: Plot of ℓ_∞ estimate versus a robust estimate (RANSAC) for a robust line fitting problem. Blue points are the inliers and red points are the outliers with respect to the RANSAC estimate. Note that similar to the least squares estimate, ℓ_∞ is also serverly biased by the outliers.

2.2 Robust model fitting techniques

2.2.1 M-estimation

M-estimation [46] is an established method in statistics which was later adopted for the task of robust estimation in computer vision. Instead of minimizing the sum of squared residuals, the M-estimate is obtained by minimizing the sum of a set of ρ functions defined over the residuals

$$\min_{\boldsymbol{\theta}} \sum_i \rho(r_i(\boldsymbol{\theta})). \tag{2.11}$$

M-estimators generalize maximum likelihood estimators (including least squares estimator). Note that when $\rho(x) = x^2$, (2.11) is identical to (2.1) and the problem becomes least squares estimation. To be outlier-robust, the ρ function should have certain characteristics [5]. Among them, the class of *redescending* M-estimator has a high degree of robustness. The following are examples of some robust ρ functions

- ℓ_1

$$\rho(x) = |x| \tag{2.12}$$

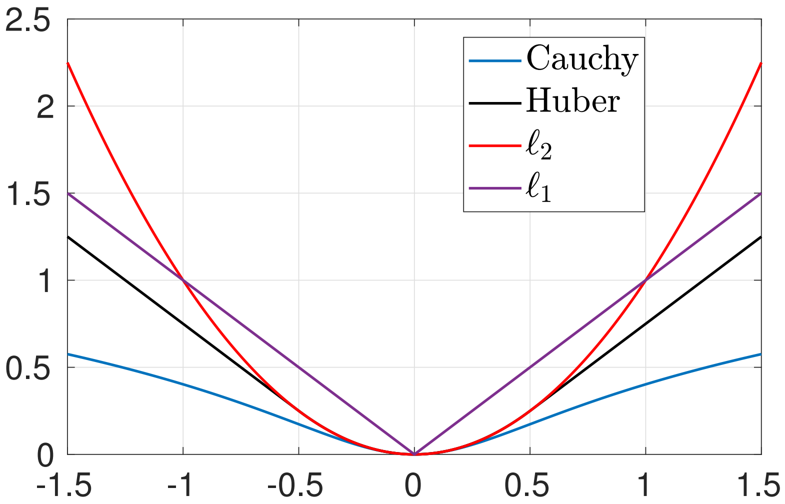


FIGURE 2.5: Plot of different robust loss functions for M-estimator.

- Huber

$$\rho(x) = \begin{cases} x^2 & \text{for } \ |x| < b \\ 2b|x| - b^2 & \text{otherwise.} \end{cases} \tag{2.13}$$

- Cauchy

$$\rho(x) = b^2 log(1 + x^2/b^2) \tag{2.14}$$

Refer to Fig. 2.5 for the plots of the ρ functions listed above. Note that some of the functions require additional parameters (e.g. b) that define the shape of the functions. Depending on the choice of ρ and the characteristics of the residuals, solving (2.11) can be difficult due to its non-convexity. Therefore, M-estimators can be computed using standard optimization algorithms, assuming that good initializations are provided.

Iteratively Reweighted Least Squares (IRLS) is commonly used for M-estimation. IRLS works by successively update $\boldsymbol{\theta}$ by solving the weighted least squares problem, where the weights at the t-th iteration is computed as

$$w_i(\boldsymbol{\theta}^t) = \frac{\rho'(r_i(\boldsymbol{\theta}^t))}{2r_i(\boldsymbol{\theta}^t)}. \tag{2.15}$$

where $\boldsymbol{\theta}^t$ denotes the value of $\boldsymbol{\theta}$ at the t-th iteration.

With the weights (2.15), the weighted least squares problem to update $\boldsymbol{\theta}$ is

$$\boldsymbol{\theta}^{t+1} \leftarrow \arg\min_{\boldsymbol{\theta}} \sum_i w_i(\boldsymbol{\theta}^t)\; r_i(\boldsymbol{\theta}) \tag{2.16}$$

If the robust loss function ρ satisfies some conditions [5], it is proved that the sequence of IRLS iterates converges to the set of critical points of (2.11).

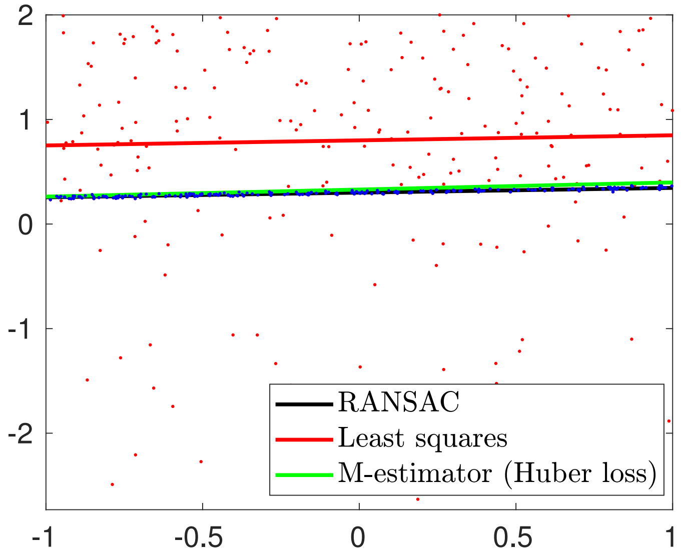


FIGURE 2.6: An example of robust line fitting using M-estimator (with Huber loss function). Unlike least squares, by using a robust loss function, M-estimator is able to return an unbiased estimate which is very close to RANSAC solution.

Figure 2.6 shows the fitting results produced by a m-estimator with Huber loss function. It can be seen that M-estimator is not as affected by the outliers as the least squares.

2.2.2 Least Median of Squares

The Least Median of Squares (LMS) is a robust estimator initially proposed by Rousseeuw [77] that seeks to find an estimate $\boldsymbol{\theta}$ that is the solution of:

$$\min_{\boldsymbol{\theta}} \operatorname{med} \, (r_i(\boldsymbol{\theta}))^2 \tag{2.17}$$

It has been established in [77] that there always exists a solution to (2.17). LMS has high breakdown point of 0.5, which means that it can handle up to 50% of outliers without producing a biased result.

Unfortunately, the non-smooth median operator makes LMS difficult to be solved, as the gradient information cannot be used to perform iterative updates. Consequently, the optimal LMS estimator can only be obtained by bruteforce search over all possible subsets of data or by employing combinatorial optimization techniques. It has been proven in [53] that for a set of N data points in $\mathbb{R}^d$, the LMS problem (2.17) has $\binom{d+\lfloor\frac{n-1}{2}\rfloor}{d}$ local minima.

As enumerating through all the local minima to solve the problem globally is computationally intractable in general, several randomized or heuristics approaches have been proposed to obtain suboptimal solutions [70, 83]. Most of the randomized algorithms proposed to solve LMS have strong connections with RANSAC, which will be discussed in Section 2.3.1.1.

2.2.3 Consensus maximization

Consider a generalization of the least median of squares where we minimize the k-th largest residual. This yields a robust estimate if there are k inliers in the data. Mathematically, this generalized form of least median of squares is defined as

$$\min_{\boldsymbol{\theta}} \max_{i} (r_i(\boldsymbol{\theta}))_k, \tag{2.18}$$

where $(r_i(\boldsymbol{\theta}))_k$ gives the k-th largest residual. This problem remains intractable.

In practice, one may not know in advance the number of true outliers. An alternative way to achieve robustness is to allow only a maximum error for the inliers. This gives rise to the maximum consensus problem

$$\begin{aligned} \max_{\boldsymbol{\theta}} \quad & |\mathcal{I}|, \\ \text{subject to} \quad & r_i(\boldsymbol{\theta}) \le \epsilon, \end{aligned} \tag{2.19}$$

where we maximize instead the consensus or the number of inliers with threshold ϵ of the candidate model $\boldsymbol{\theta}$.

Although maximum consensus is NP-hard, it is commonly used in computer vision because the inlier threshold can be guessed from the application context. For example, in homography estimation, a good estimate requires the transfer errors of the inliers to be within a few pixels.

In several computer vision applications, the residuals are often written in the fractional form

$$r_i(\boldsymbol{\theta}) = \frac{\|\mathbf{a}_i^T \boldsymbol{\theta} + \mathbf{b}_i\|}{\mathbf{c}_i^T \boldsymbol{\theta} + d_i}, \text{ s.t. } \mathbf{c}_i^T \boldsymbol{\theta} + d_i > 0, \tag{2.20}$$

which admits the pseudo-convex property.

2.3 Approximate algorithms for consensus maximization

2.3.1 Randomized Methods

Due to the the intractability of maximum consensus, many practitioners employ the randomized methods to approximately solve the problem. This section introduces the well-known RANSAC algorithm, and several of its variants.

2.3.1.1 RANSAC

RAndom **S**ample **C**onsensus is one of the most commonly used methods for consensus maximization. RANSAC works by repeatedly sampling a minimal subset then fitting the model onto the subset. The number of data points that are consistent with the hypothesis is measured for each sample. Fig. 2.7 illustrates three main steps in RANSAC's sampling process. Finally, the model with the largest consensus size is returned. The sampling process stops after a fix number of iterations or when a stopping criterion is satisfied.

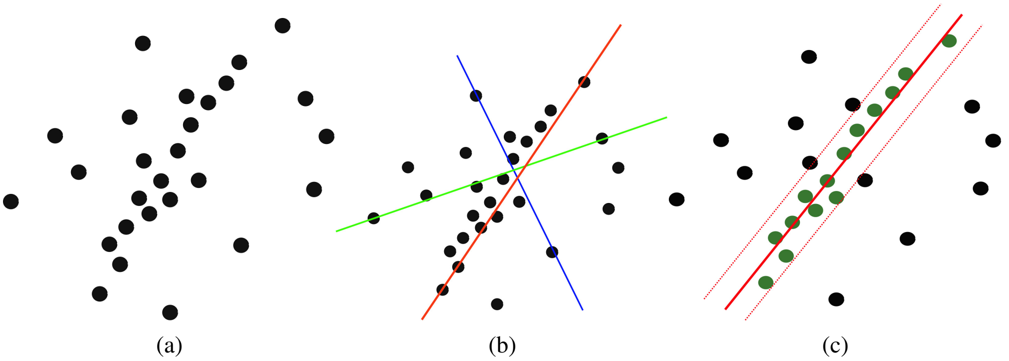


FIGURE 2.7: Illustration of the RANSAC algorithm for fitting a line in 2D. Given a set of points (a), RANSAC randomly samples minimal subsets of two points then fits a model on each subset (b). The model with the largest consensus size is returned (c).

The stopping criterion is computed based on the inlier ratio in the dataset. Let $\boldsymbol{\theta} \in \mathbb{R}^d$ and the proportion of inliers in the set $\mathcal{X}$ of measurements be η. Let $\mathcal{S}$ denote the sampled minimal subset, hence $|\mathcal{S}| = d$. As $\mathcal{S}$ is sampled uniformly from $\mathcal{X}$, the probability of $\mathcal{S}$ containing all inliers is η^d. Consequently, the probability that $\mathcal{S}$ contains at least one outlier is $1 - \eta^d$. Therefore, after T iterations, the probability that none of the minimal samples contains all inliers is $(1 - \eta^d)^T$.

Let ρ be the confidence such that after T minimal subsets have been sampled, at least one of them contains all inliers. Therefore,

$$1 - \rho = (1 - \eta^d)^T, \tag{2.21}$$

and T can be solved as

$$T = \left\lceil \frac{log(1 - \rho)}{log(1 - \eta^d)} \right\rceil \tag{2.22}$$

In practice, the inlier ratio is unknown as a priori. A commonly used technique to circumvent this issue is to incrementally update T based on the largest estimated conesensus size. The RANSAC method can be summarized by Algorithm 2.1.

Algorithm 2.1 RANSAC [31]

```
Require: Data $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$, inlier threshold $\epsilon$, size of minimal subset $d$, maximum number of
    iterations $M$, confident probability $\rho$
    Initialize $\mathcal{I}^* \leftarrow \emptyset$
    $t \leftarrow 0$.
    while $t \leq M$ do
        $t \leftarrow t + 1$
        $\mathcal{S} \leftarrow$ Randomly sample $d$ data points from $\mathcal{X}$
        $\boldsymbol{\theta} \leftarrow$ Fit a model onto $\mathcal{S}$
        $\mathcal{I} \leftarrow \{i | r_i(\boldsymbol{\theta}) \leq \epsilon\}$
        if $|\mathcal{I}| > |\mathcal{I}^*|$ then
            $\mathcal{I}^* \leftarrow \mathcal{I}$
            $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}$
            Update $T$ (stopping criterion).
            if $t \geq T$ then
                Break
            end if
        end if
    end while
    return $\mathcal{I}^*$, $\boldsymbol{\theta}^*$.
```

The iterations required by RANSAC increases exponentially as the ratio of outliers increases. See Figure 2.8 for an illustration. The runtime of RANSAC also increases with the dimensionality of the problem d, which is illustrated in Figure 2.9. This demonstrates that RANSAC is ineffective on highly-contaminated data and/or high dimensional problems.

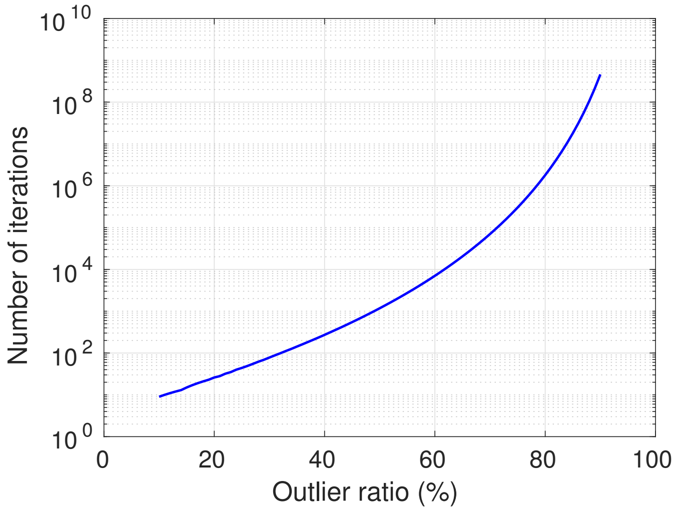


FIGURE 2.8: Number of iterations required by RANSAC versus the outlier rate as predicted with (2.22) with the succesfull probability $\rho = 0.99$ for minimal subsets of size $d = 8$.

2.3.1.2 Locally Optimal RANSAC (LO-RANSAC)

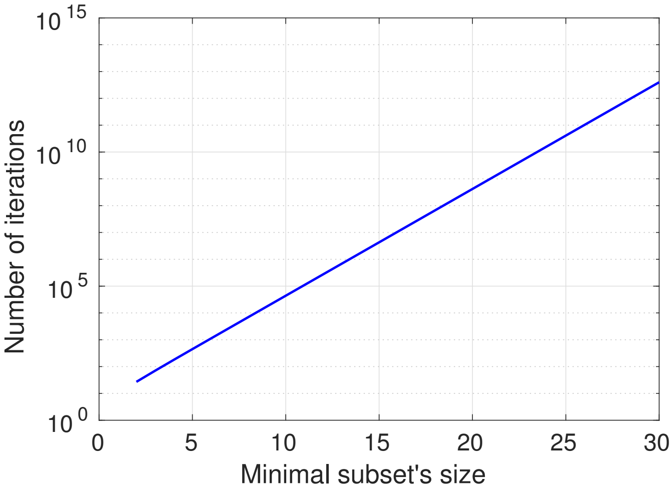


FIGURE 2.9: Number of iterations required by RANSAC versus the size of minimal subset as predicted with (2.22) with the succesfull probability $\rho = 0.99$ and outlier rate of $\rho = 50\%$

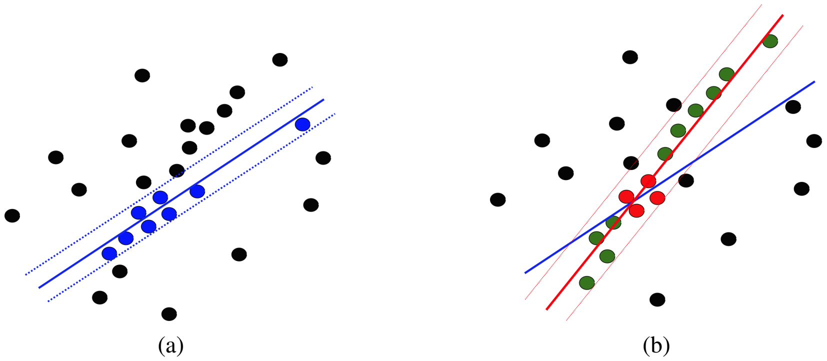


FIGURE 2.10: Illustration of LO-RANSAC algorithm. When the RANSAC solution is updated (left), the inner sampling loop is triggered (right). The inner sampling loop can sample larger-than-minimal subsets.

LO-RANSAC [23, 56] refines the RANSAC estimate by embedding an inner sampling loop into the RANSAC iterations. This is based on the assumption that the estimates from outlier-free minimal subsets are close to the optimal solution. The inner sampling process is triggered when the consensus set is updated by RANSAC. The samples for the inner loop is drawn from the current best-so-far inlier set. This gives LO-RANSAC the advantage of sampling larger-than-minimal subsets. It has been observed that LO-RANSAC gives considerable improvements over RANSAC. Figure 2.10 demonstrates the idea behind the LO-RANSAC algorithm and the method can be summarized in Algorithm 2.2.

2.3.1.3 Guided Sampling

Guided MLESAC (G-MLE) RANSAC samples minimal subsets uniformly from the data since all the measurements are treated equally. However, in two-view geometry problems inlier prior are often available. For instance, the feature correspondences provided by SIFT is associated

Algorithm 2.2 LO-RANSAC [23]

```
Require: Data $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$, inlier threshold $\epsilon$, minimal subset size $d$, maximum number of iterations $M$, confident probability $\rho$, inner sampling iterations $M_L$, size of inner samples $d_L$ ($d_L \geq d$).
$\mathcal{I}^* \leftarrow \emptyset$
$t \leftarrow 0$.
while $t \leq M$ do
    $t \leftarrow t + 1$
    $\mathcal{S} \leftarrow$ Randomly sample $d$ data points from $\mathcal{X}$
    $\boldsymbol{\theta} \leftarrow$ Fit a model onto $\mathcal{S}$
    $\mathcal{I} \leftarrow \{i | r_i(\boldsymbol{\theta}) \leq \epsilon\}$
    if $|\mathcal{I}| > |\mathcal{I}^*|$ then
        $\mathcal{I}^* \leftarrow \mathcal{I}$
        $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}$
        for $t_L = 1, 2, \cdots, M_L$ do
            $\mathcal{S}_L \leftarrow$ Randomly sample $d_L$ data points from $\mathcal{I}^*$
            $\boldsymbol{\theta}_L \leftarrow$ Fit a model onto $\mathcal{S}_L$
            $\mathcal{I}_L \leftarrow \{i | r_i(\boldsymbol{\theta}_L) \leq \epsilon\}$
            if $|\mathcal{I}_L| > |\mathcal{I}^*|$ then
                $\mathcal{I}^* \leftarrow \mathcal{I}_L$
                $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}_L$
            end if
        end for
        Update $T$ (stopping criterion).
        if $t \geq T$ then
            Break
        end if
    end if
end while
return $\mathcal{I}^*$, $\boldsymbol{\theta}^*$.
```

with matching scores. These matching scores can be utilized to prioritize data that are more likely to be inliers – this is the key idea behind Guided MLESAC [84]. The score of each data point is converted into its inlier probability, then a weighted sampling method is applied over the whole dataset in order to select the minimal subsets. As the algorithm progresses, the chance of sampling the inlier-only minimal subset is increased.

Progressive Sample Consensus (PROSAC) Similar to G-MLE, PROSAC [22] also utilizes the matching scores for selecting subsets. This method sorts the matching scores, then uses the sorted scores to sample the subsets from a progressively larger set of tentative correspondences, prioritizing data with higher scores. The set of tentative correspondences is gradually expanded after each iteration, thus PROSAC eventually converges to the standard random sampling.

2.3.2 Deterministic Approximate Methods

Besides the class of randomized methods, deterministic approximation algorithms have also been proposed previously. In the moderately difficult settings, these methods can provide decent solutions. In this section, we review two representative alogrithms, namely the ℓ_1 approximation method and the ℓ_∞ outlier removal method.

2.3.2.1 ℓ_1 approximation

The maximum consensus problem can be equivalently formulated as minimizing the number of outliers. Let $\mathbf{s} = [s_1, \dots s_N]^T$ be the vector of slack variables where s_i corresponds to the i-th constraint. Consider the outlier minimization formulation for consensus maximization:

$$\begin{aligned} \min_{\mathbf{s},\boldsymbol{\theta}} \quad & \|\mathbf{s}\|_0 \\ \text{subject to} \quad & r_i(\boldsymbol{\theta}) \leq \epsilon + s_i \;\; \forall i \end{aligned} \tag{2.23}$$

where $\|.\|_0$ denotes the ℓ_0 norm which counts the number of non-zero elements. The intuition behind (2.23) is that if the data point i is an inlier with respect to the estimate $\boldsymbol{\theta}$, then constraint $r_i(\boldsymbol{\theta}) \leq \epsilon$ holds and s_i can be set to zero. Otherwise, s_i must be assigned with a positive value to satisfy the constraint $r_i(\boldsymbol{\theta}) \leq \epsilon + s_i$. Since the problem (2.23) aims to minimize the non-zero elements of the slack vector, it is equivalently minimizing the number of outliers.

Unsurprisingly, the ℓ_0 norm minimization problem (2.23) is also intractable. The problem can be simplified by relaxing the ℓ_0 norm to ℓ_1 norm

$$\begin{aligned} \min_{\mathbf{s},\boldsymbol{\theta}} \quad & \|\mathbf{s}\|_1 \\ \text{subject to} \quad & r_i(\boldsymbol{\theta}) \leq \epsilon + s_i \;\; \forall i. \end{aligned} \tag{2.24}$$

If the residuals are convex, (2.24) becomes a convex problem which can be solved efficiently using many convex solvers. In the case of linear fitting, $r_i(\boldsymbol{\theta})$ are linear functions, hence (2.24) reduces to a linear program (LP). For some computer vision applications, the residuals can be linearised, thus making the ℓ_1 approximation algorithm an interesting alternative technique for consensus maximization [74]. In later chapters, it will be shown that ℓ_1 approximation fails for highly-contaminated data and/or the outliers are not uniformly distributed.

2.3.2.2 ℓ_∞ outlier removal

Although the minimax problem can be solved efficiently (see Section 2.1.2), it is still easily biased by outliers, especially in cases where the data is unbalanced – see Figure 2.4. Ideally, the outliers should be removed before conducting ℓ_∞ fit on the set of inliers. Based on the above idea, one can conduct recursive ℓ_∞ estimation by removing the data with the largest residuals [79]. This approach may not work for general residual functions. Fortunately, if the resiudals r_i are strictly quasiconvex, it has been shown that removing outliers by discarding the set of measurements with largest residual is a valid approach. Indeed, one can prove that the set of measurements with the largest residuals must contain at least one outlier [79].

Observed, however, that if the contraints with the largest residuals are removed and the fit is performed again with the remaining set, one would get a smaller minimax value. If this process is repeated until the minimax value is not greater than the threshold ϵ, the final solution is a fit $\hat{\boldsymbol{\theta}}$ that is consistent with all the remaining data. Fig. 2.11 shows an example of a line fitting problem where the detailed execution steps of the ℓ_∞ outlier removal algorithm are plotted.

One drawback of this method is that during the outlier removal process, genuine inlers may also be discarded. This heuristics strategy works well in practice if the inliers concentrate at the center of the dataset. In case of unbalanced data, the consensus size returned may be very low and the estiamate if very far from the correct one.

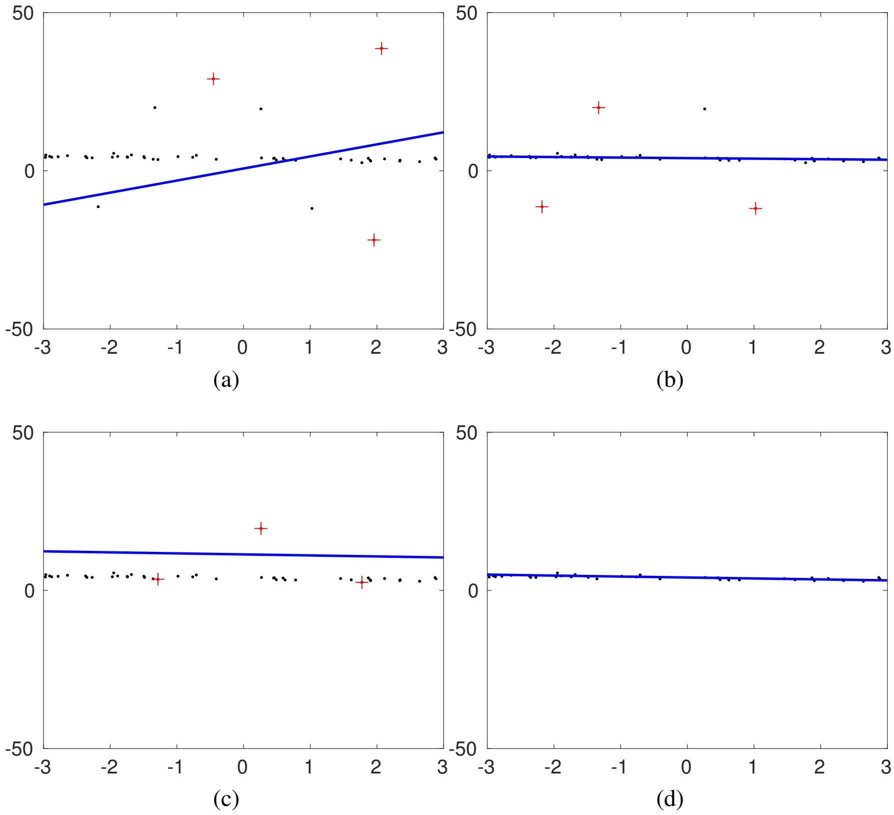
2.4 Exact Algorithms

2.4.1 Mixed Integer Linear Programming (MILP)

In the context of linear regression, one common strategy for solving consensus maximization problem globally is to convert it into an instance of MILP, which can then be tackled by any off-the-shelf solvers. Formally, consider the re-formulation of the maximum consensus problem for robust linear regression:

$$\begin{aligned} \min_{\boldsymbol{\theta},\mathbf{z}} \quad & \sum_i z_i \\ \text{subject to} \quad & |\mathbf{x}_i^T\boldsymbol{\theta} - y_i| \le \epsilon + z_i M, \\ & z_i \in \{0,1\}, \end{aligned} \tag{2.25}$$

where $\mathbf{z} = \{z_i\}_{i=1}^N$ are the indicator variables and M is a large positive number. This method is referred to as "big M" method, as M must be big enough to compensate the inconsistent constraints. Intuitively, if the i-th data point is an inlier with respect to an estimate $\boldsymbol{\theta}$, the constraint $|\mathbf{x}_i\boldsymbol{\theta} - y_i| \le \epsilon$ is satisfied, thus the indicator z_i can be set to zero. Otherwise, z_i must be "turned on" ($z_i = 1$) to satisfy the constraint. Solving (2.25) is equivalent to minimizing the number of outliers. The guidelines for choosing values of M can be found in [20]. Note that since each absolute value constraint is equivalent to two linear constraints, (2.25) is an instance



(a)

(b)

(c)

(d)

FIGURE 2.11: Example of a line fitting problem using ℓ_∞ outlier removal approximation. (a) ℓ_∞ fit on the whole dataset, points with largest residual are plotted in red. (b) (c) The data points with largest residuals are removed and the ℓ_∞ fit is recursively executed on the remaining data. Note that in (c), two genuine inliers were removed. (d) The algorithm stops when the maximum residual of the ℓ_∞ fit is not greater than the inler threshold ϵ and the the final estimate is returned.

of MILP. Given the optimal solution $\mathbf{z}^*$ of (2.25), the best inlier set can be obtained by:

$$\mathcal{I}^* = \{\mathbf{x}_i, y_i | z_i^* = 0\}.$$

In the optimization literature, solving a MILP is also intractable in general. Usually, the solvers must conduct some form of branch-and-bound search to solve for the optimal solution. It is a well-known fact that the worst-case complexity of branch-and-bound is exponential. Therefore, converting the problem into an MILP does not make the problem any easier, even with the state-of-the-art solvers.

2.4.2 Tree Search

As previously discussed, due to the combinatorial nature of consensus maximization, it can only solved globally by conducting some forms of exaustive search. Therefore, many works on exact methods for the problem have proposed different techniques to boost up the search process.

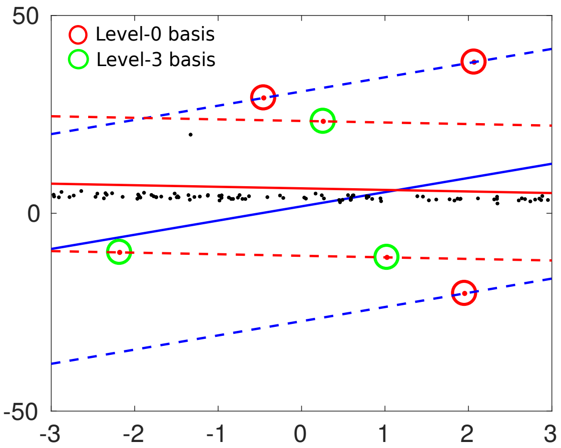


FIGURE 2.12: Illustration of the concept of basis for a line fitting problem. Red circle: level-0 basis. Green circle: level-3 basis. The blue solid line is the ℓ_∞ estimation for all the points, which is identical to the ℓ_∞ fit for the level-0 basis. There are 3 points that vioate the estimate corresponding to the level-3 basis.

Among the proposed methods in the literature, using tree search is a promising approach [67]. The benefit of the tree formulation is that one can apply common tree traversal techniques to solve the maxium consensus problem. The tree search formulation is made amenable by the theory of the LP-type problems [78], which will be introduced briefly in the following.

Definition 2.1. (LP-type problem) Given a finite set of constraints $\mathcal{S}$ and a function $f : 2^{\mathcal{S}} \rightarrow \mathcal{O}$ where $\mathcal{O}$ is an linearly ordered set. The pair $(\mathcal{S}, f)$ is called an LP-type problem if f is monotone and local, i.e., for any $\mathcal{P} \subseteq \mathcal{Q} \subseteq \mathcal{S}$

- (Monotinicity) $f(\mathcal{P}) \leq f(\mathcal{Q}) \leq f(\mathcal{S})$, and
- (Locality) If $f(\mathcal{P}) = f(\mathcal{Q}) > -\infty$ and $f(\mathcal{Q} \cup \{h\}) > f(\mathcal{Q})$ for any constraint $h \in \mathcal{S}$, then $f(\mathcal{P} \cup \{h\}) > f(\mathcal{P})$.

Definition 2.2. (Basis) A basis $\mathcal{B}_{\mathcal{X}}$ of a set $\mathcal{X}$ is a subset of $\mathcal{X}$ such that $f(\mathcal{X} \backslash \mathbf{b}_i) < f(\mathcal{X}) \; \forall \mathbf{b}_i \in \mathcal{B}_{\mathcal{X}}$.

Definition 2.3. (Violation, Level, Coverage) For a given basis $\mathcal{B}$, a constraint $h \in \mathcal{S}$ *violates* $\mathcal{B}$ if $f(\mathcal{B} \cup \{h\}) > f(\mathcal{B})$. The violation set $V(\mathcal{B})$ is the set of all constraints that violate $\mathcal{B}$. The cardinality of $V(\mathcal{B})$ is called the level $l(\mathcal{B})$ of $\mathcal{B}$. The set $\mathcal{S} \backslash V(\mathcal{B})$ is called the coverage $C(\mathcal{B})$ of $\mathcal{B}$

Definition 2.4. (Combinatorial dimension) The maximum cardinality of any basis of $\mathcal{S}$ is defined as the combinatorial dimension of the LP-type problem $(\mathcal{S}, f)$.

Definition 2.5. (Feasible basis) A basis $\mathcal{B}$ is feasible if $f(\mathcal{B}) \leq \epsilon$

The LP-type problems are considered to be the generalizations of linear programming (LP) problem [78] and have strong connection to the ℓ_∞ estimation problem discussed in Sec. 2.1.2.

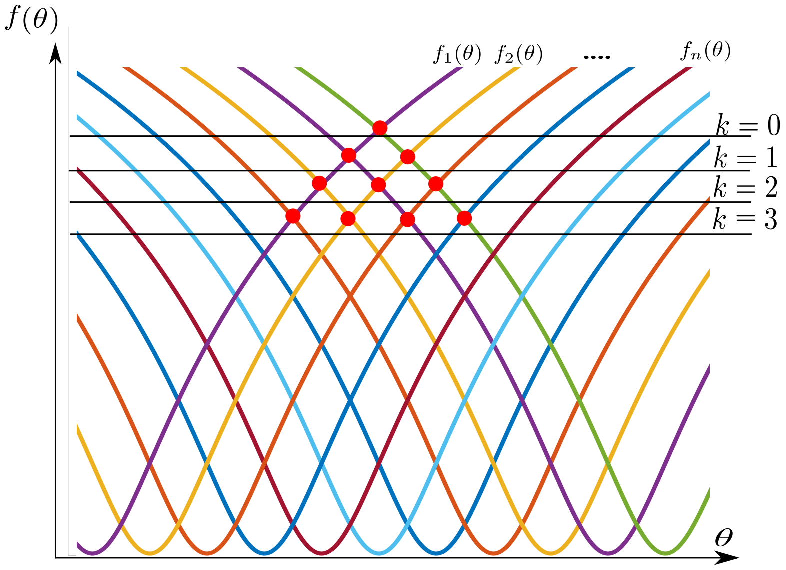


FIGURE 2.13: Tree structure of the quasiconvex problems in one dimension. $f_i(\boldsymbol{\theta}$ are the quasi-convex functions. Red dots indicate local minima of different levels. It can be seen from the tree that a node from level k can be reached from a node of level $k-1$

Particularly, it has been established that the minimax of a set of quasiconvex residuals is an LP-type problem [27]. Therefore, by letting $f(\mathcal{X})$ in Definition 2.1 to be the objective function (the minimax value) of the ℓ_∞ estimation problem for the data $\mathcal{X}$, solving the ℓ_∞ problem amounts to finding the level-0 basis for the dataset. Fig. 2.12 illustrates the level-0 basis and a level-3 basis for a line fitting problem. Observe that the ℓ_∞ fit for the whole point set is identical to the ℓ_∞ fit for the level-0 basis. Note that the combinatorial dimension of problems with quasi-convex residuals is $d+1$, given that $\boldsymbol{\theta} \in \mathbb{R}^d$ [67].

Let $\mathcal{I}^*$ be the best inlier subset of the maximum consensus problem and $\mathcal{B}^*$ be the basis of $\mathcal{I}^*$. The problem reduces to finding $\mathcal{B}^*$, which is the basis at the lowest level (shallowest node) that is feasible (i.e., $f(\mathcal{B}^*) \leq \epsilon$). Based on the reduction theorem [60], the optimal solution can be found by enumerating all subsets of size at most $(d+1)$. However, this is impractical for problems in high dimensions with large number of constraints. It has been established in the literature [67] that this problem can be solved more efficiently without exhaustive enumeration of the bases, which can be made possible by the following theorem.

Theorem 2.6. *Every basis of level k can be reached from a basis of level $k-1$.*

Refer to [67] for the proof of Theorem (3.7). This theorem enables the formulation of the tree structure for solving consensus maximization by associating each node of the tree with a basis according its level. A node of level k posesses child nodes of level $k+1$. The maximum number of children of each node is therefore the combinatorial dimension of the problem. Fig. 2.13 illustrates the concept of a tree for a problem in one dimension.

As the tree is formulated, different tree traversal techniques can be used to search the tree. The traditional BFS technique was proposed in [67]. Recently, the use of A* [18] has shown great improvements in search for the optimal solution. However, even the state-of-the-art exact algorithms are still far from practical for real-life problems with high dimensions and large number of measurements.

2.4.3 Branch and Bound

For many difficult non-convex optimization problems, if a globally optimal solution is desired, branch-and-bound is a commonly used algorithm. This method works by successively paritioning the search domain into smaller subdomains. For each subdomain, a bounding function is evaluated. Based on the associated value of the bounding function, the algorithm can decide whether the subdomain is pruned or further partitioned. Without loss of generality, consider the optimization problem over the domain $\mathbb{S}$

$$\max_{\mathbf{x}\in\mathbb{S}} f(\mathbf{x}), \tag{2.26}$$

where $f(\mathbf{x})$ can be a non-convex function. Let $f(\hat{\mathbf{x}})$ be the best-so-far objective value of the problem obtained during the search process. For a subdomain $\mathbb{D} \subseteq \mathbb{S}$, let $g(\mathbb{D})$ denotes its bounding function. The function $g(\mathbb{D})$ is constructed in such a way that it can be evaluated efficiently and the following condition must be satisfied:

$$g(\mathbb{D}) \geq f(\mathbf{x}) \;\; \forall \mathbf{x} \in \mathbb{D} \tag{2.27}$$

Clearly, if $g(\mathbb{D}) \leq f(\hat{\mathbf{x}})$, the subdomain $\mathbb{D}$ contains no values of $\mathbf{x}$ that can improve the solution,

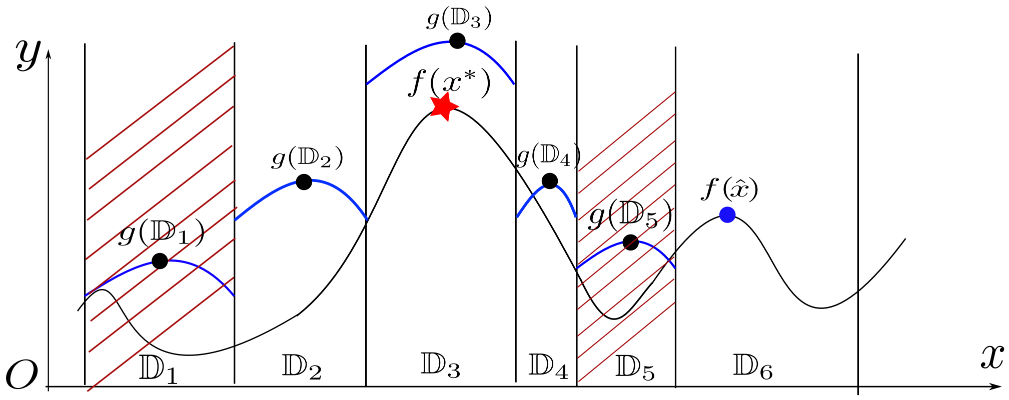


FIGURE 2.14: Illustration of branch and bound for a problem in one dimension. The current best-so-far solution is $f(\hat{x})$. The function $g(\mathbb{D}_i)$ is the (upper) bounding function for the partition $\mathbb{D}_i$. The partitions $\mathbb{D}_1$ and $\mathbb{D}_5$ are pruned because $g(\mathbb{D}_1) < f(\hat{x})$ and $g(\mathbb{D}_5) < f(\hat{x})$, while $\mathbb{D}_2$, $\mathbb{D}_3$ and $\mathbb{D}_4$ are kept for further partitioning since their upper bound values are greater than $f(\hat{x})$.

thus $\mathbb{D}$ can safely be pruned. Otherwise, the subdomain $\mathbb{D}$ is then continued to be further partitioned. If a subdomain $\mathbb{D} \subseteq \mathbb{S}$ is selected to be partitioned, it will be divided into n subdomains $\mathbb{D}_1, \mathbb{D}_2, \ldots \mathbb{D}_n$ such that

$$\mathbb{D}_1 \cup \mathbb{D}_2 \cup \cdots \cup \mathbb{D}_n = \mathbb{D}, \tag{2.28}$$

and

$$\mathbb{D}_i \cap \mathbb{D}_j = \emptyset \;\; \forall i, j \in \{1, 2, \ldots, n\} \tag{2.29}$$

The branch and bound algorithm stops when the search space can no longer be partitioned (the smallest size of a partition is defined up to a precision γ), and the best-so-far solution is returned as the global optimum. The complexity of the algorithm is exponential. Fig. 2.14 illustrates the mechanism behind branch and bound for a one dimentional optimization problem.

2.5 Preprocessing technique - guaranteed outlier removal

Although the ℓ_∞ outlier removal algorithm discussed in 2.3.2.2 can be applied if the data is not heavily biased, there are still chances that genuine inliers are removed during the process. Thus, before a data instance is discarded, it must be tested whether it is a true outliers. An algorithm to efficiently identify true outliers was developed in [17]. The algorithm begins with the MILP formulation for the maximum consensus problem (2.25) and rewrite it as a nested problem

$$\min_{k=1,..,N} \beta^k \tag{2.30}$$

where β^k is defined as the optimal value of the following subproblem:

$$\begin{aligned} \min_{\boldsymbol{\theta}, \mathbf{z}} \quad & \sum_{i \neq k} z_i \\ \text{subject to} \quad & |\mathbf{x}_i^T - y_i| \leq \epsilon + z_i M, \\ & z_i \in \{0, 1\}, \\ & |\mathbf{x}_k^T \boldsymbol{\theta} - y_k| \leq \epsilon. \end{aligned} \tag{2.31}$$

Intuitively, the objective of each subproblem (2.31) is the minimum number of data points that need to be removed to result in a consistent subset such that the data point k must not be removed. Assume that we are given the lower bound value α^k and a suboptimal solution $\hat{\boldsymbol{\theta}}, \hat{\mathbf{z}}$ to (2.31) such that $\alpha^k \leq \beta^k$ and $\sum_i \hat{z}_i \geq \sum_i z_i^*$ (with $\mathbf{z}^*$ is the optimal solution to (2.31)). Dentote $\|\hat{\mathbf{z}}\|_1$ by $\hat{u}$. Then, it can be proven that if $\alpha^k > \hat{u}$, the data point $\{\mathbf{x}_k, y_k\}$ is a true outlier [17].

This outlier removal algorithm can be used as a pre-processing step to remove points that are guaranteed to be true outliers with respect to the optimal solution. With a proper choice of points to remove, it can significantly boost the runtime for optimal solvers. Certainly, if every points in the data is tested and outliers are removed, one come up with the optimal solution for the original problem. However, this approach may not be faster than other state-of-the-art solvers as each test requires the execution of the MILP solver.

Chapter 3

Random Tree Sampling for Consensus Maximization

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

Huu Le, Tat-Jun Chin and David Suter: RATSAC - Random Tree Sampling for Consensus Maximization. In Digital Image Computing: Techniques and Applications (DICTA) 2017 - (Oral presentation) - **Winner of the DSTG award**

The published paper is available at: http://ieeexplore.ieee.org/document/8227480/

Statement of Authorship

Title of Paper	RATSAC – Random Tree Sampling for Consensus Maximization		
Publication Status	☑ Published □ Submitted for Publication	□ Accepted for Publication □ Unpublished and Unsubmitted work written in manuscript style	
Publication Details	**Huu Le**, Tat-Jun Chin and David Suter: RATSAC – Random Tree Sampling for Consensus Maximization. In Digital Image Computing: Techniques and Applications (DICTA) 2017 - (Oral presentation) - **Winner of the DSTG award**		

Principal Author

Name of Principal Author (Candidate)	Huu Le		
Contribution to the Paper	- Proposed the idea and developed the algorithm. - Implemented and tested the algorithm. - Conducted all the experiments to validate the performance of the proposed method. - Wrote the manuscript.		
Overall percentage (%)	80%		
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.		
Signature		Date	20th Dec 2017

Co-Author Contributions

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

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Tat-Jun Chin		
Contribution to the Paper	- Proposed the general research direction. - Supervised the development of this work. - Refined the manuscript.		
Signature		Date	20th Dec 2017

Name of Co-Author	David Suter		
Contribution to the Paper	- Supervised the development of this work. - Refined the manuscript.		
Signature		Date	20th Dec 2017

3.1 Introduction

Many applications in computer vision require the estimation of a model that best fits a set of observations. While closed-form solutions exist, they are highly vulnerable to outliers. Take least squares estimation as an example, the presence of only one erroneous outlier can drastically drive the estimation away form the actual solution. M-estimation algorithms [46] and other robust estimators were then developed to implicitly discard the outliers during the optimization process. Such estimators, however, are still subjected to the non-convexity of the cost function and thus hard to be solved optimally. RANSAC [31] and its variants, which belong to the class of randomized methods, are arguably the most widely used approaches due to their simplicity and the ability to provide decent results with resonable execution time. Leveraging from the random sampling scheme of RANSAC, considerable efforts have been devoted to improving its efficiency either by modifying the sampling strategy or changing the cost function. LO-RANSAC [23] (and its improvement [56]), for instance, performs a local optimization procedure every time RANSAC's solution is updated, which is done by iteratively executing the inner sampling steps with larger-than-minimal subsets then re-estimate the model with iterative re-weighted least squares (IRLS). MLESAC [85] combines the sampling strategy with the maximum likelihood framework to solve the problem. In the class of guided sampling using point priors, PROSAC [22] and Guided-MLESAC [84] utilize the feature matching scores to prioritize samples that are more likely to result in a good solution. The drawback of these guided sampling strategies lies in the fact that they rely heavily on the priors provided by the feature extractors, which may not be readily available in some applications.

In parallel to the class of randomized methods, it is also desirable to develop methods that can solve the problem globally. In this area, there has been some remarkable advances recently [18, 59, 72, 96]. However, solving the maximum consensus up to global optimality is by no mean an easy task as this problem is computationally intractable in general. Meanwhile, the solution provided by the randomized methods are quite unpredictable and still far from optimal.

To mitigate the gap between speed and accuracy, we propose an entirely new guided sampling strategy that can take advantage of the geometric information embedded in the problem. Our method works under the framework of the LP-type problems and Monte Carlo Tree Search. Unlike other guided sampling schemes such as PROSAC [22] or Guided-MLESAC [84], inlier priors are *not required* in our method, even though they can still be utilized to improve the results. During the sampling process, outliers are implicitly removed by the algorithm, which assists the later iterations in finding a good solution much easier. Empirical experiments show that our algorithm outperforms other state-of-the-art methods and can approach optimal solutions. We name our algorithm RATSAC[1] (**RA**ndom **T**ree **SA**mple **C**onsensus).

[1] Coincidentally, the acronym "RATSAC" has been used in another work [76] published 10 years ago that also tackled the robust estimation problem. Our approach, however, is different from the approach used by the authors in

3.2 Problem definition

Given a set of N observations $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ and an inlier threshold ϵ, the goal of consensus maximization is to find a parameter vector $\boldsymbol{\theta} \in \mathbb{R}^d$ that is consistent with the largest subset of measurements. In other words, one wishes to select the best subset $\mathcal{I}^* \subseteq \mathcal{X}$ that contains the maximum number of consistent constraints (or "inliers"). The problem can be mathematically formulated as:

$$\begin{aligned} &\max_{\boldsymbol{\theta}, \mathcal{I} \subseteq \mathcal{X}} && |\mathcal{I}| \\ &\text{subject to} && r_i(\boldsymbol{\theta}) \leq \epsilon \qquad \forall i \in \mathcal{I} \end{aligned} \tag{3.1}$$

where $\{r_i(\boldsymbol{\theta})\}_{i=1}^N$ are the residual functions that determine the estimation errors with respect to the transformation $\boldsymbol{\theta}$. Throughout this work, we focus on the *quasi-convex* residual functions, particularly, those in the form of:

$$r_i(\boldsymbol{\theta}) = \frac{\|\mathbf{A}_i\boldsymbol{\theta} + \mathbf{b}_i\|_p}{\mathbf{c}_i\boldsymbol{\theta} + d_i} \text{ with } \mathbf{c}_i\theta + d_i > 0 \tag{3.2}$$

where $\mathbf{A}_i \in \mathbb{R}^{2\times d}$, $\mathbf{b} \in \mathbb{R}^2$, $\mathbf{c}_i \in \mathbb{R}^d$ and $d_i \in \mathbb{R}$. The notation $\|.\|_p$ denotes the ℓ_p norm ($p = 1, 2$ and ∞ are commonly used).

Quasi-convex residual is quite popular in many geometry estimation problems. In practice, the set of observations $\{\mathbf{x}_i\}$, the inlier threshold ϵ and the residual functions $\{r_i(\boldsymbol{\theta})\}$ are problem-specific. A few examples are listed below.

- *Linear regression:* In this type of problem, an observation is given as a point $\mathbf{a}_i \in \mathbb{R}^d$ and a scalar b_i: $\mathbf{x}_i = [\mathbf{a}_i^T \; b_i]^T$. The estimate is a hyper-plane $\boldsymbol{\theta} \in \mathbb{R}^d$ with the residual function defined as

$$r_i(\boldsymbol{\theta}) = |\mathbf{a}_i^T\boldsymbol{\theta} - b_i| \tag{3.3}$$

  Besides such straightforward applications as line and plane fitting, robust linear regression is commonly employed for several other geometric problems in computer vision. In fundamental matrix estimation, for instance, the set of epipolar constraints can be linearized to form a linear regression problem. The parameter $\boldsymbol{\theta}$ that needs to be estimated is a 3×3 fundamental matrix and ϵ defines the algebraic error threshold for the inliers.

- *Homography Estimation:* Each data point $\mathbf{x}_i$ is an initial match $\mathbf{x}_i = \{\mathbf{u}_i; \mathbf{v}_i\}$ where $\mathbf{u}_i$ is a 2D image point in one view and $\mathbf{v}_i$ is the corresponding match in the other view. These initial matches are provided by some feature extractors. The estimation $\boldsymbol{\theta} \in \mathbb{R}^9$ is a 3×3

[76]. Since we are not comparing our algorithm against [76], we think that it would be interesting to contribute to the robust estimation toolbox another "RATSAC" algorithm.

homography matrix and the residual function $r_i(\boldsymbol{\theta})$ is defined as the transfer error:

$$r_i(\boldsymbol{\theta}) = \frac{\|(\theta_{1:2} - \mathbf{v}_i\theta_3)\tilde{\mathbf{u}_i}\|_p}{\theta_3\tilde{\mathbf{u}}_i} \tag{3.4}$$

where θ_j is the j-th row of the homography matrix and $\tilde{\mathbf{u}_i} = [\mathbf{u}^T 1]^T$.

- *Triangulation:* An observation in a triangulation problem is an image point $\mathbf{x}_i$ in the i-th view together with its corresponding camera matrix $\mathbf{P}^i$. The estimate $\boldsymbol{\theta}$ is the point in 3D that corresponds to the image point $\mathbf{x}_i$. The residual function for triangulation is determined by the reprojection error

$$r_i(\boldsymbol{\theta}) = \frac{\|(\mathbf{P}^i_{1:2} - \mathbf{x}_i\mathbf{P}^i_3)\boldsymbol{\theta}'\|_p}{\mathbf{P}^i_3\boldsymbol{\theta}'} \tag{3.5}$$

where $\boldsymbol{\theta}' = [\boldsymbol{\theta}^T \;\; 1]^T$, $\mathbf{P}^i_k$ is the k-th row of the camera matrix associated with the i-th view.

It has been established that (3.4) and (3.5) are quasi-convex, while (3.3) is convex which also satisfies the quasi-convex requirements. The interested readers are referred to [51] for more details about quasi-convex functions and several other applications that can be solved by our framework.

3.3 Background

Before proceeding, we introduce some definitions and problems that will be used later throughout the paper. Note that an “observation” (or “measurement”) can sometimes be regarded to as “constraint” and these terms can be used interchangeably.

Definition 3.1. **(ℓ_∞ estimation problem)** Given a set of observations $\mathcal{X}$, the ℓ_∞ problem finds the estimate $\boldsymbol{\theta}$ that minimize the maximum residual. Let $f(\mathcal{X})$ be the solution of this problem. Then $f(\mathcal{X})$ is written as:

$$f(\mathcal{X}) = \min_{\boldsymbol{\theta}} \max_{i=1:|\mathcal{X}|} r_i(\boldsymbol{\theta}) \tag{3.6}$$

Recently, the ℓ_∞ estimation problem (which is also referred to as the “minimax” problem) for quasi-convex residuals has been explored extensively in computer vision research. The attraction of (3.6) lies in the fact that it has one *single local minimum*, which is therefore also the global minimum. Multiple methods to obtain the ℓ_∞ estimate have been introduced in the literature[60, 51, 27]. Interestingly, the ℓ_∞ framework has a strong relationship with the class of LP-type problems and can be solved by searching through the subsets of size at most $(d + 1)$ [60]. As will be shown in the following sections, the problem (3.6) is the sub-problem for consensus maximization in our algorithm.

Definition 3.2. (LP-type problem) Given a finite set of constraints $\mathcal{S}$ and a function $f : 2^{\mathcal{S}} \rightarrow \mathcal{O}$ where $\mathcal{O}$ is an linearly ordered set. The pair $(\mathcal{S}, f)$ is called an LP-type problem if f is monotone and local, i.e. for any $\mathcal{P} \subseteq \mathcal{Q} \subseteq \mathcal{S}$,

- (Monotinicity) $f(\mathcal{P}) \leq f(\mathcal{Q}) \leq f(\mathcal{S})$, and
- (Locality) If $f(\mathcal{P}) = f(\mathcal{Q}) > -\infty$ and $f(\mathcal{Q} \cup \{h\}) > f(\mathcal{Q})$ for any constraint $h \in \mathcal{S}$, then $f(\mathcal{P} \cup \{h\}) > f(\mathcal{P})$.

Definition 3.3. (Basis) A basis $\mathcal{B}_{\mathcal{X}}$ of a set $\mathcal{X}$ is a subset of $\mathcal{X}$ such that such that $f(\mathcal{X} \backslash \mathbf{b}_i) < f(\mathcal{X})\ \forall \mathbf{b}_i \in \mathcal{B}_{\mathcal{X}}$.

A basis $\mathcal{B}_{\mathcal{X}}$ can also be defined as the minimal subset of $\mathcal{X}$ such that $f(\mathcal{B}_{\mathcal{X}}) = f(\mathcal{X})$. For a *non-degenerate* configuration, each subset $\mathcal{X}' \subseteq \mathcal{X}$ has a unique basis. In this work, we assume that all problems are non-degenerate, as degenerated cases can be tackled by applying infinitesimal perturbations [58].

Definition 3.4. (Violation, Level, Coverage) For a given basis $\mathcal{B}$, a constraint $h \in \mathcal{S}$ *violates* $\mathcal{B}$ if $f(\mathcal{B} \cup \{h\}) > f(\mathcal{B})$. The violation set $V(\mathcal{B})$ is the set of all constraints that violate $\mathcal{B}$. The cardinality of $V(\mathcal{B})$ is called the level $l(\mathcal{B})$ of $\mathcal{B}$. The set $\mathcal{S} \backslash V(\mathcal{B})$ is called the coverage $C(\mathcal{B})$ of $\mathcal{B}$.

Intuitively, the violation set of a basis $\mathcal{B} \subseteq \mathcal{X}$ contains constraints that have residuals greater than $f(\mathcal{B})$. For the optimal solution $\mathcal{B}^*$, $f(\mathcal{B}^*) \leq \epsilon$, the coverage $C(\mathcal{B}^*)$ is the optimal inlier set $\mathcal{I}^*$ and the violation set contains all the outliers with respect to the optimal estimate $\boldsymbol{\theta}^*$.

Definition 3.5. (Combinatorial dimension) The maximum cardinality of any basis of $\mathcal{S}$ is defined as the combinatorial dimension of the LP-type problem $(\mathcal{S}, f)$. The combinatorial dimension of all quasi-convex problems is $(d + 1)$, given that $\boldsymbol{\theta} \in \mathbb{R}^d$.

3.4 Maximum consensus as tree search

3.4.1 Outlier removal

It is a well-known fact that L_∞ optimization is still vulnerable to outliers. Starting from the initial fit for the whole set of constraints, the outliers must be removed in other to get to the optimal solution. In the context of robust estimation, there have been several proposals to remove outliers optimally or heuristically. One trivial way is to iteratively perform fitting then remove measurements that induce the largest residuals until the cost function is not greater than a given threshold ϵ [79]. However, this approach may end up removing genuine inliers. The authors in [17] introduced a method which can guarantee to identify the actual outliers. Yet, in order for all true outliers to be discarded by this method, there must be a sequence mixed integer programming problems to be solved in order to test every single data point. Consequently, for highly contaminated data, the approach of removing true outliers will not give any advantage over solving the original problem.

3.4.2 Tree structure

Let $\mathcal{I}^*$ be the best inlier subset of the problem and $\mathcal{B}^*$ be the basic of $\mathcal{I}^*$. The problem reduces to finding $\mathcal{B}^*$, which is the basic at the lowest level (shallowest node) with $f(\mathcal{B}^*) \leq \epsilon$. Based on the reduction theorem [60], one can just enumerate all subsets of size at most $(d+1)$ to find the best solution. However, this is impractical for problems in high dimensions with large number of constraints. It has been established in the literature that this problem can be solved more efficiently without exhaustive enumeration of the basics.

Definition 3.6. A basic $\mathcal{B}$ is feasible if $f(\mathcal{B}) \leq \epsilon$

Theorem 3.7. *Every basic of level k can be reached from a basic of level $k-1$. (Refer to [67] for the proof of this Theorem)*

Theorem (3.7) advocates the formulation of the tree structure for solving consensus maximization by associating each node of the tree with a basic (denote $\mathcal{B}_{\mathbf{N}}$ as the basic associated with node $\mathbf{N}$). A node of level k posesses child nodes of level $k+1$. The number of children of each node is therefore the combinatorial dimension of the problem. Figure 1a illustrates the concept of a tree for a problem in one dimension.

Given the tree structure, many popular tree traversal techniques can be applied to solve the problem. Starting from level 0 basic (root node), Matousek [67] conducts the breath-first-search to find the shallowest node that is feasible. [18] uses A* to improve the search by defining a heuristic function that assists the search process to prioritize the promising paths. This approach has gained sigificant speed improvement compared with traditional breath-first-search.

Unfortunately, all the globally optimal methods are still inferior to randomized approaches as it takes a huge amount of time to explore the basics which is impractical in real-time applications. On the other hand, if all minimal subsets are enumerated by brute-force, one still can not guarantee to find the optimal solution. This leads to the large gap between the current RANSAC variants and the globally optimal solutions, thus the RANSAC results are still far from optimal for data with high outlier rate.

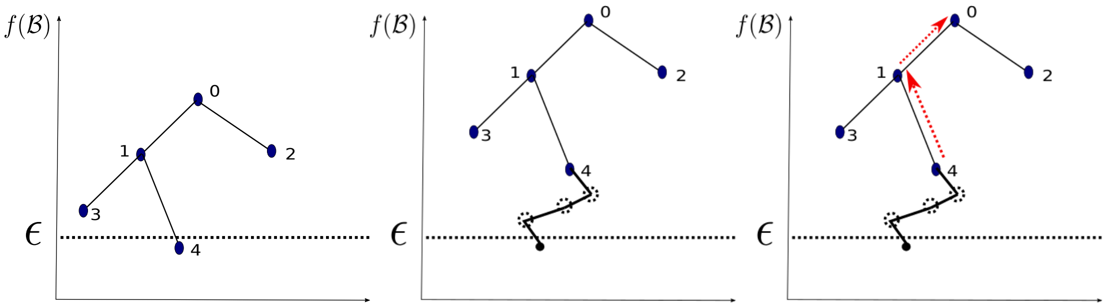


FIGURE 3.1: Left: Illustration of a simple tree for a problem in one dimension. Node 0 is the root node (node of level 0); Node 0 and 1 are fully expanded while node 2, 3, 4 are not. Node 4 is a feasible node; Middle: Illustration of a default policy starting at node 4. The child nodes are selected randomly until feasible basic is reached, the coverage of this feasible node is the reward for node 4. Right: Illustration of back propagation (BP) process. The reward of node 4 after the default policy step is updated through its parent node all the way to the root node.

From the tree structure, it is obvious that in order to solve the problem globally, one must traverse from the top of the tree in a breath-first-search manner. The traditional RANSAC method lacks this intuition, where it samples can be anywhere in the tree. In addtion, the samples generated by current random sampling approaches are independent from each other and the outlier rate at each iteration kept unchanged. In the next section, we show how these weaknesses can be tackled using Monte-Carlo Tree Search. It will also be demonstrated that by using our proposed sampling approaches, the uniform sampling nature of RANSAC is converted to a guided sampling strategy that utilizes the geometric information of the maximum consensus problem.

3.5 Monte Carlo Tree Search

As solving the problem exactly is intractable in general, we seek the suboptimal solution to the problem in a randomized approach. Taking advantage of the tree structure developed in the previous section, we introduce a new algorithm based on the foundation of Monte Carlo Tree Search (MCTS). MCTS has gained considerable attentions recently due to its success in solving various combinatorial search problems whose exact solutions are hard to find. This technique is used extensively to solve decision problems by taking random samples in the decision space and incrementally build the tree according to the sampling results. Using MCTS for searching on a tree problem can be considered as successively applying the Multi-Arm Bandit problem [8] for each node until a feasible node is reached. More details about the theory behind this method and a complete review of its applications can be found in [15].

We introduce some definitions to make it convenient for further discussions.

Definition 3.8. (Feasible node) A node $\mathbf{N}$ on the search tree is called a *feasible node* if its associated basic $\mathcal{B}_{\mathbf{N}}$ is feasible.

Definition 3.9. (Full expansion) If the number of children of a node is equal to its combinatorial dimension, it is called a *fully expanded* node.

Definition 3.10. (Reward) The reward ($\mathbf{R_N}$) of a node $\mathbf{N}$ is defined as the cardinality of $\mathcal{C}(\mathcal{B}_\mathbf{N})$ if $\mathcal{B}_\mathbf{N}$ is feasible and 0 otherwise.

Apprarently, the maximum number of children that a node posesses is equivalent to the combinatorial dimension of the problem. Therefore, it can no longer be expanded if this number reaches $d + 1$.

To simplify the algorithm, we compute reward based on the cost function of the maximum consensus problem. However, this can be extended to other types of cost function, for instance the likelihood used in MLESAC [85]

Starting from the root node, at each stage of the search process, the tree is gradually built by adding new leaf nodes. Each node $\mathbf{N}$ added to the tree is associated with a total back propagated reward $\mathbf{Q_N}$ which is the result of a random simulation process (discussed below). Note that by definition 3.10, if a node is infeasible, it reward $\mathbf{R_N}$ is 0, but its total back propagated reward $\mathbf{Q_N}$ may be greater than 0. The tree continues to grow until a feasible node is added to the tree. The whole process is repeated until a predefined budget (time, number of iterations) is met.

At each stage, MCTS algorithm executes three main procedures:

- **Tree policy (TP):** (Algo. (3.2)) starting from a given parent node $\mathbf{N}$, Tree Policy decides which nodes among its children to descend. In case the node is not fully expanded, a random child of this added to this node. Otherwise, the child node $\mathbf{N}_k$ with the largest Upper Confident Bound (UCB) is selected, where UCB is defined as:

$$U(\mathbf{N}_k, C_p) = \frac{\mathbf{Q}_{\mathbf{N}_k}}{V(\mathbf{N}_k)} + C_p\sqrt{\frac{2\ln V(\mathbf{N})}{V(\mathbf{N}_k)}} \tag{3.7}$$

  where $\mathbf{Q_N}$ is the total reward for node $\mathbf{N}$, $V(\mathbf{N})$ denotes the number of times node $\mathbf{N}$ has been visited during the search process and C_p is a positive constant which can be tuned based on applications.

- **Default Policy (DP):** (Algo. (3.3)) This procedure is used to as a *simulation process* to estimate the reward for particular node. From the newly added node, the DP randomly descend down until a terminal node is reached. See Figure 1b for an illustration.

- **Back Propagation (BP):** (Algo. (3.4)) After the DP process, a reward $\mathbf{R_N}$ of a child node $\mathbf{N}$ is recursively updated to its parent node all the way to the root node. This process is depicted in Figure 1c.

The UCB defined in (3.7) is designed to balance between exploration and exploitation and is one of the most commonly used algorithm in MCTS. There exist a number of other algorithms which and can also be applied to our framework. See [15] for more details.

3.6 Main Algorithm

The main random tree search process is described in Algorithm (3.1). Initially, the minimax problem is solved for the whole constraint set to generate the root node. The search process starts at the root node then gradually build the tree by sucessively adding nodes and descend down the tree until a feasible node is reached. This process can be repeated until a pre-defined number of iterations or run time limit is satisfied.

Algorithm 3.1 RATSAC - Random Tree Search

```
Require: Set of constraints $\mathcal{X}$, max_iter
Initialize tree with root node $\mathbf{N}_0$
$\mathcal{B}_{\mathbf{N}_0} \leftarrow \mathcal{B}_{\mathcal{X}}$
$\mathbf{N} \leftarrow \mathbf{N}_0$
while $\mathbf{N}$ is not feasbile do
    iter $\leftarrow 0$
    while iter $<$ max_iter do
        $\mathbf{N}_t \leftarrow TP(\mathbf{N})$   /*Tree Policy - Algo. (3.2)*/
        $\mathbf{R}_t \leftarrow DP(\mathbf{N}_t)$  /*Default Policy - Algo. (3.3)*/
        $BP(\mathbf{N}_t, \mathbf{R}_t)$   /*Back Propagate - Algo (3.4)*/
        iter $\leftarrow$ iter + 1
    end while
    $\mathbf{C}_\mathbf{N} \leftarrow$ set of current children of node $\mathbf{N}$
    $\mathbf{N} \leftarrow \arg\max_{\mathbf{N}_k \in \mathbf{C}_\mathbf{N}} U(\mathbf{N}_k, 0)$ /*$U$ in (3.7) */
end while
return $\mathbf{N}$.
```

3.7 Speeding up the search process

Fast descend algorithm for basic update

At a specific node, adding a child node is equivalent to removing one constraint (which belongs to the basic) and perform minimax estimation for the remaining coverage set.

One advantage of our algorithm is that the samples are close to each other (a new node is always a result of basic update from its parent node). Therefore, the fit of the parent node can be used as the initialization to compute minimax for the child node. For linear regression problem, we employ the *descend algorithm* [16, Chapter 2] to peform basic update. For problems with

Algorithm 3.2 TP-Tree Policy

```
Require: Node $\mathbf{N}$, $C_p$
while $\mathbf{N}$ is not feasbile do
    $\mathbf{C_N} \leftarrow$ set of current children of node $\mathbf{N}$
    if $\mathbf{N}$ is not fully expanded then
        for $\mathbf{x} \in \mathcal{B}_\mathbf{N}$ do
            $\mathcal{B}' \leftarrow$ basic of $[\mathcal{C}(\mathcal{B}_\mathbf{N}) \backslash \{\mathbf{x}\}]$
            Create node $\mathbf{N}_k$ with $\mathcal{B}_{\mathbf{N}_k} \leftarrow \mathcal{B}'$
            if $\mathbf{N}_k \notin \mathbf{C_N}$ then
                Add $\mathbf{N}_k$ as a child of $\mathbf{N}$
                return $\mathbf{N}_k$
            end if
        end for
    else
        $\mathbf{N} \leftarrow \arg\max_{\mathbf{N}_k \in \mathbf{C_N}} U(\mathbf{N}_k, C_p)$ /*$U$ in (3.7) */
    end if
end while
return $\mathbf{N}$.
```

Algorithm 3.3 DP - Default Policy

```
Require: Node $\mathbf{N}$
while $\mathbf{N}$ is not feasbile do
    Pick random $\mathbf{x} \in \mathcal{B}_\mathbf{N}$
    $\mathcal{B}' \leftarrow$ basic of $[\mathcal{C}(\mathcal{B}_\mathbf{N}) \backslash \{\mathbf{x}\}]$
    Create new node $\mathbf{N}_k$ with $\mathcal{B}_{\mathbf{N}_k} \leftarrow \mathcal{B}'$
    $\mathbf{N} \leftarrow \mathbf{N}_k$
end while
return $\mathbf{R_N}$.
```

Algorithm 3.4 BP - Back Propagation

```
Require: Node $\mathbf{N}$, reward $R$
while $\mathbf{N} \neq NULL$ do
    $V(\mathbf{N}) \leftarrow V(\mathbf{N}) + 1$
    $\mathbf{Q_N} \leftarrow \mathbf{Q_N} + R$
    $\mathbf{N} \leftarrow parent(\mathbf{N})$
end while
return
```

quasi-convex residuals, following [17], if $p = 1$ or $p = \infty$, each quasi-convex constraint can be converted into 4 linear contraints.

Early termination When a new basic $\mathcal{B}$ is computed for a new node during Tree Policy of Default Policy, the fitting result $\theta(\mathcal{B})$ can be used to compute the set of inliers for problem (3.1) and the inlier set with largest cardinality $\mathcal{I}'$ is kept. During the execution of the algorithm, a node $\mathbf{N}$ can stop descending down if the size its coverage is less than the current maximum number of inlier.

Initialization with ℓ_∞ outlier removal For data with high outlier rate, some heuristic algorithms can be applied to the data to remove bad outliers before executing RATSAC. Instead of initializing RATSAC with the ℓ_∞ estimation of the whole point set, the ℓ_∞ outlier removal [74] can be executed first with $\epsilon' = \kappa\epsilon$ (with $\kappa > 1$). The inliers of the ℓ_∞ estimation with the inlier threshold ϵ' is then used to initialize RATSAC. This heuristic strategy aims to remove the bad outliers which then helps RATSAC to reach a good solution faster.

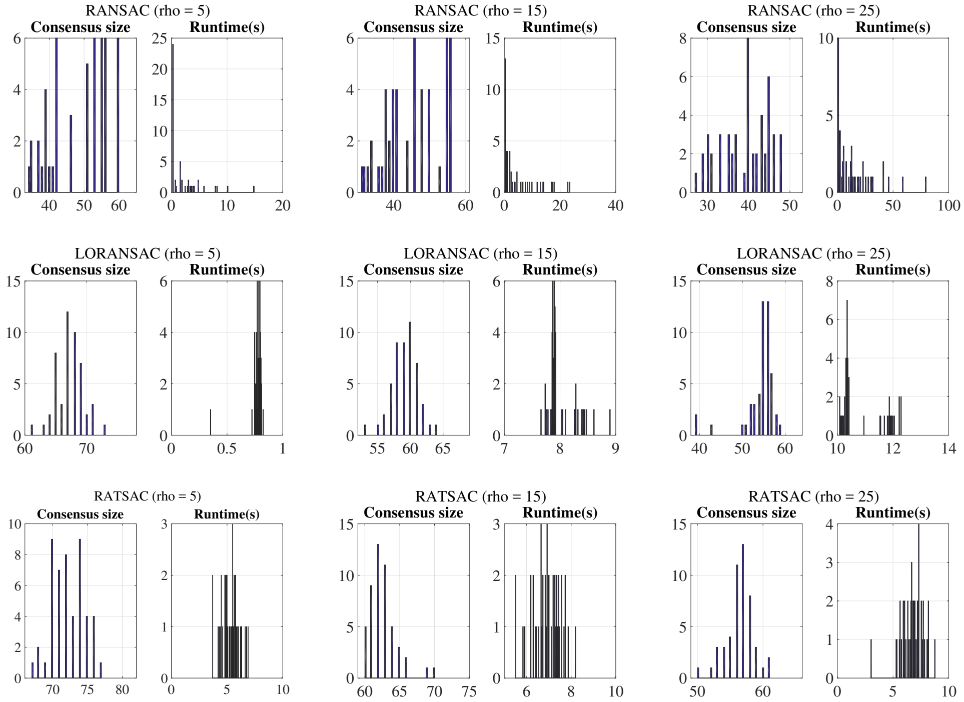


FIGURE 3.2: Histogram of consensus size and runtime for RANSAC (top row), LO-RANSAC (middle row) and RATSAC (bottom row). RATSAC outperformed RANSAC and LO-RANSAC throughout three different datasets with three outlier rates: $\rho = 5$ (fist column), $\rho = 15$ (middle column) and $\rho = 25$ (last column).

3.8 Results

The proposed algorithm (RATSAC) is tested on multiple geometric estimation problems with synthetic data as well as real data. Our method is compared against other common randomized algorithms, i.e RANSAC and its variants. In addition, some approximation algorithms are also compared. Following is the detailed list of the methods in our experiments:

- RANSAC: We use the stopping criterion of $\rho = 0.99$. To have a fair comparison, if RANSAC stops earlier than our method and the solution quality is worse than RATSAC, we increase the number of iterations to make them have approximately the same runtime.

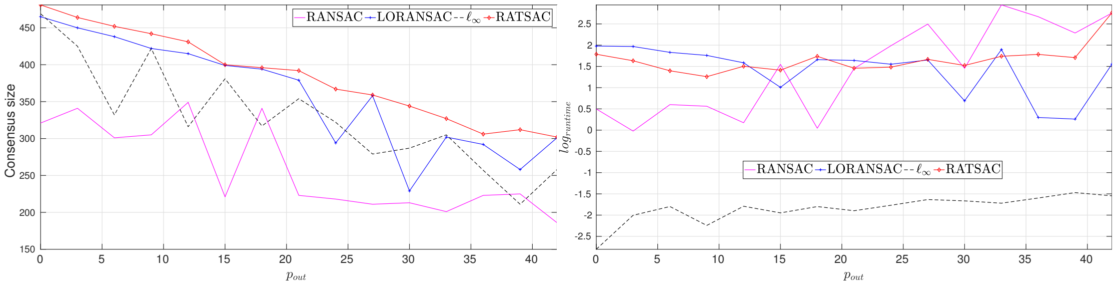


FIGURE 3.3: Experiment result for linear regression problem with $N = 500$ in $d = 8$ dimensions

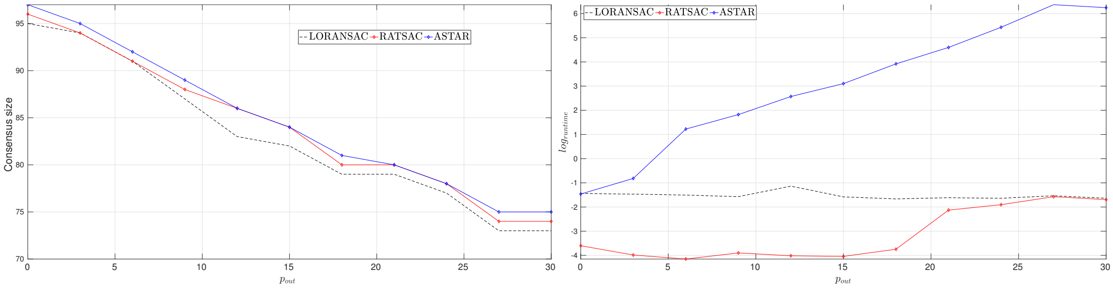


FIGURE 3.4: Experiment result for line fitting problem with $N = 100$ data points and $d = 2$

- LO-RANSAC [56]: The numer of inner sampling is set to 20, with every sample is a larger-than-minimal subset with the size of 2 times the size of the minimal subset. Like RANSAC, the stopping criterion was set to $\rho = 0.99$ throughout all experiments.

- ℓ_∞ outlier removal: Starting from the minimax solution for the whole point set, the points with largest residuals are removed until the minimax of the remaining set is less than ϵ.

- L1 approximation [74]: The maximum consensus problem is relaxed to a linear programming problem by introducing a slack variable for each constraint then minimize the ℓ_1 norm of the slack variables.

- For experiments with real data where the matching score is provided, we also compare the results with PROSAC [22].

All algorithms were implemented in C++ and run on a Ubuntu machine with 2.6GHz and 16GB of memory. We used Eigen library for all least squares and matrix computation. The source code will be made available.

3.8.1 Linear estimation

Synthetic data

	RANSAC		LO-RANSAC		PROSAC		ℓ_1 Approx.		ℓ_∞ Approx.		RATSAC	
	$\lvert\mathcal{I}\rvert$	time (s)	$\lvert\mathcal{I}\rvert$	time (s)	$\lvert\mathcal{I}\rvert$	time (s)	$\lvert\mathcal{I}\rvert$	time (s)	$\lvert\mathcal{I}\rvert$	time (s)	$\lvert\mathcal{I}\rvert$	time (s)
House (N = 556)	503	0.54	528	7.81	460	0.51	526	1.13	530	0.07	**536**	1.92
Merton (N = 590)	541	0.54	581	9.21	561	0.53	585	1.33	581	0.03	**586**	0.57
Corridor (N = 407)	393	2.53	399	7.36	391	2.67	400	0.56	395	0.03	**402**	0.49
Aerial (N = 490)	417	2.97	438	9.69	434	2.86	433	0.82	438	0.07	**439**	2.02
Hertford (N = 350)	208	0.18	301	2.87	208	0.18	305	0.32	309	0.03	**315**	1.42
Building 04 (N = 641)	301	2.2	405	5.72	391	0.86	503	1.74	493	0.15	**512**	5.78
Building 22 (N = 502)	454	0.5	**490**	11.7	456	0.49	485	0.86	482	0.06	**490**	0.96
Building 36 (N = 651)	393	0.93	557	12.48	505	0.87	554	1.98	555	0.17	**560**	5.3

TABLE 3.1: Linearised fundamental matrix estimation results

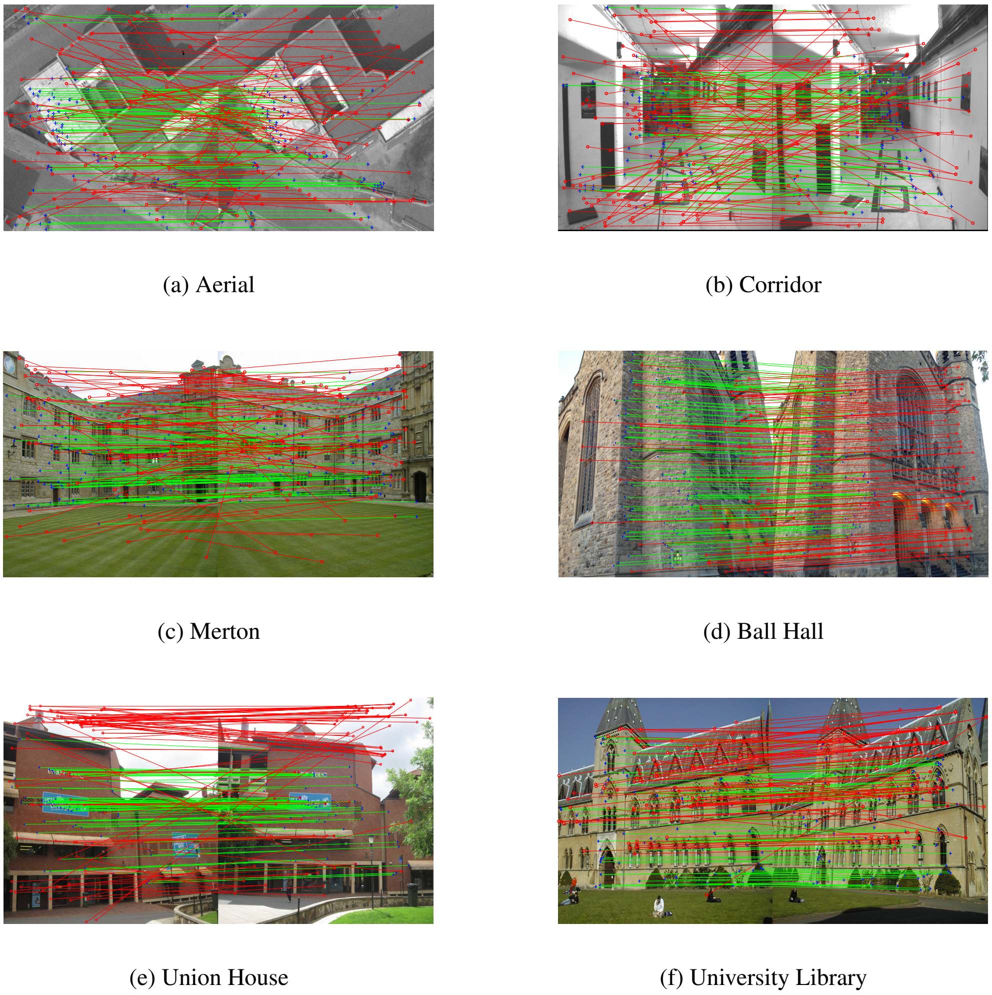
	RANSAC		LO-RANSAC		PROSAC		ℓ_1 Approx.		ℓ_∞ Approx.		RATSAC	
	$\lvert\mathcal{I}\rvert$	time (s)	$\lvert\mathcal{I}\rvert$	time (s)	$\lvert\mathcal{I}\rvert$	time (s)	$\lvert\mathcal{I}\rvert$	time (s)	$\lvert\mathcal{I}\rvert$	time (s)	$\lvert\mathcal{I}\rvert$	time (s)
University (N = 439)	312	1.0637	316	1.5794	332	1.1089	424	2.4993	425	0.080098	**429**	3.8519
Christ Church (N = 557)	492	2.6862	496	3.4523	542	2.7042	551	4.5258	543	0.037116	**554**	1.5526
Valbonne (N = 564)	334	2.6953	400	3.47	376	2.6378	433	4.8323	439	0.11894	**449**	3.822
Kapel (N = 449)	379	2.0808	392	2.6879	427	2.1101	431	2.5271	420	0.039989	**436**	6.0877
Invalides (N = 558)	482	2.5777	486	3.301	536	2.6052	546	4.5822	534	0.080517	**548**	5.8247
Union House (N = 591)	529	2.7286	551	3.6647	577	2.8466	**586**	5.1478	582	0.05316	**586**	2.8279
Classic Wing (N = 425)	398	4.0278	398	4.4749	419	4.0072	419	2.2213	418	0.039907	**423**	0.94494
Bonython (N = 507)	314	13.7554	310	14.474	343	14.8534	368	3.8895	355	0.15836	**380**	27.1775
Elder Hall (N = 406)	300	10.0547	311	11.3671	298	11.0896	318	2.016	327	0.088636	**339**	25.0695
Building 39 (N = 647)	526	3.1862	536	4.0644	607	3.2283	617	6.2035	616	0.078224	**622**	9.6927
Building 64 (N = 427)	368	2.1269	333	2.7182	402	2.1672	406	2.2919	405	0.047785	**411**	6.2971
Building 10 (N = 597)	394	3.4209	451	4.0832	462	2.9921	487	5.0714	492	0.15813	**501**	28.1173

TABLE 3.2: Homography etimation results

The synthetic data for linear estimation experiment was obtained by randomly generating a set of N data points $\mathbf{x}_i \in \mathbb{R}^d$ and an estimate $\boldsymbol{\theta} \in \mathbb{R}^d$. Then, the vector $\mathbf{y}$ is generated by setting $y_i = \mathbf{x}_i^T\boldsymbol{\theta} + n_i$ where n_i is drawn from a Gaussian distribution $\mathcal{N}(0, \sigma_{\text{in}})$. The purpose of adding n_i to each y_i is to simulate noise. Outliers were then added by randomly picking a subset $\mathcal{O}$ of $\rho\%N$ data points and purturbe y_i with $y_i = y_i + o_i$ where o_i were drawn from $\mathcal{N}(0, \sigma_{\text{out}})$.

First, our method is compared against two other commonly used randomized methods, namely RANSAC and LORANSAC in terms of solution quality and runtimes. The data was generated with $N = 100$ data points and $d = 8$ dimensions. We set $\rho = 5$ and execute each method 50 times then plot the histograms of consensus size and runtime. This experiment was repeated with two more values of $\rho = 15$ and $\rho = 25$. Figure 3.2 shows the histograms of these experiments. As expected, LO-RANSAC significantly improved the RANSAC results. RATSAC outperforms RANSAC and LO-RANSAC in terms of solution quality . In the setting with low outlier rates($\rho = 5$), RATSAC takes slightly longer time than RANSAC. As the outlier rate increases, however, RATSAC achieved the best solution quality with shorter runtime than both of its two competitors.

We generated another dataset with $N = 500$ and $d = 8$. This time, the experiment was run with $\rho = 0$ to $\rho = 45$ with the step size of 3. We add ℓ_∞ outlier removal method [74] to our comparison list. Figure (3.3) plots the consensus size and runtime (in log scale). It can be seen that RATSAC consistently outperforms other methods in term of solution quality. RANSAC and



(a) Aerial

(b) Corridor

(c) Merton

(d) Ball Hall

(e) Union House

(f) University Library

FIGURE 3.5: Qualitative results for fundamental matrix estimation (top row) and homography estimation (bottom row)

other methods are unstable in the environment with high number of outliers. ℓ_∞ can perform quite well at the at the beginning and were able to achieve solutions that are close to RATSAC, but this method failed to sustain its performance as the number of outliers increased. This result demonstrates that RATSAC can find the right path in the tree to descend and was able to reach the right solution faster than other competitors.

Line fitting To compare our method with globally optimal solution, we repeated the experiment above for a line fitting problem with 100 points in 2D. Besides the methods mentioned in the previous experiment, we also execute ASTAR search algorithm [18], which is an exact algorithm. Results are shown in figure (3.4). The number of inliers that RATSAC attains is quite close to the optimal solution while run time is much faster than LO-RANSAC. This demonstrates that RATSAC is a promising method for solving maximum consensus problem.

Fundamental matrix estimation

We selected 5 image pairs from the common VGG dataset: House, Corridor, Church, Hertford, Wadham and 5 pairs from the Zurich building dataset. The VLFeat [87] toolbox was employed for feature extraction to get around 300 to 700 correspondences for each image pair. The matching scores provided by VLFeat were also used as priors for PROSAC.

The epipolar constraints are linearized following [42, Chapter 11] and the fundamental matrices using the same set of methods. The same set of methods for linear regression experiment was executed for fundamental matrix estimation. Table (3.1) shows the results. RATSAC achived the highest number of inliers, especially much better compared with RANSAC.

3.8.2 Quasiconvex constraints

Homography Estimation

We applied the same feature extration method used in fundamental matrix estimation experiment for 5 image pairs from the VGG dataset: University Library, Christ Church, Valbonne, Kapel, Invaides; 3 pairs from the AdelaideRMF[2] dataset: Union House, Classic Wing, Bonython, Elder Hall and 3 other pairs from the Zurich building dataset. The residual function (3.4) is used to compute 2D homography transfer error for the pairs using all the methods. The results are summarised in Table (3.2). Similar to the fundamental matrix experiment, RATSAC was able to achive highest solution quality, with small increase in runtime compared to other methods.

3.9 Conclusions

We introduced a new sampling strategy for maximum consensus set problem. Unlike RANSAC and its variants, our method utilize the geometric information of the problem to search on a tree. The information from the previous samples are then used to guide the following search. Therefore, the algorithm can be regarded as a guided sampling without inlier priors. Empirical experiments have shown that our method outperforms other randomized and approximation methods and can perform well with highly contaminated data.

[2] https://cs.adelaide.edu.au/ hwong/doku.php?id=data

Chapter 4

Conformal Surface Alignment with Optimal Möbius Search

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

Huu Le, Tat-Jun Chin and David Suter: Conformal Surface Alignment With Optimal Mobius Search. In Computer Vision and Pattern Recognition (CVPR) 2016

The published paper is available at: http://ieeexplore.ieee.org/document/7780644/

Statement of Authorship

Title of Paper	Conformal Surface Alignment with Optimal Mobius Search
Publication Status	☑ Published □ Accepted for Publication □ Submitted for Publication □ Unpublished and Unsubmitted w ork w ritten in manuscript style
Publication Details	**Huu Le**, Tat-Jun Chin and David Suter: Conformal Surface Alignment With Optimal Mobius Search. In Computer Vision and Pattern Recognition (CVPR) 2016.

Principal Author

Name of Principal Author (Candidate)	Huu Le		
Contribution to the Paper	- Proposed the idea and developed the algorithm. - Implemented and tested the algorithm. - Conducted all the experiments to validate the performance of the proposed method. - Wrote the manuscript.		
Overall percentage (%)	80%		
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.		
Signature		Date	20th Dec 2017

Co-Author Contributions

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

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Tat-Jun Chin		
Contribution to the Paper	- Proposed the general research direction. - Supervised the development of this work. - Refined the manuscript.		
Signature		Date	20th Dec 2017

Name of Co-Author	David Suter		
Contribution to the Paper	- Supervised the development of this work. - Refined the manuscript.		
Signature		Date	20th Dec 2017

4.1 Introduction

Given two 3D shapes, the problem of shape correspondence is to find a meaningful relation (or mapping) between the elements of the shapes [86]. Solving such a problem is fundamental to many vision and graphics applications, such as object recognition, 3D shape retrieval, shape morphing and attribute transfer - to name a few. A plethora of variants also exist for the general problem, where the variants differ based on the type of input data (point cloud, mesh, etc.), type of alignment function, partial or full overlap, whether pre-identified landmarks are available, etc.

Within the broad literature on shape correspondence, our work belongs to the class of conformal geometric methods [92]. A conformal mapping preserves angles locally, and is thus insensitive to surface deformations [88]. It has been observed that object instances with the same intrinsic shape (e.g., faces with different expressions [88], deformable 2D shapes [14], brains between different individuals [35]) can be aligned well conformally.

The uniformization theorem [9] states that all surfaces that are topological spheres or discs can be conformally embedded to a canonical 2D domain, e.g., a unit sphere, a hyperbolic disc. In Fig. 4.1, these embeddings are represented by $\Phi_{\tilde{\mathcal{M}}}$ and $\Phi_{\tilde{\mathcal{B}}}$ for respectively two surfaces $\tilde{\mathcal{M}}$ and $\tilde{\mathcal{B}}$. The embedded surfaces can be aligned conformally by a Möbius transformation f. The direct mapping between the surfaces can then be composed as $\Phi_{\tilde{\mathcal{B}}}^{-1} \circ f \circ \Phi_{\tilde{\mathcal{M}}}$. Note that in practice, discrete analogues of the mappings are used.

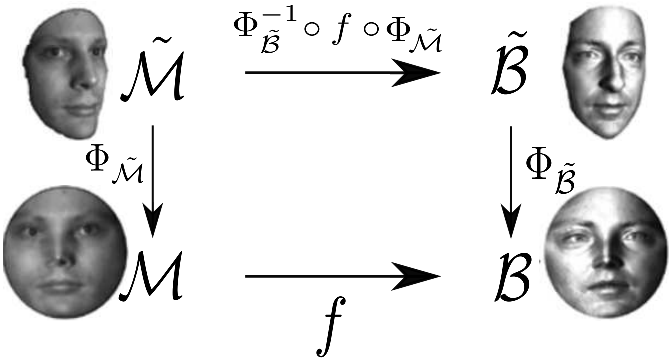

$\tilde{\mathcal{M}}$
$\Phi_{\tilde{\mathcal{B}}}^{-1} \circ f \circ \Phi_{\tilde{\mathcal{M}}}$
$\tilde{\mathcal{B}}$
$\Phi_{\tilde{\mathcal{M}}}$
$\Phi_{\tilde{\mathcal{B}}}$
$\mathcal{M}$
f
$\mathcal{B}$

FIGURE 4.1: Example of conformal mapping for surface alignment

Conformal shape correspondence thus amounts to finding $\Phi_{\tilde{\mathcal{M}}}$, $\Phi_{\tilde{\mathcal{B}}}$ and f. Many authors first calculate the embeddings $\Phi_{\tilde{\mathcal{M}}}$ and $\Phi_{\tilde{\mathcal{B}}}$ in a "flattening" step, before estimating f, e.g., [37, 35, 62, 12, 61, 52]. Various methods have been devised for flattening [7, 47, 36, 37, 81]. In this paper, we focus on the second step, i.e., Möbius alignment.

A class of existing methods for Möbius alignment are correspondence-based. Correspondences between the surfaces can be obtained by identifying landmarks (meaningful locations such as corners of eyes or tips of noses) or matching salient keypoints [37, 35, 88, 12, 65].

Given a sufficient number of correspondences, a Möbius transformation can be directly estimated [62]. The effectiveness of correspondence-based methods hinges on the veracity of the correspondences and their coverage of the surfaces.

Correspondence-free techniques as we consider in this paper must directly estimate the Möbius transformation, and implicitly the surface correspondence. Current methods include randomized search [62], iterative closest points (ICP) [10], gradient-based local optimization [52], brute force enumeration [61], and graph matching [94]. While not affected by inaccurate or insufficient pre-identified correspondences, these methods suffer, however, from other weaknesses; namely, no guarantee of success [62], dependence on good initialisation [10, 52], and very high computational cost [61, 94].

Contributions We propose a novel *globally optimal* algorithm for correspondence-free Möbius alignment. We focus on surfaces that are topological discs, i.e., f acts on the hyperbolic disc. Based on branch-and-bound (BnB) [44], our algorithm guarantees global optimality, thus obviating the need for good initializations. Further, our method is also much more efficient than previous techniques. Note that there have been previous usages of BnB for point set alignment, but these are mostly for the rigid case [13, 71, 39].

Conformally aligning surfaces that are topological discs has many practical applications, e.g., facial expression analysis [88], shape similarity analysis [61], and brain mapping [48]. Our work thus presents a useful tool to the very important area of computational conformal geometry [38].

4.1.1 Correspondence-free Möbius alignment

4.1.2 Objective function

In practice the surfaces are discrete and noisy, thus we must search for the f that is the "best" in some sense. Let $\mathcal{M} = \{\mathbf{m}_j\}_{j=1}^{N_1}$ and $\mathcal{B} = \{\mathbf{b}_k\}_{k=1}^{N_2}$ be the set of points after flattening. Recall that the points lie in $\mathbb{D}$ and are expressed as complex numbers. We adopt the objective function of [13], originally for rigid registration, to our case:

$$Q(\mathbf{z}, \theta) = \sum_j \max_k \mathbb{I}\left(|f(\mathbf{m}_j|\mathbf{z}, \theta) - \mathbf{b}_k| \leq \epsilon\right). \tag{4.1}$$

Here, $|\mathbf{x}|$ denotes the magnitude of a complex number, ϵ is a matching threshold, and $\mathbb{I}(\cdot)$ is an indicator function that returns 1 if the input statement is true, and 0 otherwise. The constant threshold ϵ can be changed to ϵ_j to make it point specific and dependent on scaling effects of the flattening on $\mathbf{m}_j$.

In words, (4.1) evaluates the number of points that are aligned under $f(\mathbf{x}|\mathbf{z},\theta)$. The inner $\max$ checks if there is a point in $\mathcal{B}$ that matches $\mathbf{m}_j$ — thus, *a priori* identified correspondences are not assumed. Further, a match is declared only if the distance between the points is within ϵ — thus, it is not expected that each point in $\mathcal{M}$ has a valid match in $\mathcal{B}$. This is crucial if the surfaces only partially overlap. Note that this objective function does not guarantee one-to-one matching, but that does not hurt the accuracy of the applications we tested.

4.1.3 Problem definition

Using (4.1), the Möbius search problem is defined as

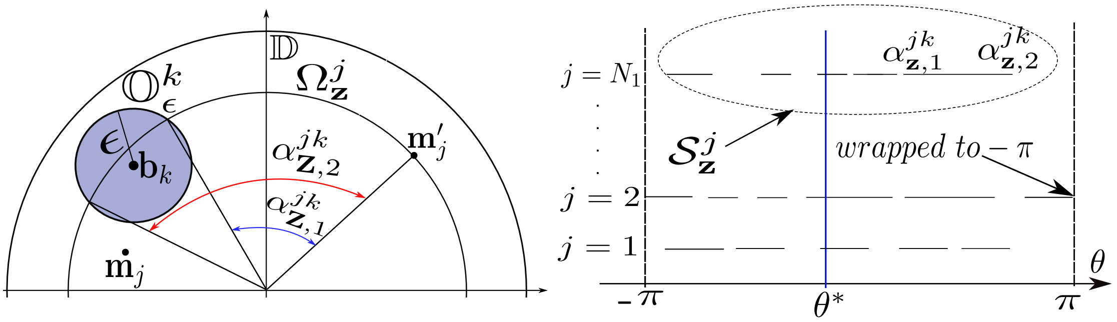
$$q^* = \max_{\mathbf{z},\theta} \; Q(\mathbf{z},\theta), \tag{4.2}$$

which equates to finding the Möbius transformation that aligns as many points from $\mathcal{M}$ with $\mathcal{B}$ as possible. The problem can be re-expressed as

$$u^* = \max_{\mathbf{z}} \; U(\mathbf{z}), \tag{4.3}$$

$$U(\mathbf{z}) = \max_{\theta} \sum_{j} \max_{k} \mathbb{I}\left(|f(\mathbf{m}_j|\mathbf{z},\theta) - \mathbf{b}_k| \leq \epsilon\right). \tag{4.4}$$

The purpose of this rearrangement is to exploit the fact that, given $\mathbf{z}$, finding θ can be done very efficiently (Sec. 4.2), such that solving for θ can seen as “evaluating” $U(\mathbf{z})$. This enables the formulation of a BnB algorithm (Sec. 6.3.2) that optimizes $\mathbf{z}$ explicitly and θ implicitly.


$\mathbb{D}$
$\Omega_{\mathbf{z}}^{j}$
$\mathbb{O}_{\epsilon}^{k}$
ϵ
$\mathbf{b}_k$
$\alpha_{\mathbf{z},2}^{jk}$
$\alpha_{\mathbf{z},1}^{jk}$
$\mathbf{m}'_j$
$\mathbf{m}_j$
$j = N_1$
$\alpha_{\mathbf{z},1}^{jk}$
$\alpha_{\mathbf{z},2}^{jk}$
$\mathcal{S}_{\mathbf{z}}^{j}$
wrapped to $-\pi$
$j = 2$
$j = 1$
θ
$-\pi$
θ^*
π

(a)

(b)

FIGURE 4.2: (a) Intersection of $\Omega_{\mathbf{z}}^{j}$ with $\mathbb{O}_{\epsilon}^{k}$. (b) Illustration of problem (4.10) .

4.2 Solving for rotation angle

Defining $\mathbf{m}'_j = h(\mathbf{m}_j|\mathbf{z})$, (4.4) can be rewritten as

$$U(\mathbf{z}) = \max_{\theta} \sum_{j} \max_{k} \mathbb{I}\left(\left|e^{i\theta}\mathbf{m}'_j - \mathbf{b}_k\right| \leq \epsilon\right). \tag{4.5}$$

Recall that $e^{i\theta}$ specifies (via Euler's equation) a rotation of the complex plane $\mathbb{C}$ about the origin. Define

$$\Omega^j_{\mathbf{z}} = \{e^{i\theta}\mathbf{m}'_j \mid \theta \in [-\pi, \pi]\} \tag{4.6}$$

as the circle resulting from rotating $\mathbf{m}'_j$ by 2π radians, and

$$\mathbb{O}^k_\epsilon = \{\mathbf{x} \mid \mathbf{x} \in \mathbb{C}, |\mathbf{x} - \mathbf{b}_k| \leq \epsilon\}. \tag{4.7}$$

as the disc centered at $\mathbf{b}_k$ of radius ϵ; see Fig. 4.2a.

Let $[\alpha^{jk}_{\mathbf{z},1}, \alpha^{jk}_{\mathbf{z},2}]$ be the range of angles θ, such that rotating $\mathbf{m}'_j$ with any θ from the range will cause the point to fall into $\mathbb{O}^k_\epsilon$. Intuitively, any $\theta \in [\alpha^{jk}_{\mathbf{z},1}, \alpha^{jk}_{\mathbf{z},2}]$ will yield

$$\mathbb{I}(|e^{i\theta}\mathbf{m}'_j - \mathbf{b}_k| \leq \epsilon) = 1. \tag{4.8}$$

The range limits $\alpha^{jk}_{\mathbf{z},1}$ and $\alpha^{jk}_{\mathbf{z},2}$ can be obtained in closed form via circle-to-circle intersections [1], see Fig. 4.2a. In the case where $\mathbb{O}^k_\epsilon$ does not intersect $\Omega^j_{\mathbf{z}}$, the range is empty, implying that no θ can cause $\mathbf{m}'_j$ to match $\mathbf{b}_k$. For details of calculating the range limits, see the supplementary material.

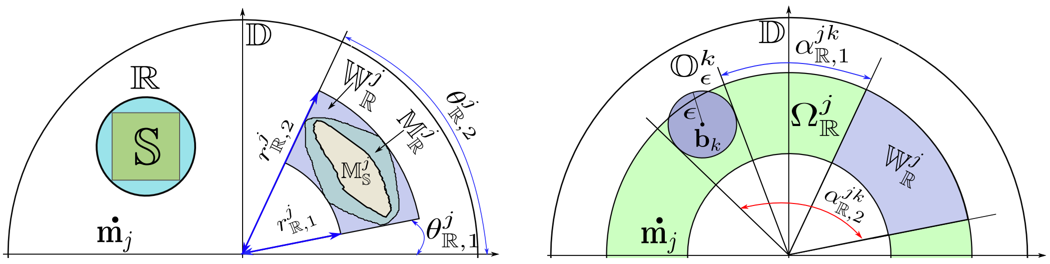
For each $\mathbf{m}'_j$, let the set of angular ranges be

$$\mathcal{S}^j_{\mathbf{z}} = \left\{[\alpha^{jk}_{\mathbf{z},1}, \alpha^{jk}_{\mathbf{z},2}]\right\}^{N_2}_{k=1}. \tag{4.9}$$

Note that overlapping ranges in $\mathcal{S}^j_{\mathbf{z}}$ are merged, while ranges that extend beyond $[-\pi, \pi]$ are "wrapped around"; see Fig. 4.2b. Function (4.5) can then be re-expressed as

$$U(\mathbf{z}) = \max_{\theta} \sum_{j} \max_{[\alpha_1,\alpha_2] \in \mathcal{S}^j_{\mathbf{z}}} \mathbb{I}(\theta \in [\alpha_1, \alpha_2]). \tag{4.10}$$

In words, evaluating $U(\mathbf{z})$ amounts to finding the θ that intersects as many as possible the angular ranges across $S^j_{\mathbf{z}}$, $j = 1, \ldots, N_1$; see Fig. 4.2b. Such a problem can be solved exactly and efficiently in $\mathcal{O}(N \log N)$ [24, Chapter 10]; see the supplementary material for the detailed algorithm.



(a) (b)

FIGURE 4.3: (a) Uncertainty area with bounding parameters. (b) Intersection of the annulus $\Omega^j_{\mathbb{R}}$ with $\mathbb{O}^k_\epsilon$.

4.3 Main algorithm

The idea of BnB is to recursively partition and prune the search space until the globally optimal solution is found. In the context of maximizing U (4.3), the search space is the hyperbolic disc $\mathbb{D}$ (as mentioned in Sec. 4.1.3, the search for θ is done implicitly). The main "design choices" are how to partition $\mathbb{D}$, and how to construct an upper bounding function $\hat{U}$ for pruning subregions of $\mathbb{D}$. Algorithm 4.2 summarizes our algorithm, and details are provided in the following.

4.3.1 Partitioning the hyperbolic disc

Algorithm 4.2 is initialized by enclosing $\mathbb{D}$ with the tightest bounding square (a subset of the complex plane $\mathbb{C}$). The square is recursively divided into four equal sub-squares (Line 13). In each sub-square $\mathbb{S}$, we attempt to update our current best solution (Line 11), or to prune $\mathbb{S}$ using the bounding function $\hat{U}$ (Line 15). A sub-square $\mathbb{S}$ that cannot be pruned is inserted into a priority queue $\mathcal{Q}$ for further processing. Note that since we actually partition the unit square that encompasses $\mathbb{D}$, a square $\mathbb{S}$ may lie outside of $\mathbb{D}$. Thus if $\mathbb{S} \cap \mathbb{D} = \emptyset$, $\mathbb{S}$ is discarded (Line 5). The above partitioning and pruning steps effectively explores the space $\mathbb{D}$. Intuitively, a tighter bounding function $\hat{U}(\mathbb{S})$ will prune more aggressively, thus leading to fewer iterations. In the following, we describe our bounding function.

4.3.2 Bounding function

Given a square region $\mathbb{S}$, we must derive an upper bounding function $\hat{U}(\mathbb{S})$ that satisfies

$$\hat{U}(\mathbb{S}) \geq \max_{\mathbf{z} \in \mathbb{S}} U(\mathbf{z}) \tag{4.11}$$

to enable pruning in Algorithm 4.2.

Algorithm 4.1 Möbius Search

```
Require: Point sets $\mathcal{M}, \mathcal{B} \subseteq \mathbb{D}$, threshold $\epsilon$
$\mathcal{Q} \leftarrow$ empty priority queue, $\mathbb{S} \leftarrow$ tightest bounding square of $\mathbb{D}$, $u^* \leftarrow 0$, $\mathbf{z}^* \leftarrow null$
Insert $\mathbb{S}$ into $\mathcal{Q}$ with priority $\hat{U}(\mathbb{S})$
while $\mathcal{Q}$ is not empty do
    Obtain a square $\mathbb{S}$ with the highest priority from $\mathcal{Q}$
    if $\mathbb{S} \cap \mathbb{D} \neq \emptyset$ then
        $\mathbf{z}_0 \leftarrow$ center point of $\mathbb{S}$
        if $U(\mathbf{z}_0) = u^*$ then
            $\mathbf{z}^* \leftarrow \mathbf{z}_0$
            return $u^*$
        else if $U(\mathbf{z}_0) > u^*$ then
            $\mathbf{z}^* \leftarrow \mathbf{z}_0$, $u^* \leftarrow U(\mathbf{z}_0)$
        end if
        Subdivide $\mathbb{S}$ into four squares $\{\mathbb{S}_l\}_{l=1}^{4}$
        for all $l = 1, \ldots, 4$ do
            if $\hat{U}(\mathbb{S}_l) > u^*$ then
                Insert $\mathbb{S}_l$ into $\mathcal{Q}$ with priority $\hat{U}(\mathbb{S}_l)$
            end if
        end for
    end if
end while
return $u^*$ and $\mathbf{z}^*$; obtain $\theta^*$ solving $U(\mathbf{z}^*)$ (4.10).
```

We begin by seeking to bound the region

$$\mathbb{M}_{\mathbb{S}}^{j} = \{h(\mathbf{m}_j|\mathbf{z}) \mid \mathbf{z} \in \mathbb{S}\} \tag{4.12}$$

arising from the uncertainty of $\mathbf{z} \in \mathbb{S}$ for each $\mathbf{m}_j$. For simplicity, we approach this via the tightest bounding disc $\mathbb{R}$ of $\mathbb{S}$ instead; see Fig. 4.3a. Clearly, since $\mathbb{S} \subseteq \mathbb{R}$, then $\mathbb{M}_{\mathbb{R}}^{j}$ is a bound over $\mathbb{M}_{\mathbb{S}}^{j}$. Now, $\mathbb{M}_{\mathbb{R}}^{j}$ can itself be bounded within a "wedge" $\mathbb{W}_{\mathbb{R}}^{j}$ defined by 4 parameters: the bounding radii

$$r_{\mathbb{R},1}^{j} = \min_{\mathbf{z} \in \mathbb{R}} |h(\mathbf{m}_j|\mathbf{z})|, \;\; r_{\mathbb{R},2}^{j} = \max_{\mathbf{z} \in \mathbb{R}} |h(\mathbf{m}_j|\mathbf{z})|, \tag{4.13}$$

and the bounding angles

$$\theta_{\mathbb{R},1}^{j} = \min_{\mathbf{z} \in \mathbb{R}} \angle h(\mathbf{m}_j|\mathbf{z}), \;\; \theta_{\mathbb{R},2}^{j} = \max_{\mathbf{z} \in \mathbb{R}} \angle h(\mathbf{m}_j|\mathbf{z}); \tag{4.14}$$

see Fig. 4.3a. Hence, to bound the region $\mathbb{M}_{\mathbb{S}}^{j}$, we determine $\mathbb{W}_{\mathbb{R}}^{j}$ that is defined by the four parameters above.

Bounding radii Based on standard identities of complex numbers [2], we observe the symmetry

$$\begin{aligned} |h(\mathbf{m}_j|\mathbf{z})| &= \left|\frac{\mathbf{m}_j - \mathbf{z}}{1 - \mathbf{m}_j\bar{\mathbf{z}}}\right| = \frac{|\mathbf{m}_j - \mathbf{z}|}{|1 - \mathbf{m}_j\bar{\mathbf{z}}|} \\ &= \frac{|-(\mathbf{m}_j - \mathbf{z})|}{|\overline{1 - \mathbf{m}_j\bar{\mathbf{z}}}|} = \frac{|\mathbf{z} - \mathbf{m}_j|}{|1 - \mathbf{z}\bar{\mathbf{m}}_j|} = |h(\mathbf{z}|\mathbf{m}_j)|. \end{aligned} \tag{4.15}$$

The bounding radii (4.13) can thus also be obtained as

$$r^j_{\mathbb{R},1} = \min_{\mathbf{z}\in\mathbb{R}} |h(\mathbf{z}|\mathbf{m}_j)|, \;\; r^j_{\mathbb{R},2} = \max_{\mathbf{z}\in\mathbb{R}} |h(\mathbf{z}|\mathbf{m}_j)|. \tag{4.16}$$

Let the center and radius of $\mathbb{R}$ be $\mathbf{c}_\mathbb{R}$ and $r_\mathbb{R}$. The range of $h(\mathbf{z}|\mathbf{m}_j)$ for all $\mathbf{z} \in \mathbb{R}$ is defined as

$$\mathbb{N}^j_\mathbb{R} = \{h(\mathbf{z}|\mathbf{m}_j) \mid \mathbf{z} \in \mathbb{R}\}. \tag{4.17}$$

Now, it is known that, if $\mathbb{R}$ is a disc, $\mathbb{N}^j_\mathbb{R}$ is also a disc [68, Chapter 3]. Further, the center and radius of $\mathbb{N}^j_\mathbb{R}$ are

$$\mathbf{c}_{\mathbb{N}^j_\mathbb{R}} = \frac{\boldsymbol{\gamma} - \mathbf{m}_j}{1 - \bar{\mathbf{m}}_j\boldsymbol{\gamma}}, \tag{4.18}$$

$$r_{\mathbb{N}^j_\mathbb{R}} = \left|\mathbf{c}_{\mathbb{N}^j_\mathbb{R}} - \frac{\mathbf{c}_\mathbb{R} + r_\mathbb{R} - \mathbf{m}_j}{1 - \bar{\mathbf{m}}_j(\mathbf{c}_\mathbb{R} + r_\mathbb{R})}\right|, \tag{4.19}$$

$$\text{where} \quad \boldsymbol{\gamma} = \mathbf{c}_\mathbb{R} - r^2_\mathbb{R}/(\overline{-1/\bar{\mathbf{m}}_j + \mathbf{c}_\mathbb{R}}). \tag{4.20}$$

Note that the region $\mathbb{M}^j_\mathbb{R}$, obtained by reversing the role of $\mathbf{z}$ and $\mathbf{m}_j$ in (4.17), is not a disc in general.

The bounding radii (4.16) can then be calculated as

$$r^j_{\mathbb{R},1} = |\mathbf{c}_{\mathbb{N}^j_\mathbb{R}}| - r_{\mathbb{N}^j_\mathbb{R}}, \quad r^j_{\mathbb{R},2} = |\mathbf{c}_{\mathbb{N}^j_\mathbb{R}}| + r_{\mathbb{N}^j_\mathbb{R}} \tag{4.21}$$

in closed form, where we offset the former to 0 if negative, and clamp the latter to 1 if greater than 1.

Bounding angles Manipulating $h(\mathbf{m}_j|\mathbf{z})$ again by

$$h(\mathbf{m}_j|\mathbf{z}) = \frac{(\mathbf{m}_j - \mathbf{z})(\overline{1 - \mathbf{m}_j\bar{\mathbf{z}}})}{(1 - \mathbf{m}_j\bar{\mathbf{z}})(\overline{1 - \mathbf{m}_j\bar{\mathbf{z}}})} = \frac{(\mathbf{m}_j - \mathbf{z})(1 - \bar{\mathbf{m}}_j\mathbf{z})}{|(1 - \mathbf{m}_j\bar{\mathbf{z}})|^2},$$

we can express $h(\mathbf{m}_j|\mathbf{z})$ as a multiplication and scaling of two complex numbers. Using the identity

$$\mathbf{ab} = |\mathbf{a}||\mathbf{b}|e^{i(\angle\mathbf{a}+\angle\mathbf{b})} \quad \forall\mathbf{a},\mathbf{b} \in \mathbb{C}, \tag{4.22}$$

of complex numbers [2], we can surmise that

$$\angle h(\mathbf{m}_j|\mathbf{z}) = \angle(\mathbf{m}_j - \mathbf{z}) + \angle(1 - \bar{\mathbf{m}}_j\mathbf{z}). \tag{4.23}$$

Define the regions

$$\mathbb{A}^j_{\mathbb{R}} = \{\mathbf{m}_j - \mathbf{z} \mid \mathbf{z} \in \mathbb{R}\}, \;\; \mathbb{B}^j_{\mathbb{R}} = \{1 - \bar{\mathbf{m}}_j\mathbf{z} \mid \mathbf{z} \in \mathbb{R}\}.$$

Clearly $\mathbb{A}^j_{\mathbb{R}}$ a disc; obtained by reflecting disc $\mathbb{R}$ and translating the result by $\mathbf{m}_j$. Its center and radius are respectively

$$\mathbf{c}_{\mathbb{A}^j_{\mathbb{R}}} = -\mathbf{c}_{\mathbb{R}} + \mathbf{m}_j, \quad r_{\mathbb{A}^j_{\mathbb{R}}} = r_{\mathbb{R}}. \tag{4.24}$$

Since multiplying two complex numbers serves to multiply their respective magnitudes (4.22), multiplying $\mathbb{R}$ with $\bar{\mathbf{m}}_i$ expands the disc by a factor of $|\mathbf{m}_j|$. Thus $\mathbb{B}^j_{\mathbb{R}}$ is also a disc with center and radius respectively

$$\mathbf{c}_{\mathbb{B}^j_{\mathbb{R}}} = -\bar{\mathbf{m}}_j\mathbf{c}_{\mathbb{R}} + 1, \quad r_{\mathbb{B}^j_{\mathbb{R}}} = |\mathbf{m}_j| r_{\mathbb{R}}. \tag{4.25}$$

The bounding angles (4.14) can then be calculated as

$$\theta^j_{\mathbb{R},1} = \min_{\mathbf{a}\in\mathbb{A}^j_{\mathbb{R}},\mathbf{b}\in\mathbb{B}^j_{\mathbb{R}}} \angle\mathbf{a} + \angle\mathbf{b}, \tag{4.26}$$

$$\theta^j_{\mathbb{R},2} = \max_{\mathbf{a}\in\mathbb{A}^j_{\mathbb{R}},\mathbf{b}\in\mathbb{B}^j_{\mathbb{R}}} \angle\mathbf{a} + \angle\mathbf{b}. \tag{4.27}$$

These values can be obtained in closed form, since the angular ranges of $\mathbb{A}^j_{\mathbb{R}}$ and $\mathbb{B}^j_{\mathbb{R}}$ are known.

Bound calculation Given the wedge $\mathbb{W}^j_{\mathbb{R}}$, we are now ready to compute the upper bound (4.11). Our strategy here is a generalization of the technique in Sec. 4.2.

First, generalizing (4.6), we define the annulus

$$\Omega^j_{\mathbb{R}} = \{e^{i\theta}\mathbf{x} \mid \mathbf{x} \in \mathbb{W}^j_{\mathbb{R}}, \theta \in [-\pi, \pi]\} \tag{4.28}$$

obtained by rotating the wedge $\mathbb{W}^j_{\mathbb{R}}$ by 2π radians; see Fig. 4.3b. Continuing the idea in Sec. 4.2, for each pair (j, k), we obtain the angular range $[\alpha^{jk}_{\mathbb{R},1}, \alpha^{jk}_{\mathbb{R},2}]$, that bounds the rotation angle θ that allows a point from $\mathbb{W}^j_{\mathbb{R}}$ to "touch" $\mathbb{O}^k_\epsilon$; see Fig. 4.3b for an intuitive example. The range limits can also be obtained in closed form; for brevity, we leave the details in the supplementary material.

Of course, if $\mathbb{O}_\epsilon^k$ does not intersect with $\Omega_\mathbb{R}^j$, then the range is empty. This implies that $f(\mathbf{m}_j|\mathbf{z},\theta)$ cannot match with $\mathbf{b}_k$ under all $\mathbf{z} \in \mathbb{S}$ and $\theta \in [-\pi,\pi]$.

Define now $\mathcal{S}_\mathbb{R}^j$ to be the set of angular intervals

$$\mathcal{S}_\mathbb{R}^j = \{[\alpha_{\mathbb{R},1}^{jk}, \alpha_{\mathbb{R},2}^{jk}]\}_{k=1}^{N_2}. \tag{4.29}$$

Again, overlapping and out-of-bound ranges are preprocessed as in Sec. 4.2. The upper bound is evaluated as

$$\hat{U}(\mathbb{S}) = \max_\theta \sum_j \max_{[\alpha_1,\alpha_2]\in\mathcal{S}_\mathbb{R}^j} \mathbb{I}(\theta \in [\alpha_1,\alpha_2]), \tag{4.30}$$

which again can be solved exactly and efficiently as a line intersection problem; cf. (4.10).

4.3.3 Algorithm convergence

Here, we establish the proofs required [44] to guarantee that Algorithm 4.2 converges to the globally optimal result.

Lemma 4.1. *$\hat{U}(\mathbb{S})$ obtained according to (4.30) satisfies (4.11).*

Proof. By design, the relationship

$$\mathbb{M}_\mathbb{S}^j \subseteq \mathbb{M}_\mathbb{R}^j \subseteq \mathbb{W}_\mathbb{R}^j \tag{4.31}$$

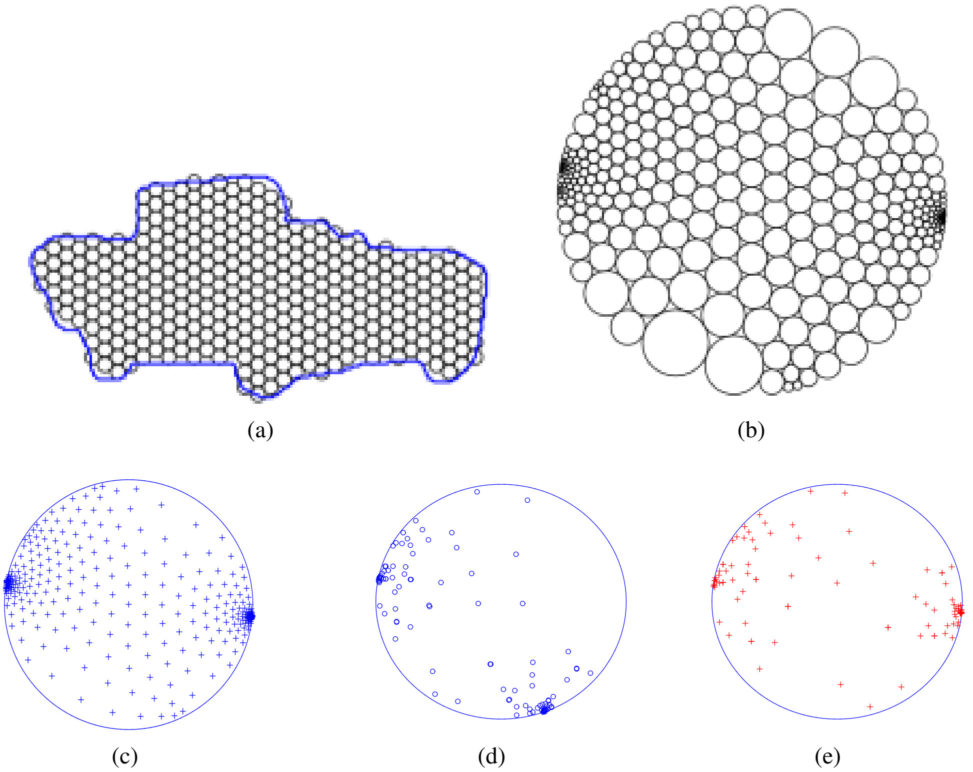
always holds. Thus, the annulus $\Omega_\mathbb{R}^j$ bounds the location of $f(\mathbf{m}_j|\mathbf{z},\theta)$ for all $\mathbf{z} \in \mathbb{S}$ and $\theta \in [-\pi,\pi]$. The angular intervals $\mathcal{S}_\mathbb{R}^j$ are also optimistic since they are constructed by aligning $\mathbb{W}_\mathbb{R}^j$ with $\mathbb{O}_\epsilon^k$ for all k. This establishes that $\hat{U}(\mathbb{S})$ cannot underestimate $U(\mathbf{z})$ for all $\mathbf{z} \in \mathbb{S}$. □ □

Lemma 4.2. *As $\mathbb{S}$ collapses to a single point $\mathbf{z}$,*

$$\hat{U}(\mathbb{S}) = U(\mathbf{z}). \tag{4.32}$$

Proof. If $\mathbb{S}$ is a single point $\mathbf{z}$, then $\mathbb{M}_\mathbb{S}^j$, defined in (4.12), equates to the singleton set $\{h(\mathbf{m}_j|\mathbf{z})\}$. Since $\mathbb{R}$ is the tightest bounding disc of $\mathbb{S}$, $\mathbb{M}_\mathbb{R}^j$ also equates to $\mathbb{M}_\mathbb{S}^j$. Now, based on definitions (4.13) and (4.14), $\mathbb{W}_\mathbb{R}^j$ also collapses to a single point $\{h(\mathbf{m}_j|\mathbf{z})\}$, thus yielding $\mathbb{M}_\mathbb{S}^j = \mathbb{M}_\mathbb{R}^j = \mathbb{W}_\mathbb{R}^j$.

The annulus $\Omega_\mathbb{R}^j$ thus becomes the circle $\Omega_\mathbf{z}^j$, and the angular ranges $\mathcal{S}_\mathbb{R}^j$ and $\mathcal{S}_\mathbf{z}^j$ are equal. Thus, $\hat{U}(\mathbb{S})$ as defined in (4.30) reduces to $U(\mathbf{z})$ as defined in (4.10). □ □



(a) (b)

(c) (d) (e)

FIGURE 4.4: Steps for generating synthetic data: (a) car-01 shape; (b) conformal map of (a); (c) centers of circles in (b); (d) $\mathcal{M}$ being sub-sampled from (c); (e) $\mathcal{B}$ was generated by applying a random Möbius transformation to $\mathcal{M}$ then added with noise and outliers.

4.4 Results

We benchmarked the performance of our algorithm (Möbius Search, henceforth represented as MS) against the previous methods surveyed in Sec. 4.1.1, namely

- Möbius voting (MV) [62]. We used the code provided by the authors [3]. However, since in this paper we focus on aligning surfaces that are topological discs (f is 3DOF), we modified the code such that a random sample consists of two randomly chosen correspondences.
- Brute force method (BF). Following [61], we implemented BF as follows: all possible pairings between $\mathcal{M}$ and $\mathcal{B}$ are considered. Each pairing is sufficient to estimate $\mathbf{z}$. For each $\mathbf{z}$, θ is enumerated across a sufficiently fine grid along $[-\pi, \pi]$ to find the best combination.
- Iterative closest points (ICP). The original method of [10] was modified as alluded in [62] for Möbius alignment.
- Graph matching (GM) [94]. We used the implementation of [57] for graph matching. Since our paper focuses on the 3DOF Möbius transform f, we included up to binary energies only in the cost function. Note, however, that this does not significantly simplify the problem, since graph matching is intractable even for binary graphs [4].

Although we did not compare against the gradient descent method of [52], as a locally convergent method, we expect its performance to be similar to ICP. Also, GM is only feasible for small input sizes N_1, N_2. In our experiments, GM was run with $N_1, N_2 \leq 20$ (in [94], input sizes of at most 15 was tested for the true graph matching part).

All experiments were run on a standard PC with 3.5GHz processor and 8GB of main memory. Due to page limits, only representative results can be shown here; see the supplementary material for more results.

4.4.1 Comparison metrics

Given a pair of conformally flattened surfaces $\mathcal{M}$ and $\mathcal{B}$, each method above was executed to estimate f. Apart from recording the runtime, we also obtained the following quality measures of the estimated f:

- Qbnb: the value of (4.1) for f.
- Qmv: the number of mutually closest pairs under f, where (j, k) is a mutually closest pair if $\mathbf{b}_k$ is the nearest neighbor of $f(\mathbf{m}_j)$ among $\mathcal{B}$ and vice versa.

Note that Qmv is as defined and used in [62, 61] for assessing deformation errors of Möbius alignment.

Where ground truth correspondences $\{\mathbf{m}_t, \mathbf{b}_t\}_{t=1}^{T}$ (from landmarks etc.) were available, we used them to calculate the following quality metric:

- Qtruth: the number of ground truth correspondences that are mutually closest pairs under f.

4.4.2 Synthetic data experiment

The purpose of this experiment is to evaluate the performance and accuracy of the methods under controlled settings. The steps to generate input point sets $\mathcal{M}$ and $\mathcal{B}$ are summarized in Fig. 4.4: first, a 2D shape from the MPEG7 dataset [54] (specifically, car-01) was chosen and conformally mapped to $\mathbb{D}$ using the circle packing technique [82]. A number of N_1 points were then randomly sampled to produce the set $\mathcal{M}$. A random Möbius transformation f was generated (by randomly choosing $\mathbf{z}$ and θ) and applied on $\mathcal{M}$ to yield the set $\mathcal{B}$. Gaussian noise of $\sigma = 0.01$ was afflicted on $\mathcal{B}$ to increase realism. Further, to simulate outliers and partially overlapping data, $\rho\%$ of points on both $\mathcal{M}$ and $\mathcal{B}$ were randomly chosen and re-sampled to lie uniformly in $\mathbb{D}$. In our experiment, we used $N_1 = \{100, 50, 20\}$ and $\rho = \{0, 25, 50\}$. For MS and (4.1), ϵ =0.01 was used. Again, note that GM is only feasible for $N_1 = 20$.

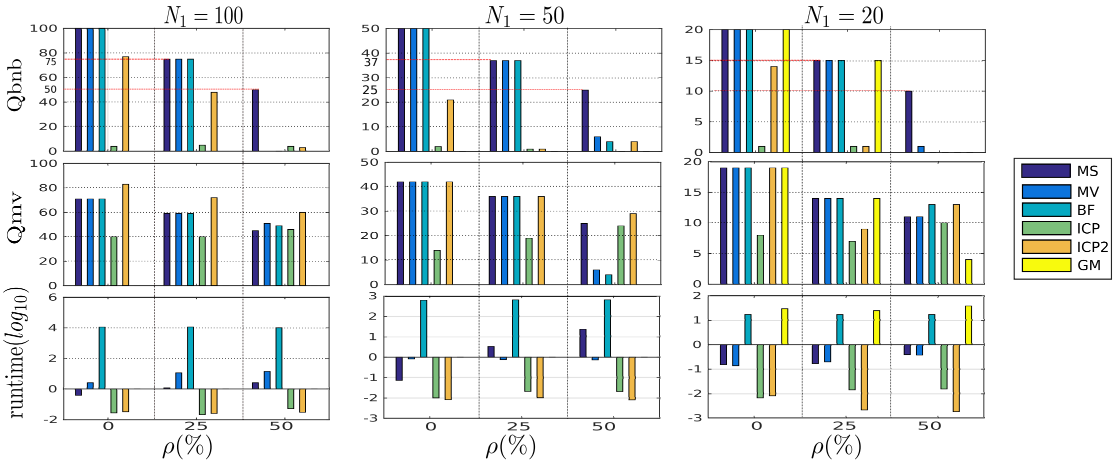
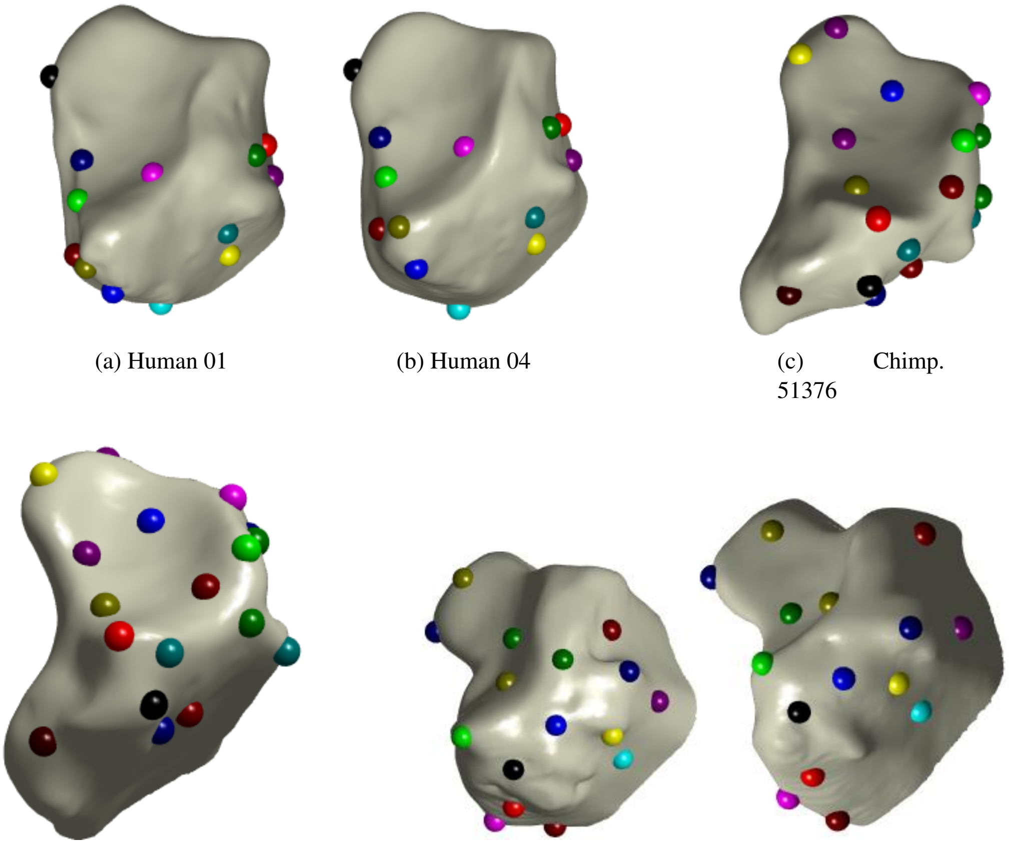


FIGURE 4.5: Results for synthetic data. Columns represent experiments for different values of N_1. Rows represent measurements of Qbnb (defined in (4.1)), Qmv (defined in 4.4.1) and runtime (in seconds). Note that the runtime is quoted in log_{10} scale. ICP2 is explained in 4.4.2. GM is only feasible for $N_1 = 20$.



(a) Human 01

(b) Human 04

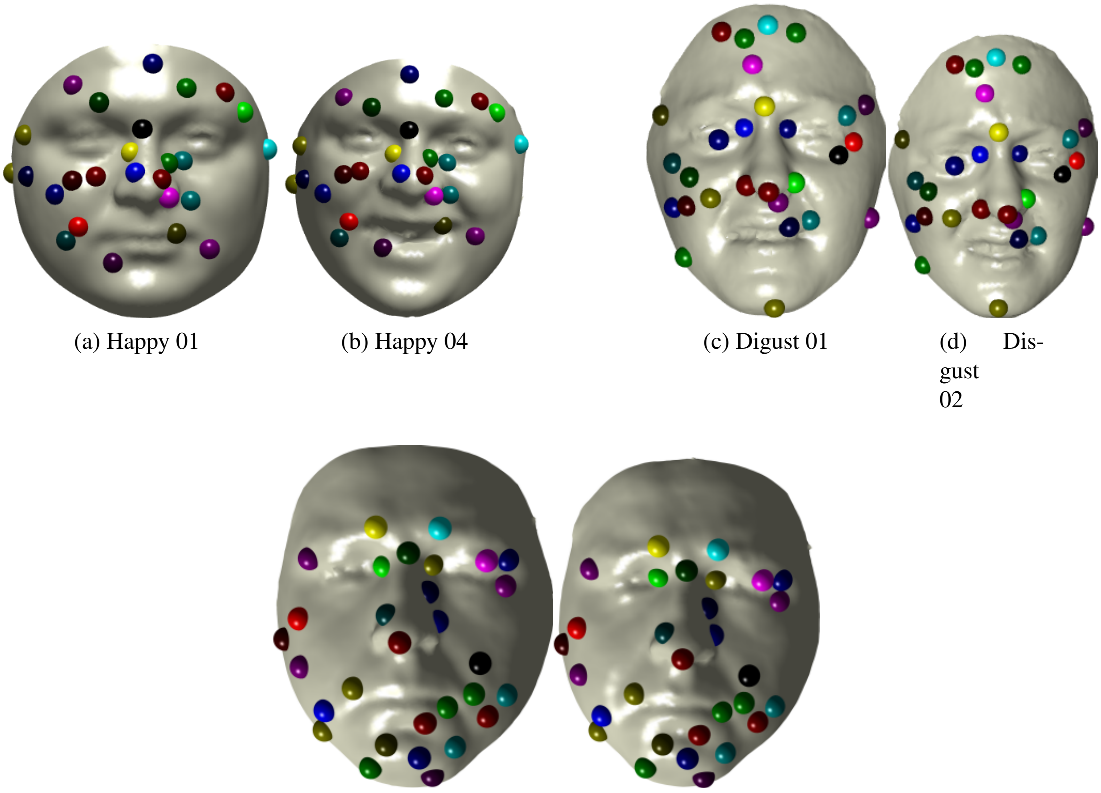
(c) Chimp. 51376

(d) Chimp.51379

(e) Gorilla 167335

(f) Gorilla 167336

FIGURE 4.6: Correspondences found by MS for three pairs of teeth.



(a) Happy 01

(b) Happy 04

(c) Digust 01

(d) Disgust 02

(e) Sad 01

(f) Sad 02

FIGURE 4.7: Correspondences found by MS between three pairs of face

Fig. 4.5 shows the results. It can be seen that MS always achieved the theoretical maximum $(1 - \rho)N_1$ of the objective function (4.1). In the presence of low oulier rates (ρ is 0% or 25%), the estimated f of MS, MV, BF and GM were of similar quality. When the outlier percentage ρ was 50%, however, only MS could produce good (in fact, optimal) results. Note that in Fig. 4.5 the quality of ICP was much lower than the others due to the lack of good initializations

To accommodate ICP, the experiment was repeated by producing a “milder” randomized Möbius transformation that relates $\mathcal{M}$ to $\mathcal{B}$, specifically by choosing the parameters such that $|\mathbf{z}| \leq 0.1$ (close to the center of $\mathbb{D}$) and $\theta \leq 10°$. The results for this repeated experiment were displayed as ICP2 in Fig. 4.5. It can easily be seen that with good initialization, ICP gave acceptable quality when there were no outliers - however, the quality degraded rapidly as the outlier rate increased. Interestingly, the runtime of MS increased marginally when the true f was close to the identity mapping - possibly because as the true $\mathbf{z}$ is closer to the center of $\mathbb{D}$, a deeper search must be conducted to before a good suboptimal solution $\mathbf{z}_0$ is retrieved to enable effective pruning.

In terms of runtime, MS and MV could terminate well within 1 minute, though MS occasionally took longer than MV. Note that MS provides guaranteed global optimality, unlike MV. The runtime for BF, as anticipated, was too long to be practical, e.g., more than 3 hours for $N = 100$. To view the actual numerical values of the results above, see the supplementary material.

4.4.3 Conformal teeth alignment experiment

We followed the experiment by Boyer et al. [12] to perform surface alignment on 3D scans of teeth. While the original aim of Boyer et al. was shape comparison, here, we focus on the alignment step. In our experiment, we chose three pairs of teeth originally used in [61], specifically, Human01 and Human04, Chimpanzee51376 and Chimpanzee51379, and Gorilla167335 and Gorilla16736. The meshes of the teeth were flattened to $\mathbb{D}$ using the program of Lipman et al. [61]. On the flattened surfaces, we also conducted the sampling process of [62, 12] to create the point sets $\mathcal{M}$ and $\mathcal{B}$. Specifically, N_1 and N_2 points ($N_1, N_2 = \{100, 50, 20\}$) were chosen using the farthest point sampling (FPS) algorithm [25]. Note that 13 ground truth correspondences $\{\mathbf{m}_t, \mathbf{b}_t\}_{t=1}^{13}$ (manually annotated landmarks) were available per problem instance, thus Qtruth value could be obtained for each method.

Table 4.2 summarizes the quantitative results, while Fig. 4.6 shows qualitative results for MS. As expected, due to the global optimality guarantees, MS returned the solution with the highest Qbnb value. Also, MS demonstrated typically superior accuracy in terms of Qtruth, as compared to BF and MV. However, when the input size was small ($N_1, N_2 = 20$), none of the methods were able to obtain satisfactory Qtruth values. This was due to the overly impoverished structural information after excessive sampling. Due to the lack of good initializations (the intialized state of $\mathcal{M}$ and $\mathcal{B}$ depends on the implementation of the conformal flattening procedure), ICP generally could not find good estimates of f, and it was able to align about half of the ground truth correspondences. While GM was feasible on $N_1, N_2 = 20$, it is apparent that the estimated f was far from ideal due to the overly sparse input data.

In terms of runtime, all the methods except BF were able to terminate in about or less than 1 minute.

4.4.4 Conformal face alignment experiment

The previous experiment was repeated for conformal face alignment, following [93]. While the previous works aimed at applications such as facial expression recognition, in our experiment, we focused on the task of Möbius alignment. From a practical standpoint, our MS algorithm can be used to automatically and deterministically find landmark correspondences, which is a crucial step in facial processing applications [93].

Again, following the previous works, we used data from the BU-3DFE face dataset [91], specifically, we chose three pairs of faces with the same expression but at different degrees:

		N_1, N_2	Methods	Qbnb	Qmv	Qtruth	Time (sec)
Human01	Human04	100	MS	**40**	80	**13**	45.056
			BF	30	**84**	**13**	10185.000
			MV	35	81	**13**	29.160
			ICP	22	76	6	**0.027**
		50	MS	**8**	38	**13**	8.780
			BF	3	**41**	12	629.000
			MV	3	39	5	6.891
			ICP	4	37	6	**0.009**
		20	MS	**11**	**18**	11	0.809
			BF	6	**18**	8	17.265
			MV	**11**	**18**	**12**	0.408
			GM	5	11	3	26.280
			ICP	3	16	7	**0.005**
Gorilla167335	Gorilla167336	100	MS	**68**	**83**	**12**	12.370
			BF	66	81	**12**	10659.000
			MV	61	80	**12**	5.651
			ICP	8	78	6	**0.052**
		50	MS	**26**	**39**	6	3.234
			BF	22	38	5	634.000
			MV	24	**39**	6	5.970
			ICP	1	36	**7**	**0.005**
		20	MS	**11**	**16**	5	0.180
			MV	10	15	4	0.347
			BF	10	**16**	**6**	17.208
			GM	8	10	1	77.513
			ICP	0	14	**6**	**0.001**
Chimpanzee51376	Chimpanzee51379	100	MS	**54**	82	**12**	26.913
			BF	52	**85**	**12**	10365.000
			MV	47	84	**12**	28.045
			ICP	22	48	1	**0.027**
		50	MS	**25**	**42**	**13**	1.774
			BF	19	**42**	8	691.000
			MV	20	38	8	3.684
			ICP	7	22	3	**0.005**
		20	MS	**13**	15	0	0.857
			BF	7	**17**	**4**	17.056
			MV	7	12	2	0.260
			GM	6	10	0	47.236
			ICP	4	9	1	**0.068**

TABLE 4.1: Results from conformally aligning three pairs of teeth. In each problem instance, the best quality measure and runtime obtained among all the methods are bolded. MS: Möbius Search, MV: Möbius voting, BF: brute force, GM: graph matching. See Sec. 4.4.1 for definitions of the quality measures.

		N_1, N_2	Methods	Qbnb	Qmv	Qtruth	Time (sec)
F0001_DI01WH_F3D	F0001_DI02WH_F3D	100	MS	**75**	81	**13**	1.999
			BF	71	**85**	**13**	10656.000
			MV	74	66	7	2.853
			ICP	46	61	1	**0.021**
		50	MS	**22**	37	**13**	1.792
			BF	19	**41**	**13**	635.000
			MV	18	40	**13**	1.870
			ICP	12	26	1	**0.005**
		20	MS	**10**	15	**9**	0.220
			BF	6	14	8	17.241
			MV	6	14	7	0.278
			GM	8	**16**	**9**	45.059
			ICP	3	11	0	**0.003**
M0044_HA01IN_F3D	M0044_HA04IN_F3D	100	MS	**53**	83	**13**	15.768
			BF	34	**84**	**13**	10269.000
			MV	40	80	12	28.648
			ICP	18	61	1	**0.022**
		50	MS	**27**	42	**13**	4.109
			BF	25	**44**	**13**	641.041
			MV	23	**44**	**13**	1.549
			ICP	3	32	1	**0.010**
		20	MS	**16**	**19**	**13**	0.373
			BF	14	**19**	**13**	17.544
			MV	**16**	**19**	**13**	0.286
			GM	7	12	9	55.105
			ICP	11	14	0	**0.002**
M0021_SA01WH_F3D	M0021_SA02WH_F3D	100	MS	**45**	85	**13**	15.081
			BF	42	**88**	**13**	9695.200
			MV	28	56	0	56.793
			ICP	32	63	0	**0.028**
		50	MS	**24**	45	**13**	14.474
			BF	23	**46**	**13**	623.966
			MV	3	19	4	1.526
			ICP	3	34	0	**0.006**
		20	MS	**14**	14	**13**	0.231
			BF	**14**	**15**	**13**	17.445
			MV	**14**	14	**13**	0.140
			GM	4	13	9	69.509
			ICP	5	12	0	**0.006**

TABLE 4.2: Results from conformally aligning three pairs of faces. In each problem instance, the best quality measure and runtime obtained among all the methods are bolded. MS: Möbius Search, MV: Möbius voting, BF: brute force, GM: graph matching. See Sec. 4.4.1 for definitions of the quality measures.

Happy 01-Happy 04, Disgust 01-Disgust 02, Sad 01 - Sad 02. The same steps as in Sec. 4.4.3 were used for flattening and subsampling; see Fig. 4.7 for the resulting data. For this dataset, since the ground truth landmarks were not available, we manually annotated 13 landmarks on the faces to create ground truth correspondences. Table 4.2 summarizes the quantitative results, while Fig. 4.7 illustrates qualitative results of MS. It can be seen that MS generally outperformed the other methods in terms of both accuracy and runtime.

4.5 Conclusions and future work

We proposed a novel approach for conformal surface alignment with guaranteed global optimum. Our experiments showed that this algorithm is much more efficient than state-of-the-art techniques for conformally aligning topological disc surfaces.

This work opens up a new direction for further research on global optimization methods in the field of computational conformal geometry. One notable expansion which can be studied in the future is 6DOF Möbius search for genus zero surfaces with spherical topology.

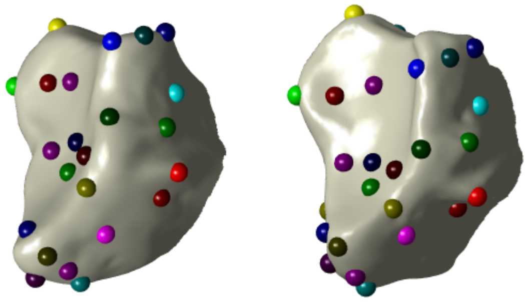
4.6 Supplementary Materials

In Sec. 4.6.1 and 4.6.2, we provide more results on real data obtained by Möbius Search(MS) and the competitors: Möbius voting (MS), brute force (BF), graph matching (GM) and ICP. In Sec. A.3, we display the numerical values used to plot the bar charts in Fig. 4 (synthetic data results) of the main paper.

4.6.1 Conformal Teeth Alignment

We chose 5 pairs of teeth and repeat the experiments described in Sec. 5.3 in the main paper. For each pair, we show the qualitative results followed by a table displaying the quantitative results.

4.6.1.1 Human09 - Human11



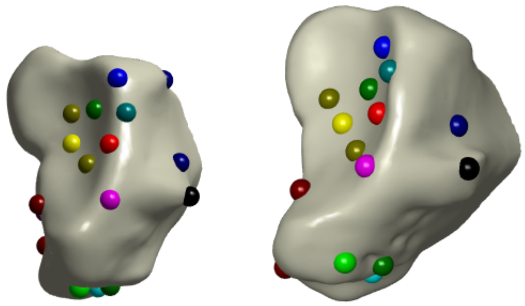
(a) (b)

FIGURE 4.8: Correspondences found by MS for Human09 - Human11

		N_1, N_2	Methods	Qbnb	Qmv	Qtruth	Time (sec)
Human09	Human11	100	MS	**40**	**81**	**13**	38.87866
			BF	32	**81**	6	10569.651
			MV	24	60	3	11.18483
			ICP	19	73	7	**0.01788**
		50	MS	**25**	43	**13**	3.16331
			BF	22	**45**	**13**	647.391
			MV	19	30	2	2.17476
			ICP	16	42	12	**0.00811**
		20	MS	**9**	16	**13**	0.36952
			BF	1	**17**	8	17.19584
			MV	3	12	3	0.40676
			GM	2	12	1	39.456
			ICP	1	16	10	**0.00191**

TABLE 4.3: Results for conformal alignment of Human09 and Human11

4.6.1.2 Orangutan 505958 - Orangutan 50960



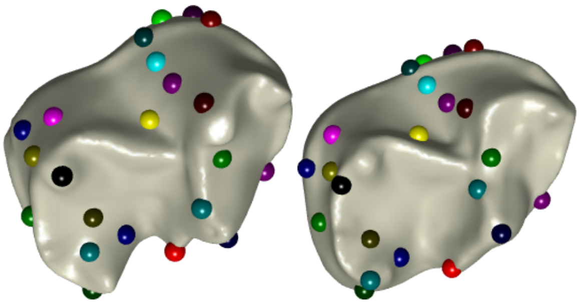
(a) (b)

FIGURE 4.9: Correspondences found by MS for Orangutan 505958 - Orangutan 50960

		N_1, N_2	Methods	Qbnb	Qmv	Qtruth	Time (sec)
Orangutan50958	Orangutan50960	100	MS	**40**	76	7	35.966
			BF	29	**78**	**12**	10049.000
			MV	26	61	2	11.283
			ICP	18	46	1	**0.022**
		50	MS	**20**	34	3	2.808
			BF	15	**37**	**12**	649.365
			MV	11	32	1	2.925
			ICP	8	32	1	**0.008**
		20	MS	**9**	12	2	0.297
			BF	2	**17**	4	16.886
			MV	3	11	1	0.541
			GM	2	16	**7**	52.207
			ICP	3	10	2	**0.002**

TABLE 4.4: Results for conformal alignment of Orangutan 505958 and Orangutan 50960

4.6.1.3 V01 - V02



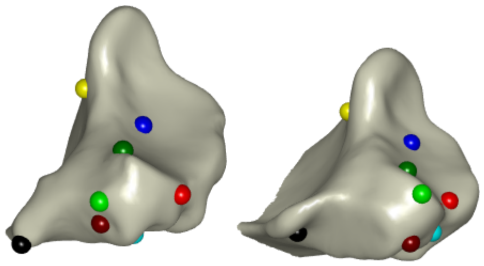
(a) (b)

FIGURE 4.10: Correspondences found by MS for V01 - V02

		N_1, N_2	Methods	Qbnb	Qmv	Qtruth	Time (sec)
V01	V02	100	MS	**34**	**81**	**15**	14.691
			BF	25	78	12	10011.000
			MV	14	51	3	2.818
			ICP	21	67	0	**0.023**
		50	MS	**16**	**44**	**13**	2.816
			BF	7	**44**	12	639.578
			MV	5	32	6	3.639
			ICP	8	38	0	**0.006**
		20	MS	**8**	**19**	**15**	0.524
			BF	4	**19**	14	17.039
			MV	0	14	5	1.211
			GM	3	7	1	59.338
			ICP	6	15	1	**0.002**

TABLE 4.5: Results for conformal alignment of V01 and V02

4.6.1.4 Bonobo 38018 - Bonobo 38019



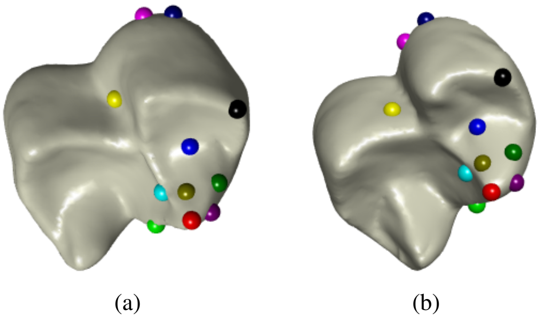
(a) (b)

FIGURE 4.11: Correspondences found by MS for Bonobo 38018 - Bonobo 38019

		N_1, N_2	Methods	Qbnb	Qmv	Qtruth	Time (sec)
Bonobo 38018	Bonobo38019	100	MS	**70**	62	**10**	10.971
			BF	45	**68**	0	10025.551
			MV	60	58	6	11.342
			ICP	28	27	1	**0.020**
		50	MS	**17**	33	**5**	6.509
			BF	7	**34**	**5**	633.254
			MV	1	29	2	6.602
			ICP	2	28	2	**0.028**
		20	MS	**5**	13	2	0.624
			BF	0	**15**	**6**	17.550
			MV	2	8	0	0.993
			GM	**5**	7	2	0.001
			ICP	3	10	2	**0.002**

TABLE 4.6: Results for conformal alignment of Bonobo38018 and Bonobo38019

4.6.1.5 x03 - x04



(a) (b)

FIGURE 4.12: Correspondences found by MS for x03-x04

		N_1, N_2	Methods	Qbnb	Qmv	Qtruth	Time (sec)
x03	x04	100	MS	**16**	**78**	**14**	30.074
			BF	3	**78**	**14**	10903.700
			MV	7	66	0	42.472
			ICP	5	73	2	**0.021**
		50	MS	**21**	**43**	**16**	4.207
			BF	16	**43**	15	621.235
			MV	12	27	2	2.181
			ICP	9	34	2	**0.005**
		20	MS	**8**	15	**13**	0.638
			BF	3	16	0	18.320
			MV	2	13	1	0.336
			GM	0	14	0	52.530
			ICP	3	**17**	4	**0.002**

TABLE 4.7: Results for conformal alignment of x03 and x04

4.6.2 Conformal Face Alignment

Similar to 4.6.1, the experiments are repeated for 5 more pairs of face.

4.6.2.1 F0015_FE01WH - F0015_FE02WH

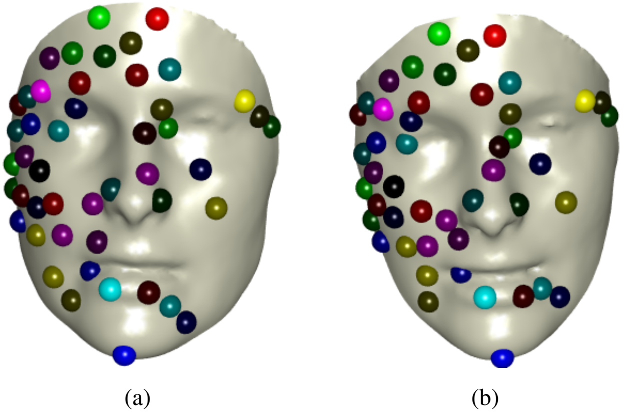
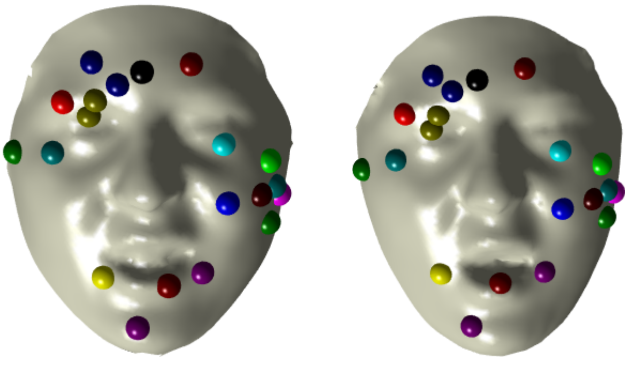

(a) (b)

FIGURE 4.13: Correspondences found by MS for F0015_FE01WH - F0015_FE02WH

		N_1, N_2	Methods	Qbnb	Qmv	Qtruth	Time (sec)
F0015_FE01WH	F0015_FE02WH	100	MS	**54**	82	**13**	15.661
			BF	42	**84**	**13**	9982.935
			MV	36	28	1	16.914
			ICP	30	58	0	**0.083**
		50	MS	**24**	40	**13**	4.088
			BF	12	**41**	1	612.245
			MV	7	35	11	1.495
			ICP	10	29	0	**0.005**
		20	MS	**14**	17	**13**	0.710
			BF	10	**18**	11	16.798
			MV	2	10	6	0.138
			GM	12	**18**	12	44.208
			ICP	4	13	0	**0.004**

TABLE 4.8: Results for conformal alignment of F0015_FE01WH - F0015_FE02WH

4.6.2.2 F0049_SU01WH - F0049_SU03WH



(a)

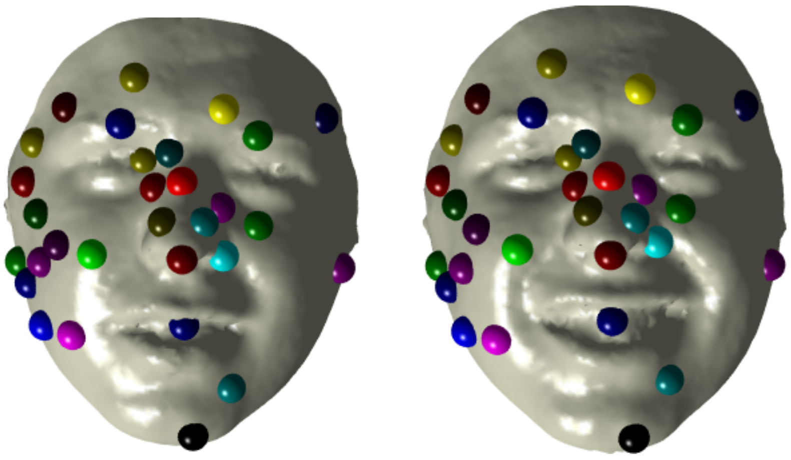
(b)

FIGURE 4.14: Correspondences found by MS for F0049_SU01WH - F0049_SU03WH

		N_1, N_2	Methods	Qbnb	Qmv	Qtruth	Time (sec)
F0049_SU01WH	F0049_SU03WH	100	MS	**51**	84	**13**	15.581
			BF	46	**85**	**13**	10011.218
			MV	38	24	0	28.224
			ICP	36	67	0	**0.023**
		50	MS	**29**	44	**13**	4.249
			BF	23	**47**	**13**	635.322
			MV	15	8	1	2.232
			ICP	7	33	0	**0.007**
		20	MS	**13**	**19**	**13**	0.378
			BF	12	**19**	**13**	16.992
			MV	3	7	2	0.693
			GM	2	16	8	66.848
			ICP	2	14	0	**0.003**

TABLE 4.9: Results for conformal alignment of F0049_SU01WH and F0049_SU03WH

4.6.2.3 M0015_HA02WH - M0015_HA04WH



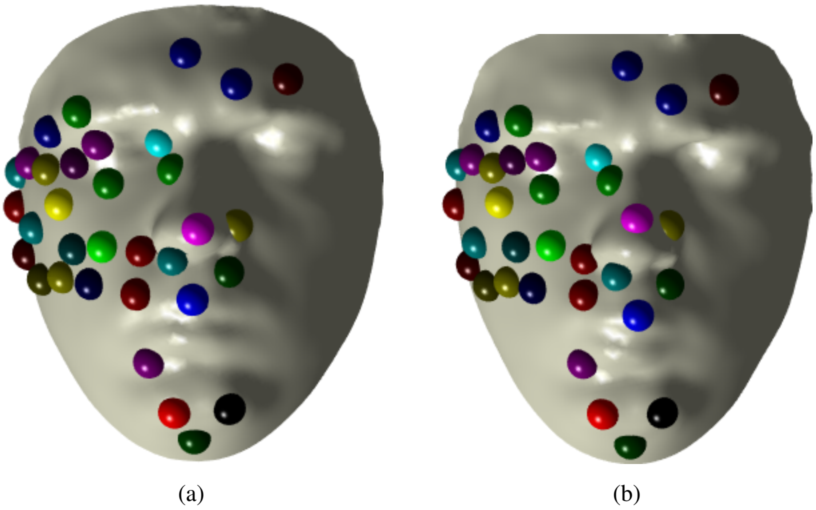
(a) (b)

FIGURE 4.15: Correspondences found by MS for M0015_HA02WH - M0015_HA04WH

		N_1, N_2	Methods	Qbnb	Qmv	Qtruth	Time (sec)
M0015_HA02WH	M0015_HA04WH	100	MS	**34**	80	**13**	16.143
			BF	28	**84**	**13**	11992.847
			MV	24	26	1	6.527
			ICP	21	79	12	**0.045**
		50	MS	**17**	42	**13**	10.316
			BF	11	**44**	**13**	648.235
			MV	6	11	2	1.506
			ICP	10	38	**13**	**0.026**
		20	MS	**10**	15	**13**	0.523
			BF	1	**17**	5	18.740
			MV	3	15	**13**	0.326
			GM	3	14	8	86.483
			ICP	6	15	**13**	**0.019**

TABLE 4.10: Results for conformal alignment of M0015_HA02WH and M0015_HA04WH

4.6.2.4 M0040_SA02WH - M0040_SA04WH



(a)

(b)

FIGURE 4.16: Correspondences found by MS for M0040_SA02WH - M0040_SA04WH

		N_1, N_2	Methods	Qbnb	Qmv	Qtruth	Time (sec)
M0040_SA02WH	M0040_SA04WH	100	MS	**38**	84	12	14.568
			BF	30	**85**	**13**	10985.872
			MV	23	11	1	14.105
			ICP	18	79	11	**0.232**
		50	MS	**17**	43	12	14.977
			BF	14	**45**	**13**	629.110
			MV	4	17	2	0.767
			ICP	1	42	10	**0.009**
		20	MS	**12**	18	**13**	0.914
			BF	4	**19**	12	16.956
			MV	2	17	12	0.137
			GM	3	17	**13**	29.429
			ICP	3	16	12	**0.002**

TABLE 4.11: Results for conformal alignment of M0040_SA02WH and M0040_SA04WH

4.6.2.5 F0036_AN02AE - F0036_AN02AE

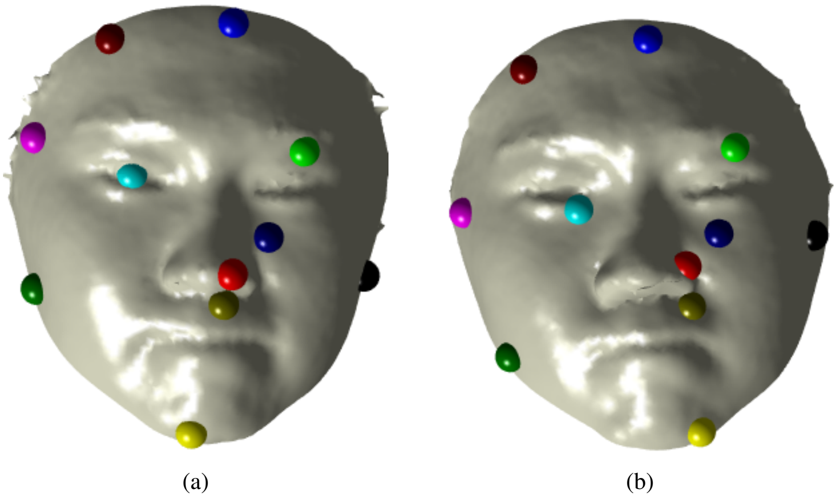


FIGURE 4.17: Correspondences found by MS for F0036_AN02AE - F0036_AN02AE

		N_1, N_2	Methods	Qbnb	Qmv	Qtruth	Time (sec)
F0036_AN02AE	F0036_AN04AE	100	MS	**24**	77	**11**	54.420
			BF	19	**84**	0	10565.549
			MV	10	43	1	2.822
			ICP	8	68	1	**0.024**
		50	MS	**19**	40	6	4.336
			BF	14	**42**	**11**	631.845
			MV	8	8	1	1.484
			ICP	12	37	1	**0.007**
		20	MS	**10**	15	**13**	0.526
			BF	1	**17**	1	16.937
			MV	2	9	2	0.187
			GM	1	14	11	36.986
			ICP	0	14	1	**0.002**

TABLE 4.12: Results for conformal alignment of F0036_AN02AE and F0036_AN02AE

4.6.3 Numerical results for synthetic data

N_1	ρ (%)	Methods	Qbnb	Qmv	Time(sec)
100	0	MS	**100**	71	0.385
		MV	**100**	71	2.557
		BF	**100**	71	11304.600
		ICP	4	40	**0.028**
		ICP2	77	**83**	0.033
	25	MS	**75**	59	1.130
		MV	**75**	59	11.435
		BF	**75**	59	11404.700
		ICP	5	40	**0.021**
		ICP2	48	**72**	0.025
	50	MS	**50**	45	2.569
		MV	0	51	14.141
		BF	0	49	10121.000
		ICP	4	46	0.052
		ICP2	3	**60**	**0.030**

TABLE 4.13: Numerical results for synthetic data with $N_1 = 100$

N_1	ρ (%)	Methods	Qbnb	Qmv	Time(sec)
50	0	MS	**50**	**42**	0.072
		MV	**50**	**42**	0.815
		BF	**50**	**42**	618.759
		ICP	2	14	0.010
		ICP2	21	**42**	**0.008**
	25	MS	**37**	**36**	3.339
		MV	**37**	**36**	0.755
		BF	**37**	**36**	642.490
		ICP	1	19	0.021
		ICP2	1	**36**	**0.010**
	50	MS	**25**	25	23.175
		MV	6	6	0.728
		BF	4	4	643.603
		ICP	0	24	0.021
		ICP2	4	**29**	**0.008**

TABLE 4.14: Numerical results for synthetic data with $N_1 = 50$

N_1	ρ (%)	Methods	Qbnb	Qmv	Time(sec)
20	0	MS	**20**	**19**	0.15343
		MV	**20**	**19**	0.1365
		BF	**20**	**19**	17.07308
		ICP	1	8	**0.00664**
		ICP2	14	**19**	0.00808
		GM	**20**	**19**	29.281
	25	MS	**15**	**14**	0.16772
		MV	**15**	**14**	0.1965
		BF	**15**	**14**	16.86983
		ICP	1	7	0.01413
		ICP2	1	9	**0.00213**
		GM	**15**	**14**	24.4612
	50	MS	**10**	11	0.38881
		MV	1	11	0.3723
		BF	0	**13**	17.02
		ICP	0	10	0.01533
		ICP2	0	**13**	**0.00182**
		GM	0	4	37.9

TABLE 4.15: Numerical results for synthetic data with $N_1 = 20$

4.6.4 Calculating range limit

4.6.4.1 Range limit for solving rotation angle

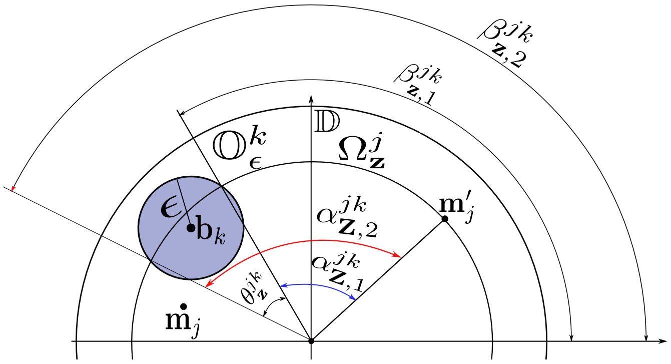

$\beta^{jk}_{\mathbf{z},2}$
$\beta^{jk}_{\mathbf{z},1}$
$\mathbb{D}$
$\mathbb{O}^k_\epsilon$
$\Omega^j_{\mathbf{z}}$
ϵ
$\mathbf{b}_k$
$\alpha^{jk}_{\mathbf{z},2}$
$\mathbf{m}'_j$
$\alpha^{jk}_{\mathbf{z},1}$
$\theta^{jk}_{\mathbf{z}}$
$\mathbf{m}_j$

FIGURE 4.18: Computing range limits for solving rotation angle

This section explains how the range limit $[\alpha^{jk}_{\mathbf{z},1}, \alpha^{jk}_{\mathbf{z},2}]$ defined in section 3 can be derived.

Let $\theta^{jk}_{\mathbf{z}}$ be the intersection angle between $\Omega^j_{\mathbf{z}}$ and $\mathbb{O}^k_\epsilon$ as depicted in Fig. 4.18. This angle can be evaluated easily using circle to circle intersection.

Define $\beta^{jk}_{\mathbf{z},1}$ and $\beta^{jk}_{\mathbf{z},2}$ to be the limiting angles of the intersection arc, which can be determined by:

$$\beta^{jk}_{\mathbf{z},1} = \angle \mathbf{b}_k - \frac{\theta^{jk}_{\mathbf{z}}}{2} \tag{4.33}$$

and

$$\beta^{jk}_{\mathbf{z},2} = \angle \mathbf{b}_k + \frac{\theta^{jk}_{\mathbf{z}}}{2} \tag{4.34}$$

As can easily be seen, the formula for the range limit $[\alpha^{jk}_{\mathbf{z},1}, \alpha^{jk}_{\mathbf{z},2}]$ will be:

$$\alpha^{jk}_{\mathbf{z},1} = \beta^{jk}_{\mathbf{z},1} - \angle \mathbf{m}'_j \tag{4.35}$$

and

$$\alpha^{jk}_{\mathbf{z},2} = \beta^{jk}_{\mathbf{z},2} - \angle \mathbf{m}'_j \tag{4.36}$$

4.6.4.2 Range limit for computing upper bound

This section details the steps to compute the range limit $[\alpha^{jk}_{\mathbb{R},1}, \alpha^{jk}_{\mathbb{R},1}]$ defined in bound calculation part in section 4.2 (cf. the main paper).

Let $\theta^{jk}_{\mathbb{R}}$ be the intersection angle between $\Omega^j_{\mathbb{R}}$ and the disk $\mathbb{O}^k_\epsilon$. The way to compute this angle depends on the relative position between $\mathbf{b}_k$ and the annulus $\Omega^j_{\mathbb{R}}$ plus the value of ϵ. Specifically,

C1 If $r^j_{\mathbb{R},1} \leq \sqrt{|\mathbf{b}_k|^2 - \epsilon^2} \leq r^j_{\mathbb{R},2}$ (Fig. 4.19): $\theta^{jk}_{\mathbb{R}}$ is the angle between two tangent lines starting from the center of the $\mathbb{O}^k_\epsilon$ disk. Mathematically,

$$\theta^{jk}_{\mathbb{R}} = 2 * \arcsin \frac{\epsilon}{|\mathbf{b}_k|} \tag{4.37}$$

C2 If $\sqrt{|\mathbf{b}_k|^2 - \epsilon^2} < r^j_{\mathbb{R},1}$ or $\sqrt{|\mathbf{b}_k|^2 - \epsilon^2} > r^j_{\mathbb{R},2}$. There are two possibilities:

C2.1 The outline of $\mathbb{O}^k_\epsilon$ intersects with either the inner **or** the outer ring of $\Omega^j_{\mathbb{R}}$: $\theta^{jk}_{\mathbb{R}}$ is computed using circle to circle intersection (Fig. 4.20)

C2.2 The outline of $\mathbb{O}^k_\epsilon$ intersects with both the inner **and** the outer ring of $\Omega^j_{\mathbb{R}}$: $\theta^{jk}_{\mathbb{R}}$ is determined by the ring that has larger intersection angle with $\mathbb{O}^k_\epsilon$ using circle to circle intersection (Fig. 4.21)

Similar to 4.6.4.1, define $\beta^{jk}_{\mathbb{R},1}$ and $\beta^{jk}_{\mathbb{R},2}$ to be the limiting angles of the intersection area

$$\beta^{jk}_{\mathbb{R},1} = \angle \mathbf{b}_k - \frac{\theta^{jk}_{\mathbb{R}}}{2} \tag{4.38}$$

and

$$\beta^{jk}_{\mathbb{R},2} = \angle \mathbf{b}_k + \frac{\theta^{jk}_{\mathbb{R}}}{2} \tag{4.39}$$

Finally, the range limit can be computed as:

$$\alpha^{jk}_{\mathbb{R},1} = \beta^{jk}_{\mathbb{R},1} - \theta^{j}_{\mathbb{R},2} \tag{4.40}$$

and

$$\alpha^{jk}_{\mathbb{R},2} = \beta^{jk}_{\mathbb{R},2} - \theta^{j}_{\mathbb{R},1} \tag{4.41}$$

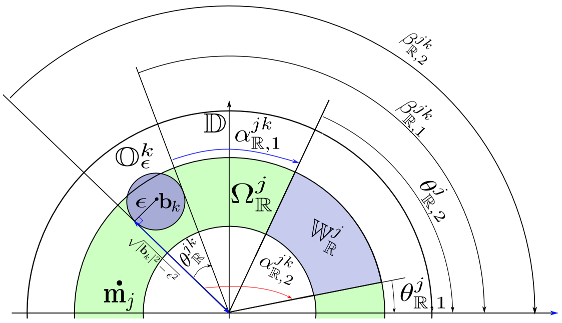
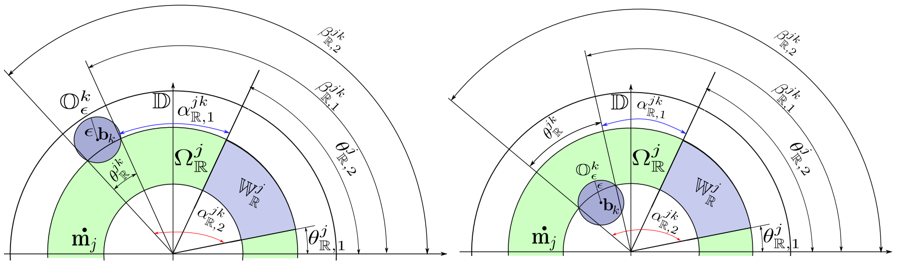

$\beta^{jk}_{\mathbb{R},2}$
$\beta^{jk}_{\mathbb{R},1}$
$\mathbb{D}$
$\alpha^{jk}_{\mathbb{R},1}$
$\mathbb{O}^k_\epsilon$
$\Omega^j_{\mathbb{R}}$
ϵ
$\mathbf{b}_k$
$W^j_{\mathbb{R}}$
$\theta^j_{\mathbb{R},2}$
$\sqrt{|\mathbf{b}_k|^2-\epsilon^2}$
$\theta^{jk}_{\mathbb{R}}$
$\alpha^{jk}_{\mathbb{R},2}$
$\dot{\mathbf{m}}_j$
$\theta^j_{\mathbb{R},1}$

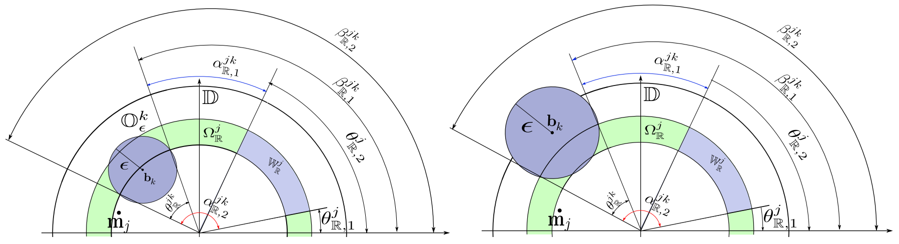
FIGURE 4.19: Illustration of C1: $r^j_{\mathbb{R},1} \leq \sqrt{|\mathbf{b}_k|^2 - \epsilon^2} \leq r^j_{\mathbb{R},2}$


$\beta^{jk}_{\mathbb{R},2}$
$\beta^{jk}_{\mathbb{R},1}$
$\mathbb{O}^k_\epsilon$
$\mathbb{D}$
$\alpha^{jk}_{\mathbb{R},1}$
ϵ
$\mathbf{b}_k$
$\Omega^j_{\mathbb{R}}$
$\theta^j_{\mathbb{R},2}$
$\theta^{jk}_{\mathbb{R}}$
$W^j_{\mathbb{R}}$
$\alpha^{jk}_{\mathbb{R},2}$
$\dot{\mathbf{m}}_j$
$\theta^j_{\mathbb{R},1}$

(a) Outline of $\mathbb{O}^k_\epsilon$ intersects with the outer ring of $\Omega^j_{\mathbb{R}}$

(b) Outline of $\mathbb{O}^k_\epsilon$ intersects with the inner ring of $\Omega^j_{\mathbb{R}}$

FIGURE 4.20: Illustration of C2.1: Outline of $\mathbb{O}^k_\epsilon$ intersects with either the inner or outer ring of the annulus $\Omega^j_{\mathbb{R}}$


$\beta^{jk}_{\mathbb{R},2}$
$\alpha^{jk}_{\mathbb{R},1}$
$\beta^{jk}_{\mathbb{R},1}$
$\mathbb{D}$
$\mathbb{O}^k_\epsilon$
$\Omega^j_{\mathbb{R}}$
$\theta^j_{\mathbb{R},2}$
ϵ
$\mathbf{b}_k$
$W^j_{\mathbb{R}}$
$\theta^{jk}_{\mathbb{R}}$
$\alpha^{jk}_{\mathbb{R},2}$
$\dot{\mathbf{m}}_j$
$\theta^j_{\mathbb{R},1}$

(a) $\theta^{jk}_{\mathbb{R}}$ is determined by intersection between the outline of $\mathbb{O}^k_\epsilon$ with the inner ring

(b) $\theta^{jk}_{\mathbb{R}}$ is determined by intersection between the outline of $\mathbb{O}^k_\epsilon$ with the outer ring

FIGURE 4.21: Illustration of C2.2: Outline of $\mathbb{O}^k_\epsilon$ intersects with both the inner or outer ring of the annulus $\Omega^j_{\mathbb{R}}$

4.6.5 Solving for rotation angle

Algorithm 1 in this document gives the method for solving problem (12) in Sec. 3 of the main paper, i.e., finding the rotation angle that intersects the highest number of angular ranges $\mathcal{S}^j_{\mathbf{z}} =$

$\left\{[\alpha^{jk}_{\mathbf{z},1}, \alpha^{jk}_{\mathbf{z},2}]\right\}_{k=1}^{N_2}$.

This algorithm runs very efficiently in $\mathcal{O}(N \log N)$ time. See Chapter 10 of [M. De Berg, M. Van Kreveld, M. Overmars, and O. C. Schwarzkopf. Computational geometry. Springer, 2000] if more details are required.

Algorithm 4.2 Interval Stabbing

```
Require: Set of angular ranges $\left\{\mathcal{S}^j_{\mathbf{z}} = \left\{[\alpha^{jk}_{\mathbf{z},1}, \alpha^{jk}_{\mathbf{z},2}]\right\}_{k=1}^{N_2}\right\}_{j=1}^{N_1}$
    Set $\mathcal{S} \leftarrow$ empty set; $\theta^* \leftarrow$ null; $U(\mathbf{z}) \leftarrow 0$; $l \leftarrow 0$
    for all $j = 1 \dots N_1$ do
        for all angular intervals $[\alpha^{jk}_{\mathbf{z},1}, \alpha^{jk}_{\mathbf{z},2}]$ in $\mathcal{S}^j_{\mathbf{z}}$ do
            $l \leftarrow l + 1$
            $s_l.a \leftarrow \alpha^{jk}_{\mathbf{z},1}$
            $s_l.f \leftarrow 1$
            Insert $s_l$ into $\mathcal{S}$
            $l \leftarrow l + 1$
            $s_l.a \leftarrow \alpha^{jk}_{\mathbf{z},2}$
            $s_l.f \leftarrow -1$
            Insert $s_l$ into $\mathcal{S}$
        end for
    end for
    Sort all elements $s_l \in \mathcal{S}$ by $s_l.a$ in ascending order $\rightarrow S'$
    $c \leftarrow 0$
    for all $s_l \in S'$ do
        $c \leftarrow c + s_l.f$
        if $c > U(\mathbf{z})$ then
            $\theta^* \leftarrow s_l.a$
            $U(\mathbf{z}) \leftarrow c$
        end if
    end for
    return $\theta^*$ and $U(\mathbf{z})$.
```

Chapter 5

An Exact Penalty Method for Locally Convergent Maximum Consensus

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

Huu Le, Tat-Jun Chin and David Suter: An Exact Penalty Method for Locally Convergent Maximum Consensus. In Computer Vision and Pattern Recognition (CVPR) 2017.

The published paper is available at: http://ieeexplore.ieee.org/document/8099531/

Statement of Authorship

Title of Paper	An Exact Penalty Method for Locally Convergent Maximum Consensus		
Publication Status	☑ Published	□ Accepted for Publication	
	□ Submitted for Publication	□ Unpublished and Unsubmitted w ork w ritten in manuscript style	
Publication Details	**Huu Le**, Tat-Jun Chin and David Suter: An Exact Penalty Method for Locally Convergent Maximum Consensus. In Computer Vision and Pattern Recognition (CVPR) 2017.		

Principal Author

Name of Principal Author (Candidate)	Huu Le		
Contribution to the Paper	- Proposed the idea and developed the algorithm. - Implemented and tested the algorithm. - Conducted all the experiments to validate the performance of the proposed method. - Wrote the manuscript.		
Overall percentage (%)	80%		
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.		
Signature		Date	20th Dec 2017

Co-Author Contributions

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

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Tat-Jun Chin		
Contribution to the Paper	- Proposed the general research direction. - Supervised the development of this work. - Refined the manuscript.		
Signature		Date	20th Dec 2017

Name of Co-Author	David Suter		
Contribution to the Paper	- Supervised the development of this work - Refined the manuscript.		
Signature		Date	20th Dec 2017

5.1 Introduction

Robust model fitting lies at the core of computer vision, due to the need of many fundamental tasks to deal with real-life data that are noisy and contaminated with outliers. To conduct robust model fitting, a robust fitting criterion is optimized w.r.t. a set of input measurements. Arguably the most popular robust criterion is *maximum consensus*, which aims to find the model that is consistent with the largest number of inliers, i.e., has the highest consensus.

Due to the critical importance of maximum consensus estimation, considerable effort has been put into devising algorithms for optimizing the criterion. A large amount of work occurred within the framework of hypothesize-and-verify methods, i.e., RANSAC [31] and its variants. Broadly speaking, these methods operate by fitting the model onto randomly sampled minimal subsets of the data, and returning the candidate with the largest inlier set. Improvements to the basic algorithm include guided sampling and speeding up the model verification step [21].

An important innovation is locally optimized RANSAC (LO-RANSAC) [23, 56]. As the name suggests, the objective of the method is to locally optimize RANSAC estimates. This is achieved by embedding in RANSAC an inner hypothesize-and-verify routine, which is triggered whenever the solution is updated in the outer loop. Different from the main RANSAC algorithm, the inner subroutine generates hypotheses from larger-than-minimal subsets sampled from the inlier set of the incumbent solution, in the hope of driving it towards an improved result.

Though efficient, there are fundamental shortcomings in the hypothesize-and-verify heuristic. Primarily, it does not give analytical assurances of the quality of its solutions. This weakness manifests in LO-RANSAC in that the algorithm does not strictly guarantee local convergence. The randomized nature of the heuristic also means that different runs may give unpredictably different results, which makes it non-ideal for tasks that require high repeatability.

More recently, there is a growing number of globally optimal algorithms for consensus maximization [73, 96, 26, 59, 18]. The fundamental intractability of maximum consensus estimation, however, means that the global optimum can only be found by searching. Indeed, the previous techniques respectively conduct branch-and-bound search [96, 59], tree search [18], or exhaustive search [73, 26]. Thus, global algorithms are practical only on problems with a small number of measurements and/or models of low dimensionality.

So far, what is sorely missing in the literature is an algorithm that lies in the middle ground between the above two extremes. Specifically, a maximum consensus algorithm that is *deterministic and locally convergent* would add significantly to the robust fitting toolbox of computer vision.

In this paper, we contribute such an algorithm. We reformulate consensus maximization with linear complementary constraints, then define a penalized version of the problem. We prove that, under easily achievable conditions, the penalty problem is equivalent to the original problem, in the sense that they have the same optima (local and global). We then develop a Frank-Wolfe algorithm to deterministically solve maximum consensus up to local optimality. Overall, our method executes a sequence of linear programs (LP), which enables it to be efficient on realistic inputs (hundreds to thousands of measurements). Further, our algorithm is naturally capable of handling the non-linear geometric residuals commonly used in computer vision [49, 51]. As will be demonstrated, our method typically achieves better results than RANSAC and LO-RANSAC, while incurring only marginally higher runtimes.

The above properties make our algorithm an excellent alternative to RANSAC and its variants, which are currently dominant in the field. Matlab code and demo program for our method are provided in the supplementary material.

5.1.1 M-estimators and IRLS

More broadly in statistics, M-estimators [46] is an established robust statistical method. The M-estimate is obtained by minimizing the sum of a set of ρ functions defined over the residuals, where ρ (e.g., the Huber norm) is responsible for discounting the effects of outliers. The primary technique for the minimization is iteratively reweighted least squares (IRLS). At each iteration of IRLS, a weighted least squares problem is solved, where the weights are computed based on the previous estimate. Provided that ρ satisfies certain properties [95, 5], IRLS will converge to a minimum. This unfortunately precludes consensus maximization, since the corresponding ρ (a symmetric step function) is not positive definite and differentiable everywhere.

Arguably, one can simply choose a robust ρ that works with IRLS and dispense with maximum consensus. However, another vital requirement for IRLS to be feasible is that the weighted least squares problem is efficiently solvable. This unfortunately is not the case for many of the geometric distances used in computer vision [49, 51, 40].

The above limitations with IRLS suggest that locally convergent algorithms for robust fitting remains an open problem, and that our proposed algorithm should represent a significant contribution towards this direction.

5.2 Problem definition

We develop our algorithm in the context of fitting linear models, before extending to models with geometric residuals in Sec. 5.4.2. The goal of maximum consensus is to find the model, parametrized by vector $\boldsymbol{\theta} \in \mathbb{R}^d$, that is consistent with as many of the input data as possible, i.e.,

$$\begin{aligned} \max_{\boldsymbol{\theta} \in \mathbb{R}^d,\, \mathcal{I} \in \mathcal{P}(N)} \quad & |\mathcal{I}| \\ \text{subject to} \quad & |\mathbf{x}_j^T \boldsymbol{\theta} - y_j| \leq \epsilon \qquad \forall j \in \mathcal{I}, \end{aligned} \tag{5.1}$$

where $\{\mathbf{x}_j, y_j\}_{j=1}^N$ is a set of N measurements for the linear model, ϵ is the inlier threshold, and $\mathcal{P}(N)$ is the power set (the set of all subsets) of the index set $\{1, 2, \ldots, N\}$.

Expressing each constraint of the form $|\mathbf{x}_j^T \boldsymbol{\theta} - y_j| \leq \epsilon$ equivalently using the two linear constraints

$$\mathbf{x}_j^T \boldsymbol{\theta} - y_j \leq \epsilon, \quad -\mathbf{x}_j^T \boldsymbol{\theta} + y_j \leq \epsilon, \tag{5.2}$$

and collecting the data into the matrices

$$\mathbf{A} = \Big[\mathbf{x}_1, -\mathbf{x}_1, \ldots, \mathbf{x}_N, -\mathbf{x}_N\Big], \tag{5.3}$$

$$\mathbf{b} = \Big[\epsilon + y_1, \epsilon - y_1, \ldots, \epsilon + y_N, \epsilon - y_N\Big]^T, \tag{5.4}$$

where $\mathbf{A} \in \mathbb{R}^{d \times M}$, $\mathbf{b} \in \mathbb{R}^M$ and $M = 2N$, we can rewrite (5.1) equivalently as

$$\begin{aligned} \max_{\boldsymbol{\theta} \in \mathbb{R}^d,\, \mathcal{I} \in \mathcal{P}(M)} \quad & |\mathcal{I}| \\ \text{subject to} \quad & \mathbf{a}_i^T \boldsymbol{\theta} - b_i \leq 0 \qquad \forall i \in \mathcal{I}, \end{aligned} \tag{5.5}$$

where $\mathbf{a}_i$ is the i-th column of $\mathbf{A}$ and b_i is the i-th element of $\mathbf{b}$. Problems (5.1) and (5.5) are equivalent in the sense that they have the same maximizer, though the maximum objective value of (5.5) is twice that of (5.1) since for any $\boldsymbol{\theta}$, at least one of the constraints in (5.2) are satisfied.

Henceforth, we will be developing our maximum consensus algorithm based on (5.5) as our target problem.

5.2.1 Complementarity constraints

Introducing indicator variables $\mathbf{u} \in \{0,1\}^M$ and slack variables $\mathbf{s} \in \mathbb{R}^M$, we reformulate (5.5) equivalently as an outlier count minimization problem

$$\begin{aligned}
\min_{\mathbf{u}\in\{0,1\}^M,\, \mathbf{s}\in\mathbb{R}^M,\, \boldsymbol{\theta}\in\mathbb{R}^d} \quad & \sum_i u_i && \text{(5.6a)}\\
\text{subject to} \quad & s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i \geq 0, && \text{(5.6b)}\\
& u_i(s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i) = 0, && \text{(5.6c)}\\
& s_i(1-u_i) = 0, && \text{(5.6d)}\\
& s_i \geq 0. && \text{(5.6e)}
\end{aligned}$$

Intuitively, s_i must be non-zero if the i-th datum is an outlier w.r.t. $\boldsymbol{\theta}$; in this case, u_i must be set to 1 to satisfy (5.6d). In turn, (5.6c) forces the quantity $(s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i)$ to be zero. Conversely, if the i-th datum is an inlier w.r.t. $\boldsymbol{\theta}$, then s_i is zero, u_i is zero and $(s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i)$ is non-zero. Observe, therefore, that (5.6c) and (5.6d) implement *complementarity* between u_i and $(s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i)$.

Note also that, due to the objective function and the condition (5.6d), the indicator variables can be relaxed without impacting the optimum, leading to the equivalent problem

$$\begin{aligned}
\min_{\mathbf{u},\mathbf{s}\in\mathbb{R}^M,\, \boldsymbol{\theta}\in\mathbb{R}^d} \quad & \sum_i u_i && \text{(5.7a)}\\
\text{subject to} \quad & s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i \geq 0, && \text{(5.7b)}\\
& u_i(s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i) = 0, && \text{(5.7c)}\\
& s_i(1-u_i) = 0, && \text{(5.7d)}\\
& 1 - u_i \geq 0, && \text{(5.7e)}\\
& s_i, u_i \geq 0. && \text{(5.7f)}
\end{aligned}$$

This, however, does not make (5.7) tractable to solve exactly, since (5.7c) and (5.7d) are bilinear in the unknowns.

To re-express (5.7) using only positive variables, define

$$\mathbf{v} = \begin{bmatrix} \boldsymbol{\theta} + \gamma\mathbf{1} \\ \gamma \end{bmatrix}, \quad \mathbf{c}_i = \begin{bmatrix} \mathbf{a}_i^T & -\mathbf{a}_i^T\mathbf{1} \end{bmatrix}^T, \tag{5.8}$$

where both are real vectors of length $(d+1)$. Problem (5.7) can then be reformulated equivalently as

$$\begin{aligned} \min_{\mathbf{u},\mathbf{s}\in\mathbb{R}^M,\, \mathbf{v}\in\mathbb{R}^{d+1}} \quad & \sum_i u_i \\ \text{subject to} \quad & s_i - \mathbf{c}_i^T\mathbf{v} + b_i \geq 0, \\ & u_i(s_i - \mathbf{c}_i^T\mathbf{v} + b_i) = 0, \\ & s_i(1-u_i) = 0, \\ & 1 - u_i \geq 0, \\ & s_i, u_i, v_i \geq 0. \end{aligned} \tag{5.9}$$

Given a solution $\hat{\mathbf{u}}$, $\hat{\mathbf{s}}$ and $\hat{\mathbf{v}}$ to (5.9), the corresponding solution $\hat{\boldsymbol{\theta}}$ to (5.7) can be obtained by simply subtracting the last element of $\hat{\mathbf{v}}$ from its first-d elements.

5.3 Penalty method

Incorporating the equality constraints in (5.9) into the cost function as a penalty term, we obtain the penalty problem

$$\begin{aligned} \min_{\mathbf{u},\mathbf{s},\mathbf{v}} \quad & \sum_i u_i + \alpha\left[u_i(s_i - \mathbf{c}_i^T\mathbf{v} + b_i) + s_i(1-u_i)\right] \\ \text{s.t.} \quad & s_i - \mathbf{c}_i^T\mathbf{v} + b_i \geq 0, \\ & 1 - u_i \geq 0, \\ & s_i, u_i, v_i \geq 0. \end{aligned} \tag{5.10}$$

The constant $\alpha \geq 0$ is called the penalty parameter. Intuitively, the penalty term discourages solutions that violate the complementarity constraints, and the strength of the penalization is controlled by α.

Henceforth, to reduce clutter, we sometimes use

$$\mathbf{z} = \left[\mathbf{u}^T\ \mathbf{s}^T\ \mathbf{v}^T\right]^T. \tag{5.11}$$

In addition, the cost function in (5.10) is rewritten as

$$P(\mathbf{z} \mid \alpha) = F(\mathbf{z}) + \alpha Q(\mathbf{z}), \tag{5.12}$$

where $F(\mathbf{z}) = \|\mathbf{u}\|_1$ and

$$Q(\mathbf{z}) = \sum_i u_i(s_i - \mathbf{c}_i^T\mathbf{v} + b_i) + s_i(1 - u_i) \tag{5.13}$$

$$= \sum_i s_i - u_i(\mathbf{c}_i^T\mathbf{v} - b_i). \tag{5.14}$$

In particular, $Q(\mathbf{z})$ is called the *complementarity residual*.

In Sec. 5.3.3, we will investigate the conditions under which solving (5.10) is equivalent to solving (5.9), and devise an algorithm in Sec. 5.4 to exploit the equivalence for consensus maximization. First, in the next two subsections, we discuss solving the penalty problem (5.10) for a given α.

5.3.1 Necessary optimality conditions

Although $P(\mathbf{z} \mid \alpha)$ is quadratic, problem (5.10) is non-convex. However, it can be shown that (5.10) has a vertex solution, i.e., is an extreme point of the convex set

$$\mathcal{P} = \{\mathbf{z} \in \mathbb{R}^{2M+d+1} \mid \mathbf{M}\mathbf{z} + \mathbf{q} \geq \mathbf{0}, \;\; \mathbf{z} \geq \mathbf{0}\}, \tag{5.15}$$

where

$$\begin{aligned} \mathbf{M} &= \begin{bmatrix} \mathbf{0} & \mathbf{I} & -\mathbf{C} \\ -\mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \\ \mathbf{C} &= \begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \dots & \mathbf{c}_M \end{bmatrix}^T, \\ \mathbf{q} &= \begin{bmatrix} \mathbf{b}^T & \mathbf{1}^T \end{bmatrix}^T; \end{aligned} \tag{5.16}$$

(here and henceforth, to minimize clutter we do not define the sizes of $\mathbf{I}$, $\mathbf{0}$ and $\mathbf{1}$, but the sizes can be worked out based on the context). To begin, observe that the minima of (5.10) obey the

KKT conditions [69, Chap. 12]

$$\begin{aligned}
&\mathbf{u}^T(-\alpha\mathbf{C}\mathbf{v} + \alpha\mathbf{b} + \mathbf{1} + \boldsymbol{\lambda}^{\mathcal{G}}) = 0,\\
&\mathbf{s}^T(\alpha\mathbf{1} - \boldsymbol{\lambda}^{\mathcal{H}}) = 0,\\
&\mathbf{v}^T(-\alpha\mathbf{C}^T\mathbf{u} + \mathbf{C}^T\boldsymbol{\lambda}^{\mathcal{H}}) = 0,\\
&(\boldsymbol{\lambda}^{\mathcal{H}})^T(\mathbf{s} - \mathbf{C}\mathbf{v} + \mathbf{b}) = 0,\\
&(\boldsymbol{\lambda}^{\mathcal{G}})^T(\mathbf{1} - \mathbf{u}) = 0,\\
&\mathbf{s} - \mathbf{C}\mathbf{v} + \mathbf{b} \geq \mathbf{0},\\
&\mathbf{1} - \mathbf{u} \geq \mathbf{0},\\
&\boldsymbol{\lambda}^{\mathcal{H}}, \boldsymbol{\lambda}^{\mathcal{G}}, \mathbf{u}, \mathbf{v}, \mathbf{s} \geq \mathbf{0},
\end{aligned} \tag{5.17}$$

where $\boldsymbol{\lambda}^{\mathcal{H}} = [\lambda_1^{\mathcal{H}} \ \ldots \ \lambda_M^{\mathcal{H}}]^T$ and $\boldsymbol{\lambda}^{\mathcal{G}} = [\lambda_1^{\mathcal{G}} \ \ldots \ \lambda_M^{\mathcal{G}}]^T$ are the Lagrange multipliers for the first two types of constraints in (5.10); see supplementary material for details.

By rearranging, the KKT conditions (5.17) can be summarized by the following relations

$$\mathbf{M}'\mathbf{z}' + \mathbf{q}' \geq \mathbf{0}, \ \ \mathbf{z}' \geq \mathbf{0}, \ \ (\mathbf{z}')^T(\mathbf{M}'\mathbf{z}' + \mathbf{q}') = 0, \tag{5.18}$$

where

$$\begin{aligned}
\mathbf{z}' &= \begin{bmatrix}\mathbf{z}^T & (\boldsymbol{\lambda}^{\mathcal{H}})^T & (\boldsymbol{\lambda}^{\mathcal{G}})^T\end{bmatrix}^T,\\
\mathbf{M}' &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & -\alpha\mathbf{C} & \mathbf{0} & \mathbf{I}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0}\\ -\alpha\mathbf{C}^T & \mathbf{0} & \mathbf{0} & \mathbf{C}^T & \mathbf{0}\\ \mathbf{0} & \mathbf{I} & -\mathbf{C} & \mathbf{0} & \mathbf{0}\\ -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\end{bmatrix},\\
\mathbf{q}' &= \begin{bmatrix}(\alpha\mathbf{b}+\mathbf{1})^T & \alpha\mathbf{1}^T & \mathbf{0}^T & \mathbf{b}^T & \mathbf{1}^T\end{bmatrix}^T.
\end{aligned} \tag{5.19}$$

Finding a feasible $\mathbf{z}'$ for (5.18) is an instance of a *linear complementarity problem (LCP)* [66]. Define the convex set

$$\mathcal{P}' = \{\mathbf{z}' \in \mathbb{R}^{4M+d+1} \mid \mathbf{M}'\mathbf{z}' + \mathbf{q}' \geq \mathbf{0}, \ \ \mathbf{z}' \geq \mathbf{0}\}. \tag{5.20}$$

We invoke the following result from [66, Lemma 2].

Theorem 5.1. *If the LCP defined by the constraints* (5.18) *has a solution, then it has a solution at a vertex of* $\mathcal{P}'$.

Theorem 5.1 implies that the KKT points of (5.10) (including the solutions of the problem) occur at the vertices of $\mathcal{P}'$. This also implies that (5.10) has a vertex solution, viz.:

Theorem 5.2. *For any vertex*

$$\mathbf{z}'_v = [\mathbf{z}_v^T \ \ (\boldsymbol{\lambda}_v^{\mathcal{H}})^T \ \ (\boldsymbol{\lambda}_v^{\mathcal{G}})^T)]^T \tag{5.21}$$

of $\mathcal{P}'$*,* $\mathbf{z}_v$ *is a vertex of* $\mathcal{P}$*.*

Proof. If $\mathbf{z}'_v$ is a vertex of $\mathcal{P}'$, then, there is a diagonal matrix $\mathbf{E}$ such that

$$\mathbf{M}'\mathbf{E}\mathbf{z}'_v + \mathbf{q}' - \boldsymbol{\gamma}' = \mathbf{0}, \tag{5.22}$$

where $\mathbf{E}_{i,i} = 1$ if the i-th column of $\mathbf{M}'$ appears in the basic solution corresponding to vertex $\mathbf{z}'_v$, and $\mathbf{E}_{i,i} = 0$ otherwise (the non-negative vector $\boldsymbol{\gamma}'$ contains the values of additional slack variables to convert the constraints in $\mathcal{P}'$ into standard form). Let $\mathbf{M}'_J$ be the last-$2M$ rows of $\mathbf{M}'$. Then,

$$\mathbf{M}'_J\mathbf{E}\mathbf{z}'_v + \begin{bmatrix}\mathbf{b}^T & \mathbf{1}^T\end{bmatrix}^T - \boldsymbol{\gamma}'_J = \mathbf{0}, \tag{5.23}$$

where $\boldsymbol{\gamma}'_J$ is the last-$2M$ elements of $\boldsymbol{\gamma}'$. Note that, since the right-most $2M \times 2M$ submatrix of $\mathbf{M}'_J$ is a zero matrix (see (5.19)), then

$$\mathbf{M}'_J\mathbf{E}_K\mathbf{z}_v + \begin{bmatrix}\mathbf{b}^T & \mathbf{1}^T\end{bmatrix}^T - \boldsymbol{\gamma}'_J = \mathbf{0}, \tag{5.24}$$

where $\mathbf{E}_K$ is the first-$(2M+d+1)$ columns of $\mathbf{E}$. Since $\mathbf{M}'_J\mathbf{E}_K = \mathbf{M}$, then (5.24) implies that $\mathbf{z}_v$ is a vertex of $\mathcal{P}$. □

5.3.2 Frank-Wolfe algorithm

Theorem 5.2 suggests an approach to solve (5.10) by searching for a vertex solution. Further, note that for a fixed $\mathbf{u}$, (5.10) reduces to an LP. Conversely, for fixed $\mathbf{s}$ and $\mathbf{v}$, (5.10) is also an LP. This advocates alternating between optimizing subsets of the variables using LPs. Algorithm 5.1 summarizes the method, which is in fact a special case of the Frank-Wolfe method [32] for non-convex quadratic minimization.

Theorem 5.3. *In a finite number of steps, Algorithm 5.1 converges to a KKT point of* (5.10).

Proof. The set of constraints $\mathcal{P}$ can be decoupled into the two disjoint subsets

$$\mathcal{P} = \mathcal{P}_1 \times \mathcal{P}_2, \tag{5.25}$$

where $\mathcal{P}_1$ involves only $\mathbf{s}$ and $\mathbf{v}$, and $\mathcal{P}_2$ is the complement of $\mathcal{P}_1$. With $\mathbf{u}$ fixed in Line 5, the LP converges to a vertex of $\mathcal{P}_1$. Similarly, with $\mathbf{s}$ and $\mathbf{v}$ fixed in Line 6, the LP converges to

Algorithm 5.1 Frank-Wolfe method for (5.10).

```
Require: Data $\{\mathbf{c}_i, b_i\}_{i=1}^M$, penalty value $\alpha$, initial solution $\mathbf{u}^{(0)}$, $\mathbf{v}^{(0)}$, $\mathbf{s}^{(0)}$, threshold $\delta$.
 1: $P^{(0)} \leftarrow P(\mathbf{u}^{(0)}, \mathbf{s}^{(0)}, \mathbf{v}^{(0)} \mid \alpha)$.
 2: $t \leftarrow 0$.
 3: while true do
 4:     $t \leftarrow t + 1$.
 5:     $\mathbf{s}^{(t)}, \mathbf{v}^{(t)} \leftarrow \arg\min_{\mathbf{s},\mathbf{v}} P(\mathbf{u}^{(t-1)}, \mathbf{s}, \mathbf{v} \mid \alpha)$ s.t. $\mathcal{P}$.
 6:     $\mathbf{u}^{(t)} \leftarrow \arg\min_{\mathbf{u}} P(\mathbf{u}, \mathbf{s}^{(t)}, \mathbf{v}^{(t)} \mid \alpha)$ s.t. $\mathcal{P}$.
 7:     $P^{(t)} \leftarrow P(\mathbf{u}^{(t)}, \mathbf{s}^{(t)}, \mathbf{v}^{(t)} \mid \alpha)$.
 8:     if $|P^{(t-1)} - P^{(t)}| \leq \delta$ then
 9:         Break.
10:     end if
11: end while
12: return $\mathbf{u}^{(t)}, \mathbf{v}^{(t)}, \mathbf{s}^{(t)}$.
```

a vertex in $\mathcal{P}_2$. Each intermediate solution $\mathbf{u}^{(t)}, \mathbf{v}^{(t)}, \mathbf{s}^{(t)}$ is thus a vertex of $\mathcal{P}$ or a KKT point of (5.10). Since each LP must reduce or maintain $P(\mathbf{z} \mid \alpha)$ which is bounded below, the process terminates in finite steps. □ □

Analysis of update steps A closer look reveals the LP in Line 5 (Algorithm 5.1) to be

$$\begin{aligned} \min_{\mathbf{s},\mathbf{v}} \quad & \sum_i s_i - u_i(\mathbf{c}_i^T \mathbf{v} - b_i) \\ \text{s.t.} \quad & s_i - \mathbf{c}_i^T \mathbf{v} + b_i \geq 0, \\ & s_i, v_i \geq 0, \end{aligned} \tag{LP1}$$

and the LP in Line 6 (Algorithm 5.1) to be

$$\begin{aligned} \min_{\mathbf{u}} \quad & \sum_i u_i \left[1 - \alpha(\mathbf{c}_i^T \mathbf{v} - b_i)\right] \\ \text{s.t.} \quad & 0 \leq u_i \leq 1. \end{aligned} \tag{LP2}$$

Observe that LP2 can be solved in closed form and it also drives $\mathbf{u}$ to integrality: if $[1-\alpha(\mathbf{c}_i^T\mathbf{v} - b_i)] \leq 0$, set $u_i = 1$, else, set $u_i = 0$. Further, LP1 can be seen as "weighted" ℓ_1-norm minimization, with $\mathbf{u}$ being the weights. Intuitively, therefore, Algorithm 5.1 alternates between residual minimization (LP1) and inlier-outlier dichotomization (LP2).

5.3.3 Exactness of penalization

The penalty problem (5.10) is an instance of a non-smooth exact penalty method [69, Sec. 17.2]. Observe that $Q(\mathbf{z})$ is the ℓ_1-norm of the LHS of the equality constraints in (5.9). The exactness of the penalization is exhibited in the following theorems (rephrased in the context of our problem).

Theorem 5.4 (based on Theorem 17.3 in [69]). *If $\mathbf{z}^*$ is a local solution of the original problem* (5.9)*, then, there exists $\alpha^* > 0$ such that for all $\alpha \geq \alpha^*$, $\mathbf{z}^*$ is also a local minimizer of $P(\mathbf{z} \mid \alpha)$ subject to constraints $\mathcal{P}$.*

Intuitively, the theorem states that there is a sufficiently large α for problem (5.10), such that any small movement away from $\mathbf{z}^*$ will be penalized strongly enough by $\alpha Q(\mathbf{z})$ to immediately negate any potential reduction to $F(\mathbf{z})$ enabled by violating the complementarity constraints. A follow-up theorem will prove more useful for our aims.

Theorem 5.5 (based on Theorem 17.4 in [69]). *Let $\hat{\mathbf{z}}$ be a KKT point of the penalized problem* (5.10) *for α greater than α^*. Then, $Q(\hat{\mathbf{z}}) = 0$, and $\hat{\mathbf{z}}$ is also a KKT point of* (5.9).

A "one shot" approach that sets α to a very large value and solves a single instance of (5.10) is unlikely to be successful, however, since we cannot globally solve the penalty problem. In the next section, we describe a more practical approach that uses an increasing sequence of α.

5.4 Main algorithm

Based on the above results, we propose our main algorithm for solving the maximum consensus problem (5.9); see Algorithm 5.2. Our method solves (5.10) using Algorithm 5.1 for successively larger α, where the solution $\hat{\mathbf{z}}$ for a particular α is used to initialize Algorithm 5.1 for the next larger α. The sequence terminates when the complementarity residual $Q(\mathbf{z})$ vanishes. As long as each $\hat{\mathbf{z}}$ is a KKT point of the associated penalty problem (5.10), which we can provably achieve thanks to Theorem 6.3, Theorem 5.5 guarantees that Algorithm 5.2 will converge to a solution for (5.9) that satisfies the first-order necessary conditions for optimality.

Algorithm 5.2 Main algorithm for solving (5.9).

```
Require: Data $\{\mathbf{c}_i, b_i\}_{i=1}^M$, initial model parameter $\boldsymbol{\theta}$, initial penalty value $\alpha$, increment rate $\kappa$,
    threshold $\delta$.
$\mathbf{v} \leftarrow \left[(\boldsymbol{\theta} + |\min_j(\theta_j)|\mathbf{1})^T \;\; |\min_j(\theta_j)|\right]^T$.
$\mathbf{u} \leftarrow \mathbb{I}(\mathbf{C}\mathbf{v} - \mathbf{b} > 0)$.
$\mathbf{s} \leftarrow \mathbf{u} \odot (\mathbf{C}\mathbf{v} - \mathbf{b})$.
while true do
    $\mathbf{u}, \mathbf{s}, \mathbf{v} \leftarrow FW(\{\mathbf{c}_i, b_i\}_{i=1}^M, \alpha, \mathbf{u}, \mathbf{s}, \mathbf{v})$. /*Algo. 5.1.*/
    if $Q(\mathbf{z}) \leq \delta$ then
        Break.
    end if
    $\alpha \leftarrow \kappa \cdot \alpha$.
end while
return $\mathbf{u}, \mathbf{s}, \mathbf{v}$.
```

It is worthwhile to note that typical non-smooth penalty functions cannot be easily minimized (e.g., no gradient information). In our case, however, we exploited the special property of (5.10) (Sec. 5.3.1) to enable efficient minimization.

5.4.1 Initialization

Algorithm 5.2 requires the initialization of $\mathbf{u}$, $\mathbf{s}$ and $\mathbf{v}$. For consensus maximization, it is more natural to initialize the model parameters $\boldsymbol{\theta}$, which in turn gives values to $\mathbf{v}$, $\mathbf{s}$ and $\mathbf{u}$. In our work, we initialize $\boldsymbol{\theta}$ as the least squares solution, or by executing RANSAC (Sec. 5.5 will compare the results of these two different initialization methods).

Other required inputs are the initial penalty parameter α and the increment rate κ. These values affect the convergence speed of Algorithm 5.2. To avoid bad minima, we set α and κ conservatively, e.g., $\alpha \in [1, 10]$, $\kappa \in [1,\ 5]$. As we will demonstrate in Sec. 5.5, these settings enable Algorithm 5.2 to find very good solutions at competitive runtimes.

5.4.2 Handling geometric distances

For most applications in computer vision, the residual function used for geometric model fitting is non-linear. It has been shown [49, 73, 6], however, that many geometric residuals have the following *generalized fractional* form

$$\frac{\|\mathbf{G}\boldsymbol{\theta} + \mathbf{h}\|_p}{\mathbf{r}^T\boldsymbol{\theta} + q} \text{ with } \mathbf{r}^T\boldsymbol{\theta} + q > 0, \tag{5.26}$$

where $\|\cdot\|_p$ is the p-norm, and $\mathbf{G} \in \mathbb{R}^{2\times d}$, $\mathbf{h} \in \mathbb{R}^2$, $\mathbf{r} \in \mathbb{R}^d$, $q \in \mathbb{R}^1$ are constants derived from the input data. For example, the reprojection error in triangulation and transfer error in homography estimation can be coded in the form (5.26). The associated maximum consensus problem is

$$\begin{aligned} \max_{\boldsymbol{\theta}\in\mathbb{R}^d,\ \mathcal{I}\in\mathcal{P}(N)} \quad & |\mathcal{I}| \\ \text{subject to} \quad & \|\mathbf{G}_j\boldsymbol{\theta} + h_j\|_p \le \epsilon(\mathbf{r}_j^T\boldsymbol{\theta} + q_j) \quad \forall j \in \mathcal{I}, \end{aligned} \tag{5.27}$$

where the denominator of (5.26) can be moved to the RHS since ϵ is non-negative (see [49] for details). We show that for $p = 1$, our method can be easily adapted to solve (5.27) up to local optimality[1]. Define

$$\mathbf{G}_j = \begin{bmatrix} \mathbf{g}_{j,1}^T \\ \mathbf{g}_{j,2}^T \end{bmatrix} \quad \mathbf{h}_j = \begin{bmatrix} h_{j,1} \\ h_{j,2} \end{bmatrix}. \tag{5.28}$$

[1] Note that, in the presence of outliers, the residuals are no longer i.i.d. Normal. Thus, the 1-norm is equally valid as the 2-norm for robust fitting.

Now, for $p = 1$, the constraint in (5.27) becomes

$$\left|\mathbf{g}_{j,1}^T\boldsymbol{\theta} + h_{j,1}\right| + \left|\mathbf{g}_{j,2}^T\boldsymbol{\theta} + h_{j,2}\right| \leq \epsilon(\mathbf{r}_j^T\boldsymbol{\theta} + q_j), \tag{5.29}$$

which in turn can be equivalently implemented using four linear constraints (see [6] for details). We can then manipulate (5.27) into the form (5.5), and the rest of our theory and algorithms will be immediately applicable.

5.5 Results

We tested our method (Algorithm 5.2, henceforth abbreviated as EP) on common parameter estimation problems. We compared EP against the following well-known methods:

- RANSAC (RS) [31]: We used confidence $\rho = 0.99$ for the stopping criterion in all the experiments. On each data instance, RANSAC was executed 10 times and the average consensus size and runtime were reported.
- LO-RANSAC (LORS) [23]: The maximum number of iterations for the inner sampling over the updated consensus set was set to 100.
- Improved LO-RANSAC (LORS1) [56]: As proposed, the local refinement will only be run if the new consensus size is higher than a pre-defined threshold (set to 10% of the data size in our experiments).
- ℓ_1 approximation (ℓ_1) [74]: This method is equivalent to introducing slack variables to problem (5.5) and minimizing the ℓ_1-norm of the slack variables to yield an approximate solution to maximum consensus.
- ℓ_∞ outlier removal (l_∞) [79]: Again, in the context of (5.5), slack variables are introduced and the maximum slack value is minimized. Data with the largest slack value are removed, and the process of repeated until the largest slack value is not greater than zero.
- For the experiments with image data where keypoint matching scores are available as inlier priors, we executed two state-of-the-art RANSAC variants: PROSAC (PS) [22] and Guided MLESAC (GMLE) [84].

All the methods and experiments were implemented in MATLAB and run on a standard desktop machine with 3.5 GHz processor and 8 GB of RAM. For EP, ℓ_1 and ℓ_∞, Gurobi was employed as the LP solver.

5.5.1 Linear models

Linear regression with synthetic data We generated $N = 500$ points $\{\mathbf{x}_j, y_j\}_{j=1}^N$ in $\mathbb{R}^9$ following a linear trend $y = \mathbf{x}^T\boldsymbol{\theta}$, where $\boldsymbol{\theta} \in \mathbb{R}^8$ and $\mathbf{x}_j \in [-1, 1]^8$ were randomly sampled. Each y_j was perturbed by Gaussian noise with standard deviation of $\sigma_{in} = 0.1$. To simulate outliers, $p_{out}\%$ of y_j's were randomly selected and corrupted. To test the ability of EP to deal with bad initializations, two different outlier settings were considered:

- Balanced data: the y_j of outliers were added with Gaussian noise of $\sigma_{out} = 1$. This evenly distributed the outliers on both sides of the hyperplane.
- Unbalanced data: as above, but the sign of the additive noise was forced to be positive. Thus, outliers were distributed only on one side of the hyperplane. On such data, the least squares solution is heavily biased.

See Fig. 6.2 for a *2D analogy* of these outlier settings. We tested with $p_{out} = \{0, 5, 10 \ldots, 60\}$. The inlier threshold for maximum consensus was set to $\epsilon = 0.1$.

EP was initialized respectively with RANSAC (variant EP-RS) and least squares (variant EP-LSQ). The initial α was set to 0.5 and $\kappa = 5$ for all the runs.

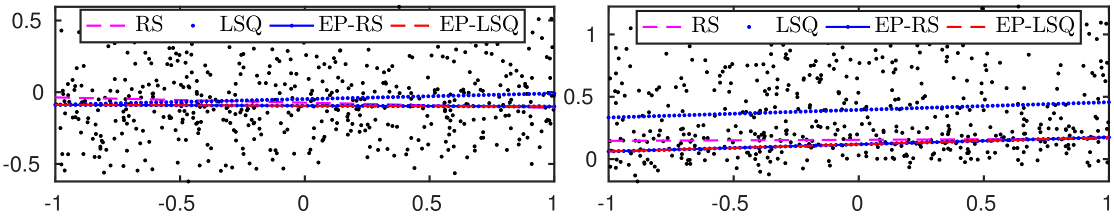
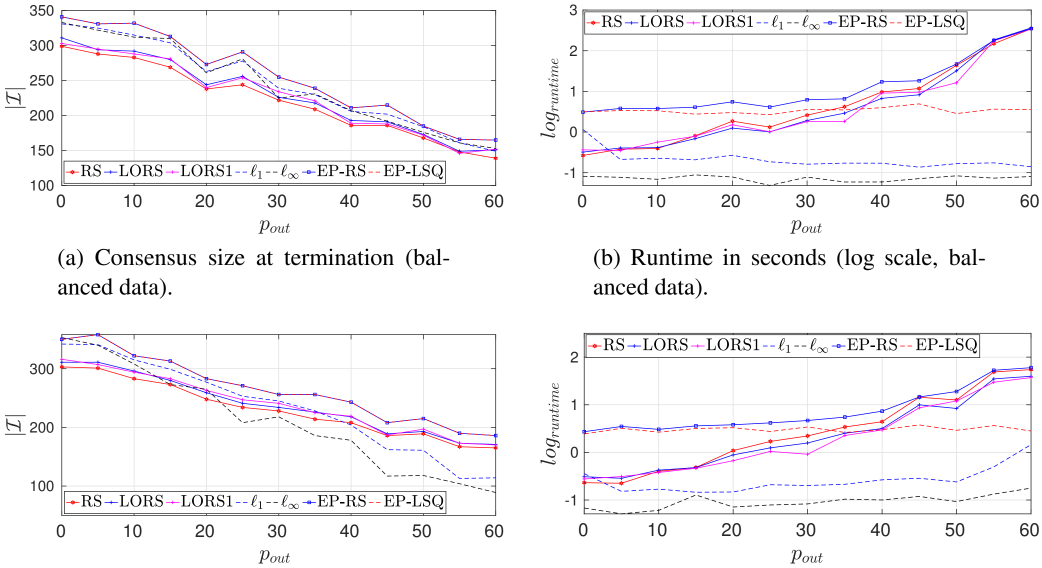


FIGURE 5.1: 2D analogy of balanced (top) and unbalanced (bottom) data generated in our experiments. The results of RANSAC, least squares, and our method initialized with the former two methods are shown. Observe that least squares is heavily biased under unbalanced data, but EP is able to recover from the bad initialization.

Fig. 6.3 shows the average consensus size at termination and runtime (in log scale) of the methods. Note that runtime of RS and LSQ were included in the runtime of EP-RS and EP-LSQ. It is clear that, in terms of solution quality, both variants of EP consistently outperformed the other methods. The fact that EP-LSQ could match the quality of EP-RS on unbalanced data attest to the ability of EP to recover from bad initializations. In terms of runtime, while both EP variants were slightly more expensive than the RANSAC variants, as p_{out} increased over 35%, EP-LSQ began to outperform the RANSAC variants (since EP-RS was initialized using RANSAC, its runtime also increased with p_{out}).

Fundamental matrix estimation Following [42, Chapter 11], the epipolar constraint is linearized to enable the fundamental matrix to be estimated linearly (note that the usual geometric



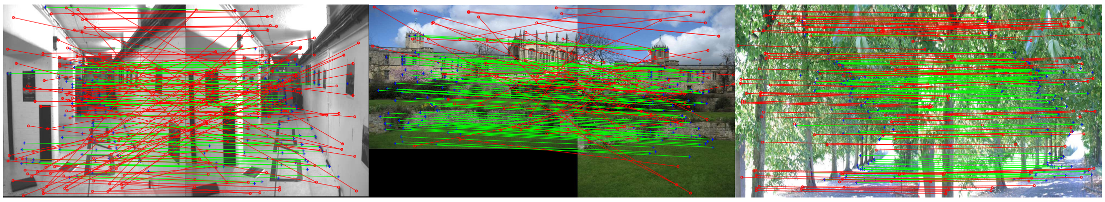
(a) Consensus size at termination (balanced data).

(b) Runtime in seconds (log scale, balanced data).

(c) Consensus size at termination (unbalanced data).

(d) Runtime in seconds (log scale, unbalanced data).

FIGURE 5.2: Results for linear regression ($d = 8$ dimensions). (a)(b) Balanced data; (c)(d) Unbalanced data.



(a) Corridor.

(b) Christ Church.

(c) Trees.

FIGURE 5.3: Qualitative results of EP on (a) fundamental matrix estimation, (b) homography estimation, and (c) affinity estimation. Green and red lines represent detected inliers and outliers. For clarity, only 100 inliers/outliers are plotted. See supp material for more results.

distances for fundamental matrix estimation do not have the generalized fractional form (5.26), thus linearization is essential to enable our method. Sec. 5.5.2 will describe results for model estimation with geometric distances).

Five image pairs from the VGG data set were used: Corridor, House, Merton II, Wadham and Aerial View I. The images were first resized before SIFT (as implemented on VLFeat [87]) was used to extract around 500 correspondences per pair. An inlier threshold of $\epsilon = 1$ was used for all image pairs. For EP, apart from initialization with RANSAC and least squares, we also initialised it with ℓ_∞ outlier removal (variant EP-ℓ_∞). For all EP variants, the initial α was set to 0.5 and $\kappa = 5$ for all the runs.

Table 5.1 summarizes the quantitative results for all methods. Regardless of the initialization method, EP was able to find the largest consensus set. Though the runtimes of EP were higher, they were still in the same order of magnitude as the others. Fig. 5.3a displays a sample qualitative result for EP; for the qualitative results on the other image pairs, please see the supplementary material.

Methods		RS	PS	GMLE	LORS	LORS1	ℓ_1	ℓ_∞	EP-RS	EP-LSQ	EP-ℓ_∞
House	$\lvert\mathcal{I}\rvert$	250	251	254	257	256	175	205	**267**	**267**	**267**
N = 556	time (s)	2.12	1.60	1.09	1.33	3.41	0.2	0.06	7.62	4.79	4.96
Aerial	$\lvert\mathcal{I}\rvert$	267	261	266	283	283	282	277	**297**	**297**	**297**
N = 421	time (s)	0.12	0.16	0.1	0.17	0.27	0.15	0.03	1.91	2.01	1.67
Merton	$\lvert\mathcal{I}\rvert$	367	344	370	377	383	408	404	**451**	**451**	**451**
N = 590	time (s)	0.14	0.27	0.09	0.21	0.32	0.25	0.04	2.84	2.75	3.69
Wadham	$\lvert\mathcal{I}\rvert$	447	426	473	470	476	503	433	**512**	**512**	**512**
N = 587	time (s)	0.05	0.08	0.04	0.12	0.15	0.2	0.04	2.99	3.29	3.06
Corridor	$\lvert\mathcal{I}\rvert$	263	269	263	266	265	246	264	**303**	**303**	**303**
N = 686	time (s)	5.23	4.22	4.64	3.87	9.06	0.72	0.08	15.26	5.57	5.75

TABLE 5.1: Fundamental matrix estimation results. Legend: $|\mathcal{I}|$ = consensus size at termination, RS = RANSAC, PS = PROSAC, GMLE = Guided MLESAC, LORS = LO-RANSAC, LORS1 = Improved LO-RANSAC, EP = proposed method with different initialization techniques.

	Methods		RS	PS	GMLE	LORS	LORS1	ℓ_1	ℓ_∞	EP-RS	EP-ℓ_∞
Homography estimation	University Library	$\lvert\mathcal{I}\rvert$	251	269	251	294	294	120	53	**301**	**301**
	N = 545	time (s)	0.73	0.62	0.69	1.90	1.89	3.10	2.49	12.76	14.49
	Christ Church	$\lvert\mathcal{I}\rvert$	235	236	227	250	246	246	160	**280**	**280**
	N = 445	time (s)	0.47	0.47	0.43	1.33	1.61	1.23	2.44	10.37	12.67
	Valbonne	$\lvert\mathcal{I}\rvert$	131	134	117	156	136	24	22	**158**	**158**
	N = 434	time (s)	3.17	2.39	5.76	3.04	5.80	1.36	1.27	17.20	14.84
	Kapel	$\lvert\mathcal{I}\rvert$	163	167	130	167	168	28	161	**170**	**170**
	N = 449	time (s)	1.19	1.15	9.89	2.18	2.70	1.62	1.16	8.46	8.68
	Invalides	$\lvert\mathcal{I}\rvert$	144	159	140	149	156	84	142	**178**	**178**
	N = 413	time (s)	1.36	0.90	1.60	2.17	2.94	1.04	0.71	10.20	9.15
Affinity estimation	Bikes	$\lvert\mathcal{I}\rvert$	424	427	425	426	424	387	431	**437**	**437**
	N = 557	time (s)	6.09	6.09	5.79	6.28	11.8	1.77	1.77	15.26	9.81
	Graff	$\lvert\mathcal{I}\rvert$	126	129	127	134	126	147	274	**276**	**276**
	N = 327	time (s)	3.51	3.35	3.14	4.07	6.61	0.99	0.23	5.94	2.70
	Bark	$\lvert\mathcal{I}\rvert$	279	288	270	284	279	298	439	**442**	**442**
	N = 458	time (s)	4.89	4.93	4.68	5.11	9.54	1.31	0.19	10.19	5.51
	Tree	$\lvert\mathcal{I}\rvert$	372	367	371	372	372	377	370	**396**	**396**
	N = 568	time (s)	5.70	6.01	5.73	6.93	11.50	4.81	0.81	15.96	11.82
	Boat	$\lvert\mathcal{I}\rvert$	476	477	476	477	476	469	464	**483**	**483**
	N = 574	time (s)	6.32	6.29	6.02	7.18	12.32	4.12	1.02	14.86	9.33

TABLE 5.2: Homography and Affinity estimation results. Legend: $|\mathcal{I}|$ = consensus size at termination, RS = RANSAC, PS = PROSAC, GMLE = Guided MLESAC, LORS = LO-RANSAC, LORS1 = Improved LO-RANSAC, EP = proposed method with different initialization techniques.

5.5.2 Models with geometric distances

Homography estimation We estimated 2D homographies based on the transfer error using all the methods. Five image pairs form the VGG dataset were used: University Library, Christ Church, Valbonne, Kapel and Paris's Invalides. The same preprocessing and correspondence extraction method as in the fundamental matrix estimation experiment was used to produce 500

	Point 719 N = 192		Point 585 N = 153		Point 570 N = 167		Point 24 N = 155		Point 1 N = 167	
	$\lvert\mathcal{I}\rvert$	time	$\lvert\mathcal{I}\rvert$	time	$\lvert\mathcal{I}\rvert$	time	$\lvert\mathcal{I}\rvert$	time	$\lvert\mathcal{I}\rvert$	time
RS	102	0.26	77	0.13	47	0.14	111	0.14	94	0.15
LORS	102	1.16	77	0.60	47	0.65	111	0.71	94	0.78
LORS1	103	0.29	77	0.24	47	0.26	111	0.25	94	0.26
ℓ_1	61	0.27	20	0.17	14	0.23	60	0.13	62	0.33
ℓ_∞	96	1.29	61	0.75	35	0.95	111	0.46	81	1.06
EP-RS	**107**	2.06	**80**	1.02	**54**	1.40	**113**	1.10	**96**	0.96
EP-ℓ_∞	**107**	3.08	**80**	1.70	**54**	2.22	**113**	1.35	**96**	2.16

TABLE 5.3: Triangulation results. Legend: $|\mathcal{I}|$ = consensus size at termination, RS = RANSAC, PS = PROSAC, GMLE = Guided MLESAC, LORS = LO-RANSAC, LORS1 = Improved LO-RANSAC, EP = proposed method with different initialization techniques.

matches per image pair. The inlier threshold of $\epsilon = 4$ pixels was used for all input data. Initial α was set to 10 and $\kappa = 1.5$ for all EP variants.

Quantitative results are shown in Table 5.2, and a sample qualitative result for EP is shown in Fig. 5.3b. Similar to the fundamental matrix case, the EP variants outperformed the other methods in terms of solution quality, but were slower though its runtime was still within the same order of magnitude. Note that EP-LSQ was not invoked here, since finding least squares estimates based on geometric distances is intractable in general [40].

Affinity estimation The previous experiment was repeated for affinity (6 DoF affine transformation) estimation with initial α set to 0.5, $\kappa = 5$ and $\epsilon = 2$ pixels. Five image pairs from VGG's affine image dataset were used: Bikes, Graff, Bark, Tree and Boat. Quantitative results are given in Table 5.2, and sample qualitative result is shown in Fig. 5.3c. Similar conclusions can be drawn.

Triangulation Coordinates of 3D points were estimated using the reprojection error under outlier contamination. We selected five feature tracks from the NotreDame dataset [80] with more than $N = 150$ views each to test our algorithm. The inlier threshold for maximum consensus was set to $\epsilon = 1$ pixel. α was initially set to 0.5 and $\kappa = 1.5$ for all variants of EP.

Table 5.3 shows the quantitative results. Again, the EP variants are better than the other methods in terms of solution quality. The runtime gap was not as significant here due to the low-dimensionality of the model ($d = 3$).

5.6 Conclusions

We introduced a novel locally convergent algorithm for maximum consensus, based on exact penalization of complementary constraints. In terms of solution quality, our algorithm outperforms other heuristic and approximate methods - this was demonstrated particularly by our method being able to improve upon the solution of RANSAC. Even when presented with bad initializations (i.e., when using least squares to initialize on unbalanced data), our method was able to recover and attain good solutions. Though our method can be slower, it is able to guarantee convergence to local optimum, unlike the randomized heuristics. In fact, at high outlier rates, our method is actually faster than the RANSAC variants, while yielding higher-quality results.

Overall, the experiments illustrate that the proposed method can serve well in settings where slight additional runtime is a worthwhile expense for guaranteed convergence to an improved maximum consensus solution.

5.7 Supplementary Material

5.7.1 Derivation of KKT conditions

The penalty problem derived in the main paper:

$$\begin{aligned} \min_{\mathbf{u},\mathbf{s},\mathbf{v}} \quad & \sum_i u_i + \alpha\left[u_i(s_i - \mathbf{c}_i^T\mathbf{v} + b_i) + s_i(1-u_i)\right] \\ \text{s.t.} \quad & s_i - \mathbf{c}_i^T\mathbf{v} + b_i \geq 0, \\ & 1 - u_i \geq 0, \\ & s_i, u_i, v_i \geq 0. \end{aligned} \tag{5.30}$$

Define the functions correspond to the set of constraints of the penalty problem (5.30):

$$\begin{aligned} \mathcal{H}_i &= s_i - \mathbf{c}_i^T\mathbf{v} + b_i \\ \mathcal{G}_i &= 1 - u_i \\ \mathcal{S}_i &= s_i \\ \mathcal{U}_i &= u_i \\ \mathcal{V}_i &= v_i \end{aligned} \tag{5.31}$$

Also, define $\boldsymbol{\lambda}^{\mathcal{H}}$, $\boldsymbol{\lambda}^{\mathcal{G}}$, $\boldsymbol{\lambda}^{\mathcal{S}}$, $\boldsymbol{\lambda}^{\mathcal{U}}$, $\boldsymbol{\lambda}^{\mathcal{V}} \in \mathbb{R}^M$ be the Larange multipliers for the constraints in (5.30).

Derivaties of the cost function in (5.30) with respect to $\mathbf{u}$, $\mathbf{v}$, $\mathbf{s}$ respectively:

$$\begin{aligned} \nabla_{\mathbf{u}} P(\mathbf{u}, \mathbf{s}, \mathbf{v}|\alpha) &= -\alpha\mathbf{C}\mathbf{v} + \alpha\mathbf{b} + \mathbf{1} \\ \nabla_{\mathbf{s}} P(\mathbf{u}, \mathbf{s}, \mathbf{v}|\alpha) &= \alpha\mathbf{1} \\ \nabla_{\mathbf{v}} P(\mathbf{u}, \mathbf{s}, \mathbf{v}|\alpha) &= -\alpha\mathbf{C}^T\mathbf{u} \end{aligned}$$

$\mathbf{u}, \mathbf{s}, \mathbf{v}$ is a stationary point if the KKT condition is satisfied:

$$
\begin{aligned}
&\nabla_{\mathbf{u}} P - \sum_i \lambda_i^{\mathcal{G}} \nabla_u \mathcal{G}_i + \lambda_i^{\mathcal{U}} \nabla_u \mathcal{U}_i = \mathbf{0} \\
&\nabla_{\mathbf{s}} P - \sum_i \lambda_i^{\mathcal{H}} \nabla_s \mathcal{H}_i + \lambda_i^{\mathcal{S}} \nabla_s \mathcal{S}_i = \mathbf{0} \\
&\nabla_{\mathbf{v}} P - \sum_i \lambda_i^{\mathcal{H}} \nabla_v \mathcal{H}_i + \lambda_i^{\mathcal{V}} \nabla_v \mathcal{V}_i = \mathbf{0} \\
&\lambda_i^{\mathcal{H}} \mathcal{H}_i = 0 \\
&\lambda_i^{\mathcal{G}} \mathcal{G}_i = 0 \\
&\lambda_i^{\mathcal{S}} \mathcal{S}_i = 0 \\
&\lambda_i^{\mathcal{U}} \mathcal{U}_i = 0 \\
&\lambda_i^{\mathcal{V}} \mathcal{V}_i = 0 \\
&s_i - c_i^T v + b_i \geq 0 \\
&1 - u_i \geq 0 \\
&u_i, v_i, s_i \geq 0 \\
&\lambda_i^{\mathcal{H}}, \lambda_i^{\mathcal{G}}, \lambda_i^{\mathcal{U}}, \lambda_i^{\mathcal{V}}, \lambda_i^{\mathcal{S}} \geq 0
\end{aligned}
\tag{5.32}
$$

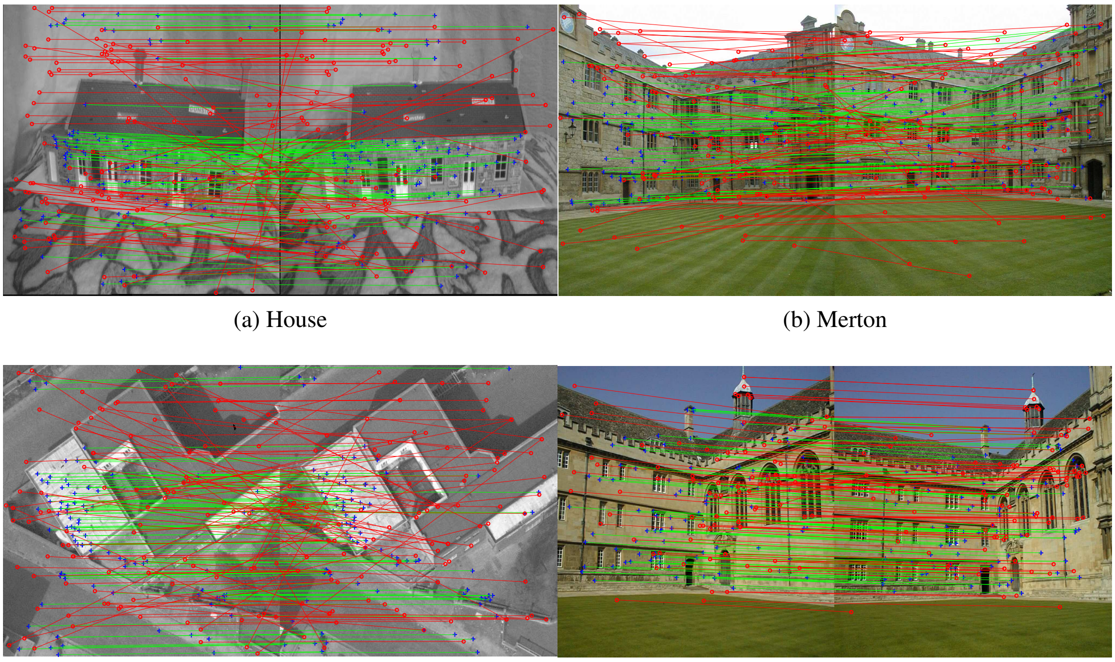
which is equivalent to:

$$
\begin{aligned}
-\alpha \mathbf{C} \mathbf{v} + \alpha \mathbf{b} + \mathbf{1} + \boldsymbol{\lambda}^{\mathcal{G}} - \boldsymbol{\lambda}^{\mathcal{U}} &= \mathbf{0} \\
\alpha \mathbf{1} - \boldsymbol{\lambda}^{\mathcal{H}} - \boldsymbol{\lambda}^{\mathcal{S}} &= \mathbf{0} \\
-\alpha \mathbf{C}^T \mathbf{u} + \mathbf{C}^T \boldsymbol{\lambda}^{\mathcal{H}} - \boldsymbol{\lambda}^{\mathcal{V}} &= \mathbf{0} \\
\lambda_i^{\mathcal{H}} (s_i - c_i^T v + b_i) &= 0 \\
\lambda_i^{\mathcal{G}} (1 - u_i) &= 0 \\
\lambda_i^{\mathcal{V}} v_i &= 0 \\
\lambda_i^{\mathcal{U}} u_i &= 0 \\
\lambda_i^{\mathcal{S}} s_i &= 0 \\
\lambda_i^{\mathcal{H}}, \lambda_i^{\mathcal{G}}, \lambda_i^{\mathcal{U}}, \lambda_i^{\mathcal{V}}, \lambda_i^{\mathcal{S}} &\geq 0 \\
u_i, v_i, s_i &\geq 0 \\
s_i - c_i^T v + b_i &\geq 0 \\
1 - u_i &\geq 0
\end{aligned}
\tag{5.33}
$$

By rearranging and substitution, (5.33) can be reduced to

$$\begin{aligned}
\mathbf{u}^T(-\alpha\mathbf{C}\mathbf{v} + \alpha\mathbf{b} + \mathbf{1} + \boldsymbol{\lambda}^{\mathcal{G}}) &= 0 \\
\mathbf{s}^T(\alpha\mathbf{1} - \boldsymbol{\lambda}^{\mathcal{H}}) &= 0 \\
\mathbf{v}^T(-\alpha\mathbf{C}^T\mathbf{u} + \mathbf{C}^T\boldsymbol{\lambda}^{\mathcal{H}}) &= 0 \\
(\boldsymbol{\lambda}^{\mathcal{H}})^T(\mathbf{s} - \mathbf{C}\mathbf{v} + \mathbf{b}) &= 0 \\
(\boldsymbol{\lambda}^{\mathcal{G}})^T(\mathbf{1} - \mathbf{u}) &= 0 \\
\mathbf{s} - \mathbf{C}\mathbf{v} + \mathbf{b} &\geq \mathbf{0} \\
\mathbf{1} - \mathbf{u} &\geq \mathbf{0} \\
\boldsymbol{\lambda}^{\mathcal{H}}, \boldsymbol{\lambda}^{\mathcal{G}}, \mathbf{u}, \mathbf{s}, \mathbf{v} &\geq \mathbf{0}
\end{aligned} \tag{5.34}$$

5.7.2 Qualitative results for real image data



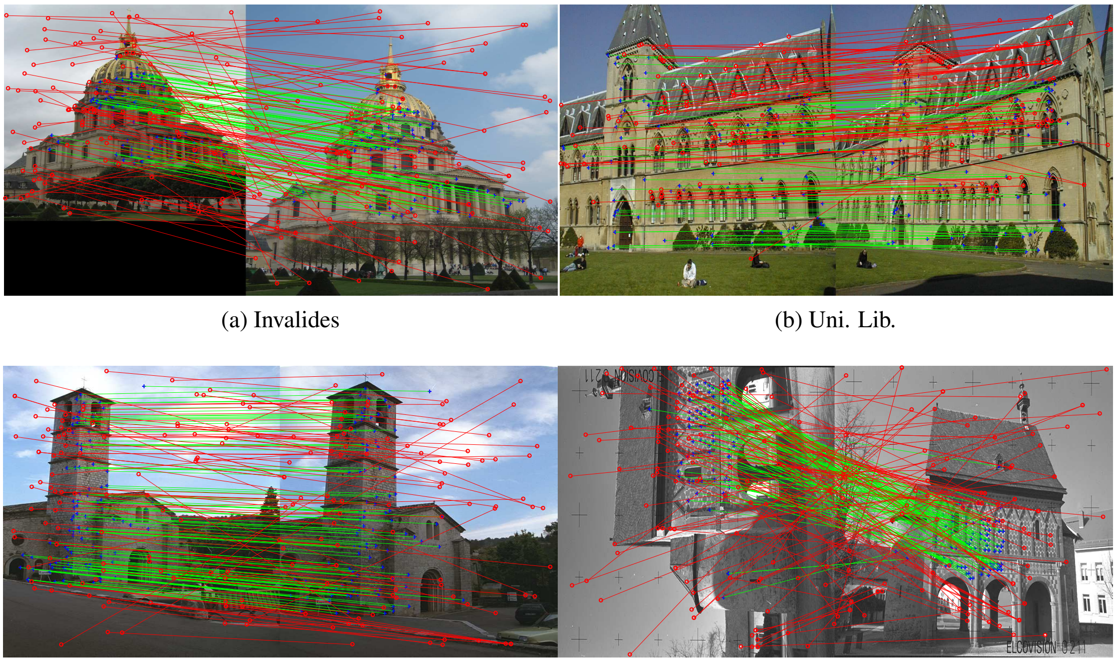
(a) House

(b) Merton

(c) Aerial

(d) Wadham

FIGURE 5.4: Qualitative results for Fundamental Matrix Estimation



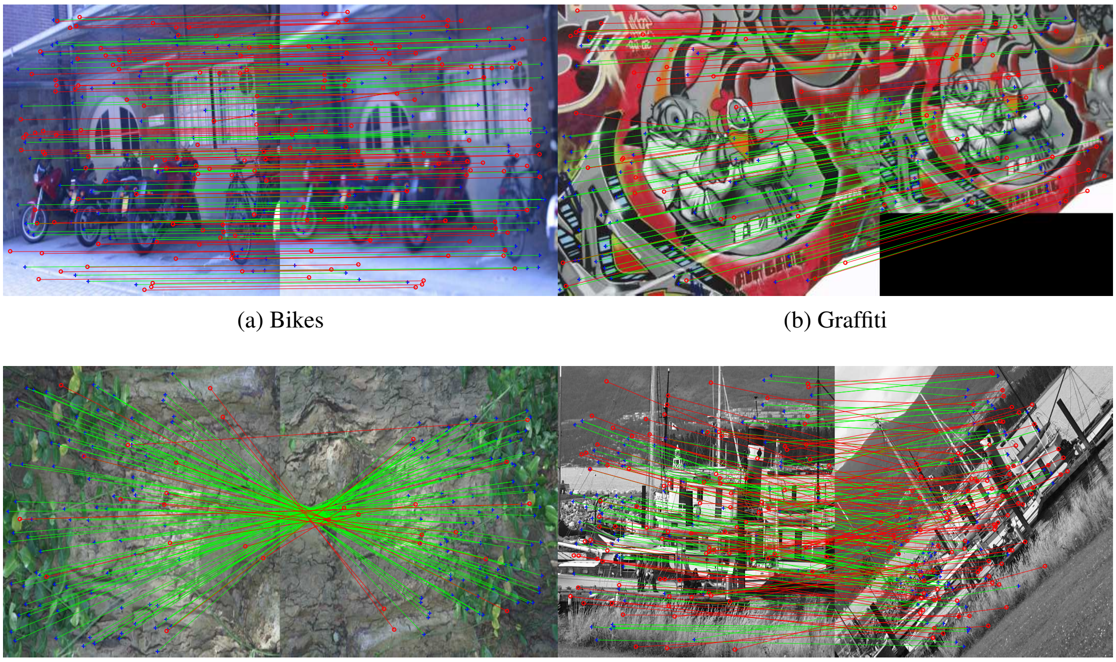
(a) Invalides

(b) Uni. Lib.

(c) Valbonne

(d) Kapel

FIGURE 5.5: Qualitative results for Homography Estimation



(a) Bikes

(b) Graffiti

(c) Bark

(d) Boat

FIGURE 5.6: Qualitative results for Affinity Estimation

Chapter 6

Deterministic Approximate Methods for Maximum Consensus Robust Fitting

The work contained in this chapter has been sumitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)

Huu Le, Tat-Jun Chin, Anders Eriksson and David Suter: Deterministic approximate Deterministic Approximate Methods for Maximum Consensus Robust Fitting.

The submitted paper is available at: https://arxiv.org/abs/1710.10003

Statement of Authorship

Title of Paper	Deterministic Approximate Methods for Maximum Consensus Robust Fitting		
Publication Status	□ Published ☑ Submitted for Publication	□ Accepted for Publication □ Unpublished and Unsubmitted w ork w ritten in manuscript style	
Publication Details	**Huu Le**, Tat-Jun Chin, Anders Eriksson and David Suter: Deterministic Approximate Methods for Maximum Consensus Robust Fitting. *Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* - arxiv: https://arxiv.org/abs/1710.10003		

Principal Author

Name of Principal Author (Candidate)	Huu Le		
Contribution to the Paper	- Proposed the general idea and developed the algorithms. - Implemented and tested the algorithm. - Conducted all the experiments to validate the performance of the proposed method. - Wrote the manuscript.		
Overall percentage (%)	80%		
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.		
Signature		Date	20th Dec 2017

Co-Author Contributions

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

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Tat-Jun Chin		
Contribution to the Paper	- Proposed the general research direction. - Supervised the development of this work. - Refined the manuscript.		
Signature		Date	20th Dec 2017

Name of Co-Author	Anders Eriksson		
Contribution to the Paper	- Supervised the development of ADMM-based algorithm. - Refined the manuscript.		
Signature		Date	20th Dec 2017

Name of Co-Author	David Suter		
Contribution to the Paper	- Supervised the development of this work. - Refined the manuscript.		
Signature		Date	20th Dec 2017

6.1 Introduction

Robust model fitting lies at the core of computer vision, due to the need of many fundamental tasks to deal with real-life data that are noisy and contaminated with outliers. To conduct robust model fitting, a robust fitting criterion is optimized w.r.t. a set of input measurements. Arguably the most popular robust criterion is *maximum consensus*, which aims to find the model that is consistent with the largest number of inliers, i.e., has the highest consensus.

Due to the critical importance of maximum consensus estimation, considerable effort has been put into devising algorithms for optimizing the criterion. A large amount of work occurred within the framework of hypothesize-and-verify methods, i.e., RANSAC [31] and its variants. Broadly speaking, these methods operate by fitting the model onto randomly sampled minimal subsets of the data, and returning the candidate with the largest inlier set. Improvements to the basic algorithm include guided sampling and speeding up the model verification step [21].

An important innovation is locally optimized RANSAC (LO-RANSAC) [23, 56]. As the name suggests, the objective of the method is to locally optimize RANSAC estimates. This is achieved by embedding in RANSAC an inner hypothesize-and-verify routine, which is triggered whenever the solution is updated in the outer loop. Different from the main RANSAC algorithm, the inner subroutine generates hypotheses from larger-than-minimal subsets sampled from the inlier set of the incumbent solution, in the hope of driving it towards an improved result.

Though efficient, there are fundamental shortcomings in the hypothesize-and-verify heuristic. Primarily, its randomized nature does not guarantee finding a good maximum consensus estimate; different runs may also give unpredictably different results. In LO-RANSAC, this weakness also manifests in the inner RANSAC routine, in that it is essentially a randomized trial-and-error procedure instead of a directed search to improve the estimate.

More recently, there is a growing number of globally optimal algorithms for consensus maximization [73, 96, 26, 59, 18]. The fundamental intractability of maximum consensus estimation, however, means that the global optimum can only be found by searching. The previous techniques respectively conduct branch-and-bound search [96, 59], tree search [18], or enumeration [73, 26]. Thus, global algorithms are practical only on problems with a small number of measurements and/or models of low dimensionality.

So far, what is sorely missing in the literature is an algorithm that lies in the middle ground between the above two extremes. Specifically, a maximum consensus algorithm that is *approximate* and *deterministic*, would add significantly to the robust fitting toolbox of computer vision.

In this paper, we contribute two such algorithms. Our starting point is to reformulate consensus maximization with linear complementarity constraints. We then develop an algorithm based on non-smooth penalty method supported by a Frank-Wolfe-style optimization scheme, and another algorithm based on the ADMM. In both algorithms, the calculation of the update step involves executing convex subproblems, which are efficient and enable the algorithms to handle realistic input sizes (hundreds to thousands of measurements). Further, our algorithms are naturally capable of handling the non-linear geometric residuals commonly used in computer vision [49, 51].

As will be demonstrated experimentally, our algorithms can significantly improve rough estimates obtained using an initial method, such as least squares or a fast randomized scheme such as RANSAC. Qualitative improvements achieved by our algorithms are also greater than that of LO-RANSAC, while incurring only marginally higher runtimes.

6.1.1 Deterministic robust fitting

M-estimators [46] are established class of robust statistical methods. The M-estimate is obtained by minimizing the sum of a set of ρ functions defined over the residuals, where ρ (e.g., the Huber norm) is responsible for discounting the effects of outliers. The primary technique for the minimization is iteratively reweighted least squares (IRLS). At each iteration, a weighted least squares problem is solved, where the weights are computed based on the previous estimate. Provided that ρ satisfies certain properties [95, 5], IRLS will deterministically reduce the cost until a local minimum is reached. This however precludes consensus maximization, since the corresponding ρ (a symmetric step function) is not positive definite and differentiable. Sec. 6.2.1 will further explore the characteristics of the maximum consensus objective.

Arguably, one can simply choose a robust ρ that works with IRLS and dispense with maximum consensus. However, another vital requirement for IRLS to be feasible is that the weighted least squares problem is efficiently solvable. This unfortunately is not the case for many of the geometric distances used in computer vision [49, 51, 40].

The above limitations with IRLS suggest that deterministic approximate methods for robust fitting remain an open problem, and our proposed algorithms should represent significant contributions towards this direction.

6.1.2 Road map

The paper is structured as follows:

- Sec. 6.2 defines the maximum consensus problem and characterizes the solution. It then describes the crucial reformulation with complementarity constraints.

- Sec. 6.3 describes the non-smooth penalty method.
- Sec. 6.4 describes the ADMM-based algorithm.
- Sec. 6.5 shows the applicability of our methods to estimation problems with quasiconvex geometric residuals.
- Sec. 6.6 demonstrates the effectiveness of our methods through a set of experiments with synthetic and real data on common computer vision applications.

This paper is an extension of the conference version [55], which proposed only the method based on non-smooth penalization. Sec. 6.6 of the present paper experimentally compares the new ADMM technique with the penalty method.

6.2 Problem definition

We develop our algorithms in the context of fitting linear models, before extending to models with geometric residuals in Sec. 6.5. Given a set of N measurements $\{\mathbf{x}_j, y_j\}_{j=1}^N$ for the linear model parametrized by vector $\boldsymbol{\theta} \in \mathbb{R}^d$, the goal of maximum consensus is to find the $\boldsymbol{\theta}$ that is consistent with as many of the input data as possible, i.e.,

$$\max_{\boldsymbol{\theta} \in \mathbb{R}^d} \quad \Psi(\boldsymbol{\theta}) \tag{6.1}$$

where the objective function

$$\Psi(\boldsymbol{\theta}) = \sum_{j=1}^{N} \mathbb{I}\left(|\mathbf{x}_j^T \boldsymbol{\theta} - y_j| \leq \epsilon\right) \tag{6.2}$$

is the *consensus* of $\boldsymbol{\theta}$. Here, $\mathbb{I}$ is the indicator function, which returns 1 if its input predicate is true, and 0 otherwise. The inlier-outlier dichotomy is achieved by comparing a residual $|\mathbf{x}_j^T \boldsymbol{\theta} - y_j|$ with the pre-defined threshold ϵ.

Expressing each inequality of the form $|\mathbf{x}_j^T \boldsymbol{\theta} - y_j| \leq \epsilon$ equivalently using the two linear constraints

$$\mathbf{x}_j^T \boldsymbol{\theta} - y_j \leq \epsilon, \quad -\mathbf{x}_j^T \boldsymbol{\theta} + y_j \leq \epsilon, \tag{6.3}$$

and collecting the data into the matrices

$$\begin{aligned} \mathbf{A} &= \left[\mathbf{x}_1, -\mathbf{x}_1, \dots, \mathbf{x}_N, -\mathbf{x}_N\right], \\ \mathbf{b} &= \left[\epsilon + y_1, \epsilon - y_1, \dots, \epsilon + y_N, \epsilon - y_N\right]^T, \end{aligned} \tag{6.4}$$

where $\mathbf{A} \in \mathbb{R}^{d \times M}$, $\mathbf{b} \in \mathbb{R}^M$ and $M = 2N$, we can redefine consensus as

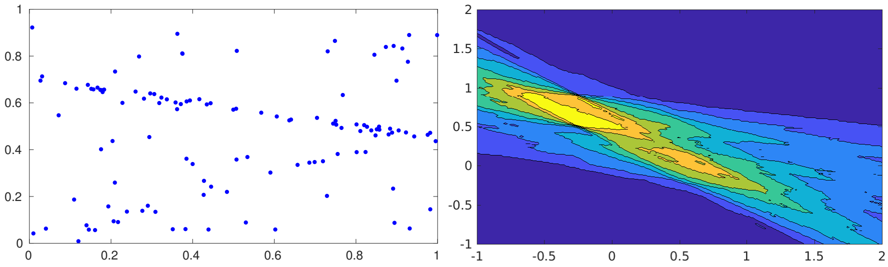
$$\Psi(\boldsymbol{\theta}) = \sum_{i=1}^{M} \mathbb{I}\left(\mathbf{a}_i^T \boldsymbol{\theta} - b_i \leq 0\right), \tag{6.5}$$

where $\mathbf{a}_i$ is the i-th column of $\mathbf{A}$ and b_i is the i-th element of $\mathbf{b}$. Plugging (6.5) instead of (6.2) into (6.1) yields an equivalent optimization problem, in the sense that both objective functions have the same maximizers.

Henceforth, we will be developing our maximum consensus algorithm based on (6.5) as the definition of consensus.

6.2.1 Characterizing the solution

What does Ψ look like? Consider the problem of robustly fitting a line onto a set of points $\{p_j, q_j\}_{j=1}^N$ on the plane. To apply formulation (6.1), set $\mathbf{x}_j = \begin{bmatrix} p_j & 1 \end{bmatrix}^T$ and $y_j = q_j$. The vector $\boldsymbol{\theta} \in \mathbb{R}^2$ then corresponds to the slope and intercept of the line. Fig. 6.1 plots $\Psi(\boldsymbol{\theta})$ for a sample point set $\{p_j, q_j\}_{j=1}^N$. As can be readily appreciated, Ψ is a piece-wise constant step function, owing to the thresholding and discrete counting operations in the calculation of consensus.



(a) (b)

FIGURE 6.1: (a) Sample point set $\{p_j, q_j\}_{j=1}^N$. (b) A plot of $\Psi(\boldsymbol{\theta})$ in $\mathbb{R}^2$ based on the sample point set. Each unique color represents a specific consensus value. Regions corresponding to the maximum consensus value are indicated in yellow.

We define the *global* or *exact solution* to (6.1) as the vector $\boldsymbol{\theta}^*$ such that $\Psi(\boldsymbol{\theta}^*) \geq \Psi(\boldsymbol{\theta})$ for all $\boldsymbol{\theta} \in \mathbb{R}^d$. In general, $\boldsymbol{\theta}^*$ is not unique, and can only be identified by searching. Recall that a *local solution* of an unconstrained optimization problem

$$\max_{\boldsymbol{\theta} \in \mathbb{R}^d} \quad f(\boldsymbol{\theta}) \tag{6.6}$$

is a vector $\hat{\boldsymbol{\theta}}$ such that there exists a neighborhood $\mathcal{N} \subset \mathbb{R}^d$ of $\hat{\boldsymbol{\theta}}$ where $f(\hat{\boldsymbol{\theta}}) \geq f(\boldsymbol{\theta})$ for all $\boldsymbol{\theta} \in \mathcal{N}$ [69, Chap. 2]. By this definition, since Ψ is piece-wise constant, all $\boldsymbol{\theta} \in \mathbb{R}^d$ are local solutions to (6.1). The concept of *local optimality* is thus not meaningful in the context of consensus maximization. Indeed, the lack of gradient information in Ψ complicates the usage of standard nonlinear optimization schemes, which strive for local optimality, on problem (6.1) (cf. IRLS).

Unlike nonlinear optmization methods or IRLS, the proposed algorithms do not depend on the existence of gradients; instead, our algorithms solve derived convex subproblems to deterministically and efficiently update an approximate solution to the maximum consensus problem. As mentioned in the introduction, such techniques have not been considered previously in the literature.

6.2.2 Reformulation with complementarity constraints

Introducing indicator variables $\mathbf{u} \in \{0,1\}^M$ and slack variables $\mathbf{s} \in \mathbb{R}^M$, we first reformulate (6.1) equivalently as an outlier count minimization problem

$$\begin{aligned} \min_{\mathbf{u}\in\{0,1\}^M,\, \mathbf{s}\in\mathbb{R}^M,\, \boldsymbol{\theta}\in\mathbb{R}^d} \quad & \sum_i u_i && \text{(6.7a)}\\ \text{subject to} \quad & s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i \geq 0, && \text{(6.7b)}\\ & u_i(s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i) = 0, && \text{(6.7c)}\\ & s_i(1-u_i) = 0, && \text{(6.7d)}\\ & s_i \geq 0. && \text{(6.7e)} \end{aligned}$$

Intuitively, s_i must be non-zero if the i-th datum is an outlier w.r.t. $\boldsymbol{\theta}$; in this case, u_i must be set to 1 to satisfy (6.7d). In turn, (6.7c) forces the quantity $(s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i)$ to be zero. Conversely, if the i-th datum is an inlier w.r.t. $\boldsymbol{\theta}$, then s_i is zero, u_i is zero and $(s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i)$ is non-zero. Observe, therefore, that (6.7c) and (6.7d) implement *complementarity* between u_i and $(s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i)$.

Note also that, due to the objective function and condition (6.7d), the indicator variables can be relaxed without impacting the optimum, leading to the equivalent problem

$$\begin{aligned} \min_{\mathbf{u},\mathbf{s}\in\mathbb{R}^M,\, \boldsymbol{\theta}\in\mathbb{R}^d} \quad & \sum_i u_i && \text{(6.8a)}\\ \text{subject to} \quad & s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i \geq 0, && \text{(6.8b)}\\ & u_i(s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i) = 0, && \text{(6.8c)}\\ & s_i(1-u_i) = 0, && \text{(6.8d)}\\ & 1 - u_i \geq 0, && \text{(6.8e)}\\ & s_i, u_i \geq 0. && \text{(6.8f)} \end{aligned}$$

This, however, does not make (6.8) tractable, since (6.8c) and (6.8d) are bilinear in the unknowns.

To re-express (6.8) using only positive variables, define

$$\mathbf{v} = \begin{bmatrix} \boldsymbol{\theta} + \gamma \mathbf{1} \\ \gamma \end{bmatrix}, \quad \mathbf{c}_i = \begin{bmatrix} \mathbf{a}_i^T & -\mathbf{a}_i^T \mathbf{1} \end{bmatrix}^T, \tag{6.9}$$

where both are real vectors of length $(d+1)$. Problem (6.8) can then be reformulated equivalently as

$$\begin{aligned} \min_{\mathbf{u},\mathbf{s} \in \mathbb{R}^M,\ \mathbf{v} \in \mathbb{R}^{d+1}} \quad & \sum_i u_i \\ \text{subject to} \quad & s_i - \mathbf{c}_i^T \mathbf{v} + b_i \geq 0, \\ & u_i(s_i - \mathbf{c}_i^T \mathbf{v} + b_i) = 0, \\ & s_i(1 - u_i) = 0, \\ & 1 - u_i \geq 0, \\ & s_i, u_i, v_i \geq 0. \end{aligned} \tag{6.10}$$

Given a solution $\hat{\mathbf{u}}$, $\hat{\mathbf{s}}$ and $\hat{\mathbf{v}}$ to (6.10), the corresponding solution $\hat{\boldsymbol{\theta}}$ to (6.8) can be obtained by simply subtracting the last element of $\hat{\mathbf{v}}$ from its first-d elements.

While the relaxation does not change the fundamental intractability of (6.1), that all the variables are now continuous allows to bring a broader class of optimization techniques to bear on the problem—as we will show next.

6.3 Non-smooth penalty method

Our first deterministic refinement algorithm is based on the technique of non-smooth penalization [69, Sec. 17.2]. Incorporating the equality constraints in (6.10) into the cost function as a penalty term, we obtain the penalty problem

$$\begin{aligned} \min_{\mathbf{u},\mathbf{s} \in \mathbb{R}^M, \mathbf{v} \in \mathbb{R}^{d+1}} \quad & \sum_i u_i + \alpha \left[u_i(s_i - \mathbf{c}_i^T \mathbf{v} + b_i) + s_i(1 - u_i) \right] \\ \text{s.t.} \quad & s_i - \mathbf{c}_i^T \mathbf{v} + b_i \geq 0, \\ & 1 - u_i \geq 0, \\ & s_i, u_i, v_i \geq 0. \end{aligned} \tag{6.11}$$

The constant $\alpha \geq 0$ is called the *penalty parameter*. Intuitively, the penalty term discourages solutions that violate the complementarity constraints, and the strength of the penalization is controlled by α. Observe also that the remaining constraints in (6.101) define a convex domain.

Henceforth, to reduce clutter, we sometimes use

$$\mathbf{z} = \left[\mathbf{u}^T \; \mathbf{s}^T \; \mathbf{v}^T\right]^T. \tag{6.12}$$

The cost function in (6.101) can be rewritten as

$$P(\mathbf{z} \mid \alpha) = F(\mathbf{z}) + \alpha Q(\mathbf{z}), \tag{6.13}$$

where $F(\mathbf{z}) = \|\mathbf{u}\|_1$ and

$$\begin{aligned} Q(\mathbf{z}) &= \sum_i u_i(s_i - \mathbf{c}_i^T\mathbf{v} + b_i) + s_i(1 - u_i) && (6.14)\\ &= \sum_i s_i - u_i(\mathbf{c}_i^T\mathbf{v} - b_i). && (6.15) \end{aligned}$$

Note that each summand in $Q(\mathbf{z})$ is non-negative, and the penalty term can be viewed as the ℓ_1-norm (a non-smooth function) of the *complementarity residual* vector

$$\mathbf{r}(\mathbf{z}) = \begin{bmatrix} r_1(\mathbf{z}) & \dots & r_M(\mathbf{z}) \end{bmatrix}^T, \tag{6.16}$$

where

$$r_i(\mathbf{z}) = s_i - u_i(\mathbf{c}_i^T\mathbf{v} - b_i). \tag{6.17}$$

In Sec. 6.3.2, we will devise a consensus maximization algorithm based on solving a sequence of the penalty problem (6.101) with increasing values of α. Before that, in Sec. 6.3.1, we will discuss a method to solve the penalty problem for a given (constant) α.

6.3.1 Solving the penalty problem

6.3.1.1 Necessary optimality conditions

Although $P(\mathbf{z} \mid \alpha)$ is quadratic, problem (6.101) is non-convex. However, it can be shown that (6.101) has a vertex solution, i.e., a solution that is an extreme point of the convex set

$$\begin{aligned}\mathcal{P} = \{\mathbf{z} \in \mathbb{R}^{2M+d+1} \, |s_i - \mathbf{c}_i^T\mathbf{v} + b_i &\geq 0, \\ 1 - u_i &\geq 0, \\ s_i, u_i, v_i &\geq 0, \\ i &= 1, \dots, M\}\end{aligned} \tag{6.18}$$

To minimize clutter, rewrite

$$\mathcal{P} = \{\mathbf{z} \in \mathbb{R}^{2M+d+1} \mid \mathbf{Mz} + \mathbf{q} \geq \mathbf{0}, \;\; \mathbf{z} \geq \mathbf{0}\}, \tag{6.19}$$

where

$$\begin{aligned}\mathbf{M} &= \begin{bmatrix} \mathbf{0} & \mathbf{I} & -\mathbf{C} \\ -\mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \\ \mathbf{C} &= \begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \dots & \mathbf{c}_M \end{bmatrix}^T, \\ \mathbf{q} &= \begin{bmatrix} \mathbf{b}^T & \mathbf{1}^T \end{bmatrix}^T;\end{aligned} \tag{6.20}$$

(we do not define the sizes of $\mathbf{I}$, $\mathbf{0}$ and $\mathbf{1}$, but the sizes can be worked out based on the context). To begin, observe that the minima of (6.101) obey the KKT conditions [69, Chap. 12]

$$\begin{aligned}&\mathbf{u}^T(-\alpha\mathbf{Cv} + \alpha\mathbf{b} + \mathbf{1} + \boldsymbol{\lambda}^{\mathcal{G}}) = 0, \\ &\mathbf{s}^T(\alpha\mathbf{1} - \boldsymbol{\lambda}^{\mathcal{H}}) = 0, \\ &\mathbf{v}^T(-\alpha\mathbf{C}^T\mathbf{u} + \mathbf{C}^T\boldsymbol{\lambda}^{\mathcal{H}}) = 0, \\ &(\boldsymbol{\lambda}^{\mathcal{H}})^T(\mathbf{s} - \mathbf{Cv} + \mathbf{b}) = 0, \\ &(\boldsymbol{\lambda}^{\mathcal{G}})^T(\mathbf{1} - \mathbf{u}) = 0, \\ &\mathbf{s} - \mathbf{Cv} + \mathbf{b} \geq \mathbf{0}, \\ &\mathbf{1} - \mathbf{u} \geq \mathbf{0}, \\ &\boldsymbol{\lambda}^{\mathcal{H}}, \boldsymbol{\lambda}^{\mathcal{G}}, \mathbf{u}, \mathbf{v}, \mathbf{s} \geq \mathbf{0},\end{aligned} \tag{6.21}$$

where $\boldsymbol{\lambda}^{\mathcal{H}} = [\lambda_1^{\mathcal{H}} \, \dots \, \lambda_M^{\mathcal{H}}]^T$ and $\boldsymbol{\lambda}^{\mathcal{G}} = [\lambda_1^{\mathcal{G}} \, \dots \, \lambda_M^{\mathcal{G}}]^T$ are the Lagrange multipliers for the first two types of constraints in (6.101); see the supplementary material for details.

By rearranging, the KKT conditions (6.105) can be summarized by the following relations

$$\mathbf{M}'\mathbf{z}' + \mathbf{q}' \geq \mathbf{0}, \;\; \mathbf{z}' \geq \mathbf{0}, \;\; (\mathbf{z}')^T(\mathbf{M}'\mathbf{z}' + \mathbf{q}') = 0, \tag{6.22}$$

where

$$\begin{aligned}\mathbf{z}' &= \begin{bmatrix}\mathbf{z}^T & (\boldsymbol{\lambda}^{\mathcal{H}})^T & (\boldsymbol{\lambda}^{\mathcal{G}})^T\end{bmatrix}^T, \\ \mathbf{M}' &= \begin{bmatrix}\mathbf{0} & \mathbf{0} & -\alpha\mathbf{C} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ -\alpha\mathbf{C}^T & \mathbf{0} & \mathbf{0} & \mathbf{C}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & -\mathbf{C} & \mathbf{0} & \mathbf{0} \\ -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\end{bmatrix}, \\ \mathbf{q}' &= \begin{bmatrix}(\alpha\mathbf{b} + \mathbf{1})^T & \alpha\mathbf{1}^T & \mathbf{0}^T & \mathbf{b}^T & \mathbf{1}^T\end{bmatrix}^T.\end{aligned} \tag{6.23}$$

Finding a feasible $\mathbf{z}'$ for (6.22) is an instance of a *linear complementarity problem (LCP)* [66]. Define the convex set

$$\mathcal{P}' = \{\mathbf{z}' \in \mathbb{R}^{4M+d+1} \mid \mathbf{M}'\mathbf{z}' + \mathbf{q}' \geq \mathbf{0}, \;\; \mathbf{z}' \geq \mathbf{0}\}. \tag{6.24}$$

We invoke the following result from [66, Lemma 2].

Theorem 6.1. *If the LCP defined by the constraints* (6.22) *has a solution, then it has a solution at a vertex of* $\mathcal{P}'$.

Theorem 6.1 implies that the KKT points of (6.101) (including the solutions of the problem) occur at the vertices of $\mathcal{P}'$. This also implies that (6.101) has a vertex solution, viz.:

Theorem 6.2. *For any vertex*

$$\mathbf{z}'_v = [\mathbf{z}_v^T \;\; (\boldsymbol{\lambda}_v^{\mathcal{H}})^T \;\; (\boldsymbol{\lambda}_v^{\mathcal{G}})^T)]^T \tag{6.25}$$

of $\mathcal{P}'$, $\mathbf{z}_v$ *is a vertex of* $\mathcal{P}$.

Proof. If $\mathbf{z}'_v$ is a vertex of $\mathcal{P}'$, then, there is a diagonal matrix $\mathbf{E}$ such that

$$\mathbf{M}'\mathbf{E}\mathbf{z}'_v + \mathbf{q}' - \boldsymbol{\gamma}' = \mathbf{0}, \tag{6.26}$$

where $\mathbf{E}_{i,i} = 1$ if the i-th column of $\mathbf{M}'$ appears in the basic solution corresponding to vertex $\mathbf{z}'_v$, and $\mathbf{E}_{i,i} = 0$ otherwise (the non-negative vector $\boldsymbol{\gamma}'$ contains the values of additional slack variables to convert the constraints in $\mathcal{P}'$ into standard form). Let $\mathbf{M}'_J$ be the last-$2M$ rows of

$\mathbf{M}'$. Then,

$$\mathbf{M}'_J \mathbf{E} \mathbf{z}'_v + \begin{bmatrix} \mathbf{b}^T & \mathbf{1}^T \end{bmatrix}^T - \boldsymbol{\gamma}'_J = \mathbf{0}, \tag{6.27}$$

where $\boldsymbol{\gamma}'_J$ is the last-$2M$ elements of $\boldsymbol{\gamma}'$. Note that, since the right-most $2M \times 2M$ submatrix of $\mathbf{M}'_J$ is a zero matrix (see (6.23)), then

$$\mathbf{M}'_J \mathbf{E}_K \mathbf{z}_v + \begin{bmatrix} \mathbf{b}^T & \mathbf{1}^T \end{bmatrix}^T - \boldsymbol{\gamma}'_J = \mathbf{0}, \tag{6.28}$$

where $\mathbf{E}_K$ is the first-$(2M+d+1)$ columns of $\mathbf{E}$. Since $\mathbf{M}'_J \mathbf{E}_K = \mathbf{M}$, then (6.28) implies that $\mathbf{z}_v$ is a vertex of $\mathcal{P}$. □ □

6.3.1.2 Frank-Wolfe algorithm

Theorem 6.2 suggests an approach to solve (6.101) by searching for a vertex solution. Further, note that for a fixed $\mathbf{u}$, (6.101) reduces to an LP. Conversely, for fixed $\mathbf{s}$ and $\mathbf{v}$, (6.101) is also an LP. This advocates alternating between optimizing subsets of the variables using LPs. Algorithm 6.1 summarizes the method, which is in fact a special case of the Frank-Wolfe method [32] for non-convex quadratic minimization.

Algorithm 6.1 Frank-Wolfe method for (6.101).

```
Require: Data $\{\mathbf{c}_i, b_i\}_{i=1}^{M}$, penalty value $\alpha$, initial solution $\mathbf{u}^{(0)}$, $\mathbf{v}^{(0)}$, $\mathbf{s}^{(0)}$, threshold $\delta$.
 1: $P^{(0)} \leftarrow P(\mathbf{u}^{(0)}, \mathbf{s}^{(0)}, \mathbf{v}^{(0)} \mid \alpha)$.
 2: $t \leftarrow 0$.
 3: while true do
 4:     $t \leftarrow t + 1$.
 5:     $\mathbf{s}^{(t)}, \mathbf{v}^{(t)} \leftarrow \arg\min_{\mathbf{s},\mathbf{v}} P(\mathbf{u}^{(t-1)}, \mathbf{s}, \mathbf{v} \mid \alpha)$ s.t. $\mathcal{P}$.
 6:     $\mathbf{u}^{(t)} \leftarrow \arg\min_{\mathbf{u}} P(\mathbf{u}, \mathbf{s}^{(t)}, \mathbf{v}^{(t)} \mid \alpha)$ s.t. $\mathcal{P}$.
 7:     $P^{(t)} \leftarrow P(\mathbf{u}^{(t)}, \mathbf{s}^{(t)}, \mathbf{v}^{(t)} \mid \alpha)$.
 8:     if $|P^{(t-1)} - P^{(t)}| \leq \delta$ then
 9:         Break.
10:     end if
11: end while
12: return $\mathbf{u}^{(t)}, \mathbf{v}^{(t)}, \mathbf{s}^{(t)}$.
```

Theorem 6.3. *In a finite number of steps, Algorithm 6.1 converges to a KKT point of* (6.101).

Proof. The set of constraints $\mathcal{P}$ can be decoupled into the two disjoint subsets

$$\mathcal{P} = \mathcal{P}_1 \times \mathcal{P}_2, \tag{6.29}$$

where $\mathcal{P}_1$ involves only $\mathbf{s}$ and $\mathbf{v}$, and $\mathcal{P}_2$ is the complement of $\mathcal{P}_1$. With $\mathbf{u}$ fixed in Line 5, the LP converges to a vertex of $\mathcal{P}_1$. Similarly, with $\mathbf{s}$ and $\mathbf{v}$ fixed in Line 6, the LP converges to a vertex in $\mathcal{P}_2$. Each intermediate solution $\mathbf{u}^{(t)}, \mathbf{v}^{(t)}, \mathbf{s}^{(t)}$ is thus a vertex of $\mathcal{P}$ or a KKT point of (6.101). Since each LP must reduce or maintain $P(\mathbf{z} \mid \alpha)$ which is bounded below, the process terminates in finite steps. □ □

Analysis of update steps A closer look reveals the LP in Line 5 (Algorithm 6.1) to be

$$\begin{aligned} \min_{\mathbf{s},\mathbf{v}} \quad & \sum_i s_i - u_i(\mathbf{c}_i^T \mathbf{v} - b_i) \\ \text{s.t.} \quad & s_i - \mathbf{c}_i^T \mathbf{v} + b_i \geq 0, \\ & s_i, v_i \geq 0, \end{aligned} \tag{LP1}$$

and the LP in Line 6 (Algorithm 6.1) to be

$$\begin{aligned} \min_{\mathbf{u}} \quad & \sum_i u_i \left[1 - \alpha(\mathbf{c}_i^T \mathbf{v} - b_i)\right] \\ \text{s.t.} \quad & 0 \leq u_i \leq 1. \end{aligned} \tag{LP2}$$

Observe that LP2 can be solved in closed form and it also drives $\mathbf{u}$ to integrality: if $[1-\alpha(\mathbf{c}_i^T\mathbf{v} - b_i)] \leq 0$, set $u_i = 1$, else, set $u_i = 0$. Further, LP1 can be seen as "weighted" ℓ_1-norm minimization, with $\mathbf{u}$ being the weights. Intuitively, therefore, Algorithm 6.1 alternates between residual minimization (LP1) and inlier-outlier dichotomization (LP2).

6.3.2 Main algorithm

Intuitively, if the penalty parameter α is small, Algorithm 6.1 will pay more attention to minimizing $\sum_i u_i$ and less attention to ensuring that the optimized variables are feasible w.r.t. the original problem (6.10). Conversely, if α is large, the complementarity residual $Q(\mathbf{z})$ will be reduced more aggressively, thus the optimized $\mathbf{z}$ tends to be "more feasible". If α is sufficiently large, $Q(\mathbf{z})$ will be reduced to zero, and any movement to attempt to reduce $\sum_i u_i$ will not pay-off, thus preserving the feasibility of $\mathbf{z}$— Section 6.3.2.1 will formally establish this condition.

The above observations argue for a deterministic consensus maximization algorithm based on solving (6.101) for progressively larger α's; see Algorithm 6.2. For each α, our method solves (6.101) using Algorithm 6.1. The solution $\hat{\mathbf{z}}$ for a particular α is then used to initialize Algorithm 6.1 for the next larger α. The sequence terminates when the complementarity residual $Q(\mathbf{z})$ vanishes or becomes insignificant.

It is worthwhile to note that typical non-smooth penalty functions cannot be easily minimized (e.g., no gradient information). In our case, however, we exploited the special property of (6.101) (Sec. 6.3.1.1) to enable efficient minimization.

Algorithm 6.2 Non-smooth penalty method for solving (6.10).

```
Require: Data $\{\mathbf{c}_i, b_i\}_{i=1}^M$, initial model parameter $\boldsymbol{\theta}$, initial penalty value $\alpha$, increment rate $\kappa$,
    threshold $\delta$.
$\mathbf{v} \leftarrow \left[(\boldsymbol{\theta} + |\min_j(\theta_j)|\mathbf{1})^T \;\; |\min_j(\theta_j)|\right]^T$.
$\mathbf{u} \leftarrow \mathbb{I}(\mathbf{C}\mathbf{v} - \mathbf{b} > 0)$.
$\mathbf{s} \leftarrow \mathbf{u} \odot (\mathbf{C}\mathbf{v} - \mathbf{b})$.
while true do
    $\mathbf{u}, \mathbf{s}, \mathbf{v} \leftarrow FW(\{\mathbf{c}_i, b_i\}_{i=1}^M, \alpha, \mathbf{u}, \mathbf{s}, \mathbf{v})$. /*Algo. 6.1.*/
    if $Q(\mathbf{z}) \leq \delta$ then
        Break.
    end if
    $\alpha \leftarrow \kappa \cdot \alpha$.
end while
return $\mathbf{u}, \mathbf{s}, \mathbf{v}$.
```

6.3.2.1 Convergence

Theorem 6.4. *If α is sufficiently large, Algorithm 6.2 converges to a point $\hat{\mathbf{z}}$ where $Q(\hat{\mathbf{z}}) = 0$, i.e., $\hat{\mathbf{z}}$ is a feasible solution of problem* (6.10).

Proof. Let $\hat{\mathbf{s}}$ and $\hat{\mathbf{v}}$ be the solution of LP1 (for a fixed $\hat{\mathbf{u}}$ from the previous iteration). When updating $\mathbf{u}$ in LP2, for each constraint i, the possible outcomes for u_i are:

- If $\mathbf{c}_i^T\hat{\mathbf{v}} - b_i \leq 0$: We say that the i-th constraint is consistent with $\hat{\mathbf{v}}$. LP2 will set u_i to 0 regardless of α.
- If $\mathbf{c}_i^T\hat{\mathbf{v}} - b_i > 0$: We say that the i-th constraint violates $\hat{\mathbf{v}}$. LP2 will set u_i according to

$$u_i = \begin{cases} 0 & \text{if } \; 1 - \alpha(\mathbf{c}_i^T\hat{\mathbf{v}} - b_i) \geq 0, \\ 1 & \text{if } \; 1 - \alpha(\mathbf{c}_i^T\hat{\mathbf{v}} - b_i) < 0. \end{cases}$$

If α is large enough, then LP2 will set $u_i = 1$ for all the violating constraints. Given a $\hat{\mathbf{u}}$ that was obtained under such a sufficiently large α in LP2, in the subsequent invocation of LP1, the minimal cost of 0 can be obtained by maintaining the previous $\hat{\mathbf{v}}$ and setting

$$\hat{s}_i = \begin{cases} 0 & \text{if } \; \hat{u}_i = 0, \\ \mathbf{c}_i^T\hat{\mathbf{v}} - b_i & \text{if } \; \hat{u}_i = 1. \end{cases}$$

Recognizing that the objective function of LP1 is equal to $Q(\mathbf{z})$ completes the proof. □ □

6.3.2.2 Initialization

Algorithm 6.2 requires the initialization of $\mathbf{u}$, $\mathbf{s}$ and $\mathbf{v}$. For consensus maximization, it is more natural to initialize the model parameters $\boldsymbol{\theta}$, which in turn gives values to $\mathbf{v}$, $\mathbf{s}$ and $\mathbf{u}$. In our work, we initialize $\boldsymbol{\theta}$ as the least squares solution, or by executing RANSAC (Sec. 6.6 will compare the results of these two different initialization methods).

Other required inputs are the initial penalty parameter α and the increment rate κ. These values affect the convergence speed of Algorithm 6.2. To avoid bad minima, we set α and κ conservatively, e.g., $\alpha \in [1, 10]$, $\kappa \in [1,\ 5]$. As we will demonstrate in Sec. 6.6, these settings enable Algorithm 6.2 to find very good solutions at competitive runtimes.

6.4 ADMM-based algorithm

Our second technique derives from the class of proximal splitting algorithms [11]. Specifically, we apply the ADMM to construct a deterministic approximate algorithm for our target problem (6.10). The ADMM was originally developed for convex optimization problems. However, its use for nonconvex nonsmooth optimization has been investigated recently, with strong convergence results [43, 89]. While ADMM has recently found usage in several geometric vision problems, e.g., bundle adjustment [30, 28], triangulation [29], its application to robust fitting is relatively unexplored.

6.4.1 ADMM formulation

The specific version of ADMM used in our work is *consensus ADMM* [11], where the term "consensus" takes a different meaning[1] than ours—to avoid confusion, we will simply call the technique "ADMM". To the original problem (6.10), where the objective function has M summands and the original variables are $\mathbf{z} = [\mathbf{u}^T\ \mathbf{s}^T\ \mathbf{v}^T]^T \in \mathbb{R}^{2M+d+1}$, introduce M auxilary parameter vectors $\mathbf{z}'_1, \dots, \mathbf{z}'_M$, where

$$\mathbf{z}'_i = [u'_i\ s'_i\ (\mathbf{v}'_i)^T]^T \in \mathbb{R}^{d+3}, \tag{6.30}$$

as well as the "coupling" parameter vector

$$\mathbf{z}_C = [\mathbf{s}_C^T\ \mathbf{v}_C^T]^T \in \mathbb{R}^{M+d}. \tag{6.31}$$

[1] Consensus ADMM is a version commonly used for distributed optimization [11]. For brevity, we do not explore distributed optimization in our work, though our algorithm is amenable to such a scheme.

Then, rewrite (6.10) as

$$
\begin{aligned}
\min_{\mathbf{z},\{\mathbf{z}'_i\},\mathbf{z}_C} \quad & \sum_i \left[u'_i + \mathbb{I}_B(\mathbf{z}'_i)\right] + \mathbb{I}_C(\mathbf{z}_C) && \text{(6.32a)}\\
\text{s.t.} \quad & \mathbf{u} = \mathbf{u}', && \text{(6.32b)}\\
& \mathbf{s} = \mathbf{s}' = \mathbf{s}_C, && \text{(6.32c)}\\
& \mathbf{v} = \mathbf{v}'_i = \mathbf{v}_C, && \text{(6.32d)}
\end{aligned}
$$

where $\mathbb{I}_\mathbf{B}$ is an indicator function that enforces the bilinear constraints

$$
\mathbb{I}_B(\mathbf{z}'_i) = \begin{cases} 0 & \text{if} \begin{cases} u'_i(s'_i - \mathbf{c}_i^T\mathbf{v}'_i + b_i) = 0, \\ s'_i(1-u'_i) = 0, \\ u'_i \in \{0,1\}, \end{cases} \\ \infty & \text{otherwise,} \end{cases} \tag{6.33}
$$

and $\mathbb{I}_C$ is an indicator function that enforces $\mathbf{z}_C$ to statisfy the convex constraints

$$
\mathbb{I}_C(\mathbf{z}_C) = \begin{cases} 0 & \text{if} \begin{cases} \mathbf{s}_C - \mathbf{C}\mathbf{v}_C + \mathbf{b} \geq \mathbf{0}, \\ \mathbf{s}_C, \mathbf{v}_C \geq \mathbf{0}, \end{cases} \\ \infty & \text{otherwise.} \end{cases} \tag{6.34}
$$

Note that the objective function (6.32a) is a composition of $M+1$ totally separate subfunctions, where each subfunction of the form $u'_i + \mathbb{I}_B(\mathbf{z}'_i)$ involves only $\mathbf{z}'_i$, and the final subfunction $\mathbb{I}_C(\mathbf{z}_C)$ involves only $\mathbf{z}_C$. Intuitively, the constraints (6.32b), (6.32c), and (6.32d) ensure that the auxiliary and the original variables must converge to the same point, and hence are referred to as "coupling constraints". It can thus be appreciated that problem (6.32) is identical to problem (6.10), in that solving (6.32) results in the same optimum as (6.10). The benefit of the decomposition is that the problem can be solved by iteratively solving smaller subproblems which are convex, as we elaborate in the next subsection.

It can further be realized that the solution of the problem (6.32) does not change if the term $\|\mathbf{u}\|^2$ is added to the cost function (6.32a). Thus, to aid the convergence of our proposed algorithm (refer to the supplementary material for more details), the solution of (6.32) can be obtained by solving the following problem:

$$\min_{\mathbf{z},\{\mathbf{z}'_i\},\mathbf{z}_C} \quad \sum_i \left[u'_i + \mathbb{I}_B(\mathbf{z}'_i)\right] + \mathbb{I}_C(\mathbf{z}_C) + \|\mathbf{u}\|^2 \tag{6.35a}$$

$$\text{s.t.} \quad \mathbf{u} = \mathbf{u}', \tag{6.35b}$$

$$\mathbf{s} = \mathbf{s}' = \mathbf{s}_C, \tag{6.35c}$$

$$\mathbf{v} = \mathbf{v}'_i = \mathbf{v}_C, \tag{6.35d}$$

6.4.1.1 Augmented Lagrangian

Now consider the the augmented Lagrangian of (6.35)

$$\begin{aligned} \mathcal{L}_\rho(\mathbf{z},\{\mathbf{z}'_i\},\mathbf{z}_C,\boldsymbol{\lambda}) = & \sum_i \left[u'_i + \mathbb{I}_B(\mathbf{z}'_i)\right] + \mathbb{I}_C(\mathbf{z}_C) + \|\mathbf{u}\|^2 \\ & + \rho(\|\mathbf{u}' - \mathbf{u} + \boldsymbol{\lambda}^{\mathbf{u}}\|_2^2 - \|\boldsymbol{\lambda}^{\mathbf{u}}\|_2^2) \\ & + \rho(\|\mathbf{s}' - \mathbf{s} + \boldsymbol{\lambda}^{\mathbf{s}}\|_2^2 - \|\boldsymbol{\lambda}^{\mathbf{s}}\|_2^2) \\ & + \rho(\|\mathbf{s}_C - \mathbf{s} + \boldsymbol{\lambda}^{\mathbf{s}}_C\|_2^2 - \|\boldsymbol{\lambda}^{\mathbf{s}}_C\|_2^2) \\ & + \rho(\|\mathbf{v}_C - \mathbf{v} + \boldsymbol{\lambda}^{\mathbf{v}}_C\|_2^2 - \|\boldsymbol{\lambda}^{\mathbf{v}}_C\|_2^2) \\ & + \rho\sum_i(\|\mathbf{v}'_i - \mathbf{v} + \boldsymbol{\lambda}^{\mathbf{v}}_i\|_2^2 - \|\boldsymbol{\lambda}^{\mathbf{v}}_i\|_2^2), \end{aligned} \tag{6.36}$$

where

$$\mathbf{u}' = \begin{bmatrix} u'_1 & \dots u'_M \end{bmatrix}^T, \quad \mathbf{s}' = \begin{bmatrix} s'_1 & \dots s'_M \end{bmatrix}^T, \tag{6.37}$$

and ρ is the penalty parameter. The vector

$$\boldsymbol{\lambda} = [(\boldsymbol{\lambda}^{\mathbf{u}})^T \; (\boldsymbol{\lambda}^{\mathbf{s}})^T \; (\boldsymbol{\lambda}^{\mathbf{s}}_C)^T \; (\boldsymbol{\lambda}^{\mathbf{v}}_C)^T \; \{(\boldsymbol{\lambda}^{\mathbf{v}}_i)^T\}_{i=1}^M]^T \tag{6.38}$$

contains all the scaled dual variables associated with the constraints in (6.35). Intuitively, the penalty parameter ρ controls the strength of the penalization of the deviation of the auxilary variables from the original ones.

ADMM alternates between updating the auxilary variables $\{\mathbf{z}'_i\}$ and $\mathbf{z}_C$, followed by the original variables $\mathbf{z}$, w.r.t. the augmented Lagrangian. The Lagrange multipliers $\boldsymbol{\lambda}$ are also updated, following the dual variable update principle [11]. Sec. 6.4.3 will elaborate on the overall algorithm and the associated convergence guarantee. Next in Sec. 6.4.2 we will first examine in detail the individual update steps.

6.4.2 Update steps

The vectors $\{\mathbf{z}'_i\}$, $\mathbf{z}_C$, and $\mathbf{z}$ are respectively updated by minimizing the augmented Lagrangian with respect to the target vector, while keeping the other vectors fixed. Specifically, these updates are

$$\mathbf{z}'_i \leftarrow \underset{\mathbf{z}'_i}{\arg\min}\ \mathcal{L}_\rho(\mathbf{z}, \{\mathbf{z}'_i\}, \mathbf{z}_C, \boldsymbol{\lambda}),\ \forall i, \tag{6.39a}$$

$$\mathbf{z}_C \leftarrow \underset{\mathbf{z}_C}{\arg\min}\ \mathcal{L}_\rho(\mathbf{z}, \{\mathbf{z}'_i\}, \mathbf{z}_C, \boldsymbol{\lambda}), \tag{6.39b}$$

$$\mathbf{z} \leftarrow \underset{\mathbf{z}}{\arg\min}\ \mathcal{L}_\rho(\mathbf{z}, \{\mathbf{z}'_i\}, \mathbf{z}_C, \boldsymbol{\lambda}), \tag{6.39c}$$

where, to avoid clutter, we don't distinguish between the target vector and the other vectors on the RHS.

After the vectors $\{\mathbf{z}'_i\}$, $\mathbf{z}_C$, and $\mathbf{z}$ are revised, the ADMM procedure updates the Lagrange multipliers as follows

$$\begin{aligned}
\boldsymbol{\lambda}^{\mathbf{u}} &\leftarrow \boldsymbol{\lambda}^{\mathbf{u}} + \mathbf{u}' - \mathbf{u},\\
\boldsymbol{\lambda}^{\mathbf{s}} &\leftarrow \boldsymbol{\lambda}^{\mathbf{s}} + \mathbf{s}' - \mathbf{s},\\
\boldsymbol{\lambda}^{\mathbf{s}}_C &\leftarrow \boldsymbol{\lambda}^{\mathbf{s}}_C + \mathbf{s}_C - \mathbf{s},\\
\boldsymbol{\lambda}^{\mathbf{v}}_C &\leftarrow \boldsymbol{\lambda}^{\mathbf{v}}_C + \mathbf{v}_C - \mathbf{v},\\
\boldsymbol{\lambda}^{\mathbf{v}}_i &\leftarrow \boldsymbol{\lambda}^{\mathbf{v}}_i + \mathbf{v}'_i - \mathbf{v},\ \forall i.
\end{aligned} \tag{6.40}$$

Intuitively, from the way vector $\boldsymbol{\lambda}$ is being updated, the vector can be interpreted as the accumulated shift of the auxiliary variables from the original variables [11].

In the following, we take a deeper look into the subproblems in (6.39).

6.4.2.1 Updating $\mathbf{z}'_i$

Due to the decomposable nature of the augmented Langrangian (6.58), the problem in (6.39a) can be reduced to

$$\begin{aligned}
\underset{\mathbf{z}'_i}{\arg\min}\ & u'_i + \mathbb{I}_B(\mathbf{z}'_i) + \rho(u'_i - u_i + \lambda^{\mathbf{u}}_i)^2\\
& + \rho(s'_i - s_i + \lambda^{\mathbf{s}}_i)^2 + \rho\|\mathbf{v}'_i - \mathbf{v} + \boldsymbol{\lambda}^{\mathbf{v}}_i\|_2^2,
\end{aligned} \tag{6.41}$$

where terms not affected by $\mathbf{z}'_i$ have also been ignored. This can be reverted to the constrained problem

$$\begin{aligned}
\underset{\mathbf{z}'_i}{\arg\min} \quad & u'_i + \rho(u'_i - u_i + \lambda^{\mathbf{u}}_i)^2 \\
& + \rho(s'_i - s_i + \lambda^{\mathbf{s}}_i)^2 + \rho\|\mathbf{v}'_i - \mathbf{v} + \boldsymbol{\lambda}^{\mathbf{v}}_i\|_2^2 && (6.42a)\\
\text{s.t.} \quad & u'_i(s'_i - \mathbf{c}_i^T\mathbf{v}'_i + b_i) = 0, && (6.42b)\\
& s'_i(1 - u'_i) = 0, && (6.42c)\\
& u'_i \in \{0, 1\}. && (6.42d)
\end{aligned}$$

Due to the complementarity constraints (6.42b) and (6.42c), and the binary restriction (6.42d) on u'_i , (6.42) can be solved by simply enumerating u'_i:

- $u'_i = 0$: Then s'_i must also be 0 to satisfy all the constraints, and $\mathbf{v}'_i$ must be assigned the value of $\mathbf{v} - \boldsymbol{\lambda}^{\mathbf{v}}_i$ to minimize (6.42a).
- $u'_i = 1$: To satisfy (6.42b), s'_i must be equal to $\mathbf{c}_i^T\mathbf{v}'_i - b_i$ Then problem (6.42) becomes the unconstrained convex quadratic program (QP)

  $$\min_{\mathbf{v}'_i} \; (\mathbf{c}_i^T\mathbf{v}'_i - b_i - s_i + \lambda^{\mathbf{s}}_i)^2 + \|\mathbf{v}'_i - \mathbf{v} + \boldsymbol{\lambda}^{\mathbf{v}}_i\|_2^2. \tag{6.43}$$

  When $\mathbf{v}'_i$ is obtained, s'_i can be computed accordingly.

The revised $\mathbf{z}'_i$ is simply chosen as the combination of the variables that results in the smaller objective value in (6.42). Note that the value of ρ would affect the chosen $\mathbf{z}'_i$.

6.4.2.2 Updating $\mathbf{z}_C$

Ignoring terms unrelated to $\mathbf{z}_C$, the problem in (6.39b) can be reexpressed as

$$\underset{\mathbf{z}_C}{\arg\min} \; \mathbb{I}_C(\mathbf{z}_C) + \rho\|\mathbf{s}_C - \mathbf{s} + \boldsymbol{\lambda}^{\mathbf{s}}_C\|_2^2 + \rho\|\mathbf{v}_C - \mathbf{v} + \boldsymbol{\lambda}^{\mathbf{v}}_C\|_2^2. \tag{6.44}$$

The above is equivalent to the convex QP

$$\begin{aligned}
\min_{\mathbf{z}_C} \quad & \|\mathbf{s}_C - \mathbf{s} + \boldsymbol{\lambda}^{\mathbf{s}}_C\|_2^2 + \|\mathbf{v}_C - \mathbf{v} + \boldsymbol{\lambda}^{\mathbf{v}}_C\|_2^2, \\
\text{s.t.} \quad & \mathbf{s}_C - \mathbf{C}\mathbf{v}_C + \mathbf{b} \geq \mathbf{0}, \\
& \mathbf{s}_C, \mathbf{v}_C \geq \mathbf{0},
\end{aligned} \tag{6.45}$$

where the constraints in $\mathbb{I}_C$ are now listed explicitly.

6.4.2.3 Updating $\mathbf{z}$

Again ignoring terms unrelated to the variables of interest, the problem in (6.39c) reduces to

$$\begin{aligned}\arg\min_{\mathbf{z}} \quad & \rho(\|\mathbf{u}' - \mathbf{u} + \boldsymbol{\lambda}^{\mathbf{u}}\|_2^2 + \|\mathbf{s}' - \mathbf{s} + \boldsymbol{\lambda}^{\mathbf{s}}\|_2^2 \\ & + \|\mathbf{s}_C - \mathbf{s} + \boldsymbol{\lambda}^{\mathbf{s}}_C\|_2^2 + \|\mathbf{v}_C - \mathbf{v} + \boldsymbol{\lambda}^{\mathbf{v}}_C\|_2^2 \\ & + \sum_i \|\mathbf{v}'_i - \mathbf{v} + \boldsymbol{\lambda}^{\mathbf{v}}_i\|_2^2) + \|\mathbf{u}\|^2.\end{aligned} \tag{6.46}$$

The three components $\mathbf{u}$, $\mathbf{s}$ and $\mathbf{v}$ of $\mathbf{z}$ decouple, and in fact can be solved for easily as the "mean vectors"

$$\begin{aligned}\mathbf{u} &= \frac{\rho}{\rho+1}(\mathbf{u}' + \boldsymbol{\lambda}^{\mathbf{u}}), \\ \mathbf{s} &= \frac{1}{2}\left(\mathbf{s}' + \boldsymbol{\lambda}^{\mathbf{s}} + \mathbf{s}_C + \boldsymbol{\lambda}^{\mathbf{s}}_C\right), \\ \mathbf{v} &= \frac{1}{M+1}\left[\sum_{i=1}^{M}(\mathbf{v}'_i + \boldsymbol{\lambda}^{\mathbf{v}}_i) + \mathbf{v}_C + \boldsymbol{\lambda}^{\mathbf{v}}_C\right].\end{aligned}$$

Finally, we emphasize that all the update steps above can be solved efficiently, requiring no more than a convex QP.

6.4.3 Main algorithm

Similar to the non-smooth penalty algorithm discussed in Sec. 6.3.2, directly setting ρ to a very large value will likely lead to a bad suboptimal result. Therefore, also applied here is a heuristic strategy that initializes ρ to a small value then gradually increases ρ after each ADMM update cycle. The algorithm is terminated when the variable $\mathbf{z}$ converges. Algorithm 6.3 summarizes the overall procedure.

6.4.3.1 Convergence

Theorem 6.5. *For a sufficiently large ρ, the ADMM update iterations in* (6.39) *converge to a stationary point of the augmented Lagrangian* (6.58)*, which is also a feasible solution of* (6.10)*, after a finite number of steps.*

Proof. The detailed proof for this theorem can be found in the supplementary material. For completeness, an outline of the proof is provided in this section.

Consider the $(t+1)$-th update cycle of Algorithm. 6.3. To prevent clutter, let $\{\mathbf{z}_i\}^+, \mathbf{z}_C^+, \mathbf{z}^+$ and $\boldsymbol{\lambda}^+$ denote the updated value of the variables while $\{\mathbf{z}_i\}, \mathbf{z}_C, \mathbf{z}$ and $\boldsymbol{\lambda}$ represent the variables carried from the (t)-th iteration.

Algorithm 6.3 ADMM-based method for solving (6.10).

```
Require: Data $\{\mathbf{c}_i, b_i\}_{i=1}^M$, initial model parameter $\boldsymbol{\theta}$, initial penalty value $\rho$, increment rate $\sigma$,
    threshold $\delta$.
t ← 0
$\mathbf{v}^{(t)} \leftarrow \left[(\boldsymbol{\theta} + |\min_j(\theta_j)|\mathbf{1})^T \ \ |\min_j(\theta_j)|\right]^T$.
$\mathbf{u}^{(t)} \leftarrow \mathbb{I}(\mathbf{C}\mathbf{v} - \mathbf{b} > 0)$.
$\mathbf{s}^{(t)} \leftarrow \mathbf{u} \odot (\mathbf{C}\mathbf{v} - \mathbf{b})$.
${\mathbf{z}_i}^{(t)} = \mathbf{z}^{(t)}$ ; ${\mathbf{z}_C}^{(t)} = [\mathbf{s}^{(t)}; \mathbf{v}^{(t)}]$ ; $\boldsymbol{\lambda}^{(t)} = \mathbf{0}$
while true do
    $t \leftarrow t + 1$
    Update ${\mathbf{z}_i}^{(t)}$ by solving (6.42) $\forall\ i = 1..N$
    Update ${\mathbf{z}_C}^{(t)}$ by solving (6.45)
    Update $\mathbf{z}^{(t)}$ by solving (6.46)
    if $\|\mathbf{z}^{(t)} - \mathbf{z}^{(t-1)}\| \le \delta$ then
        Break.
    end if
    $\rho^{(t)} \leftarrow \sigma \cdot \rho^{(t-1)}$.
end while
return $\mathbf{u}, \mathbf{s}, \mathbf{v}$.
```

During the update steps of $\{\mathbf{z}_i\}$ and $\mathbf{z}_C$, since (6.42) and (6.45) can be solved optimally, it follows that:

$$\mathcal{L}_\rho(\mathbf{z}, \{\mathbf{z}'_i\}, \mathbf{z}_C, \boldsymbol{\lambda}) \ge \mathcal{L}_\rho(\mathbf{z}, \{\mathbf{z}'_i\}^+, \mathbf{z}_C^+, \boldsymbol{\lambda}) \tag{6.47}$$

Then, after $\mathbf{z}$ and $\boldsymbol{\lambda}$ are updated, with a sufficiently large ρ, it can be proven that:

$$\mathcal{L}_\rho(\mathbf{z}, \{\mathbf{z}'_i\}^+, \mathbf{z}_C^+, \boldsymbol{\lambda}) \ge \mathcal{L}_\rho(\mathbf{z}^+, \{\mathbf{z}'_i\}^+, \mathbf{z}_C^+, \boldsymbol{\lambda}^+) \tag{6.48}$$

(detailed proof is provided in the supplementary material). From (6.47) and (6.48), the following inequality holds:

$$\mathcal{L}_\rho(\mathbf{z}, \{\mathbf{z}'_i\}, \mathbf{z}_C, \boldsymbol{\lambda}) \ge \mathcal{L}_\rho(\mathbf{z}^+, \{\mathbf{z}'_i\}^+, \mathbf{z}_C^+, \boldsymbol{\lambda}^+) \tag{6.49}$$

given that ρ is large enough.

The inequality (6.49) states that, with a sufficiently large ρ, the augmented Lagrangian (6.58) is monotonically nonincreasing after every ADMM update cycle. As this function is bounded below with a sufficiently large ρ (detailed proof is given in the supplementary material), its convergence to a point $\mathbf{z}^*$ is guaranteed (see the supplementary material for details). At convergence, all the constraints (6.32b), (6.32c) and (6.32d) are satisfied and $\mathbf{z}^*$ is also a feasible solution of (6.10). □

□

6.4.3.2 Initialization

Similar to Alg. 6.2, $\mathbf{u}, \mathbf{s}, \mathbf{v}$ can be initialized from a suboptimal solution such as RANSAC or least squares fit. To avoid bad local minmima, the starting values of ρ are chosen to be relatively small ($0 \leq \rho \leq 10$) with a conservative increase rate σ ($1.01 \leq \sigma \leq 5$). It will be demonstrated in Section 6.6 that with this choice of the parameters, the algorithm was able to significantly improve the solution from an intial starting point.

6.5 Handling geometric distances

For most applications in computer vision, the residual function used for geometric model fitting is non-linear. It has been shown [49, 73, 6], however, that many geometric residuals have the following *generalized fractional* form

$$\frac{\|\mathbf{G}\boldsymbol{\theta} + \mathbf{h}\|_p}{\mathbf{r}^T\boldsymbol{\theta} + q} \quad \text{with} \quad \mathbf{r}^T\boldsymbol{\theta} + q > 0, \tag{6.50}$$

where $\|\cdot\|_p$ is the p-norm, and $\mathbf{G} \in \mathbb{R}^{2\times d}$, $\mathbf{h} \in \mathbb{R}^2$, $\mathbf{r} \in \mathbb{R}^d$, $q \in \mathbb{R}^1$ are constants derived from the input data. For example, the reprojection error in triangulation and transfer error in homography fitting can be coded in the form (6.50). The associated maximum consensus problem is

$$\max_{\boldsymbol{\theta}\in\mathbb{R}^d} \quad \Psi(\boldsymbol{\theta}), \tag{6.51}$$

where

$$\Psi(\boldsymbol{\theta}) = \sum_{j=1}^{N} \mathbb{I}\left(\|\mathbf{G}_j\boldsymbol{\theta} + h_j\|_p \leq \epsilon(\mathbf{r}_j^T\boldsymbol{\theta} + q_j)\right). \tag{6.52}$$

In (6.52), we have moved the denominator of (6.50) to the RHS since ϵ is non-negative (see [49] for details). We show that for $p = 1$, our method can be easily adapted to solve maximum consensus for geometric residuals (6.51)[2]. Define

$$\mathbf{G}_j = \begin{bmatrix} \mathbf{g}_{j,1}^T \\ \mathbf{g}_{j,2}^T \end{bmatrix} \quad \mathbf{h}_j = \begin{bmatrix} h_{j,1} \\ h_{j,2} \end{bmatrix}. \tag{6.53}$$

Now, for $p = 1$, the constraint in (6.51) becomes

$$\left|\mathbf{g}_{j,1}^T\boldsymbol{\theta} + h_{j,1}\right| + \left|\mathbf{g}_{j,2}^T\boldsymbol{\theta} + h_{j,2}\right| \leq \epsilon(\mathbf{r}_j^T\boldsymbol{\theta} + q_j), \tag{6.54}$$

[2] Note that, in the presence of outliers, the residuals are no longer i.i.d. Normal. Thus, the 1-norm is arguably as valid as the 2-norm for maximum consensus robust fitting.

which in turn can be equivalently implemented using four linear constraints (see [6] for details). We can then manipulate (6.52) into the form (6.5), and the rest of our theory and algorithms will be immediately applicable.

6.6 Results

We tested our method (Algorithm 6.2 and Algorithm 6.3, henceforth abbreviated as EP and ADMM, respectively) on common parameter estimation problems. We compared EP and ADMM against the following well-known methods:

- RANSAC (RS) [31]: We used confidence $\rho = 0.99$ for the stopping criterion in all the experiments. On each data instance, RANSAC was executed 10 times and the average consensus size and runtime were reported.
- LO-RANSAC (LORS) [23]: The maximum number of iterations for the inner sampling over the best consensus set was set to 100. The size of the inner sampled subsets was set to be twice the size of the minimal subset.
- Improved LO-RANSAC (LORS1) [56]: Following [56], the inner RANSAC routine will only be run if the new consensus size is higher than a pre-defined threshold (set to 10% of the data size in our experiments).
- ℓ_1 approximation (ℓ_1) [74]: This method is equivalent to introducing slack variables to problem (6.2) and minimizing the ℓ_1-norm of the slack variables to yield an approximate solution to maximum consensus.
- ℓ_∞ outlier removal (l_∞) [79]: Again, in the context of (6.2), slack variables are introduced and the maximum slack value is minimized. Data with the largest slack value are removed, and the process of repeated until the largest slack value is not greater than zero.
- For the experiments with image data where keypoint matching scores are available as inlier priors, we executed two state-of-the-art RANSAC variants: PROSAC (PS) [22] and Guided MLESAC (GMLE) [84].

All the methods and experiments were implemented in MATLAB and run on a standard desktop machine with 3.5 GHz processor and 8 GB of RAM. For EP, ADMM, ℓ_1 and ℓ_∞, Gurobi was employed as the LP and QP solver.

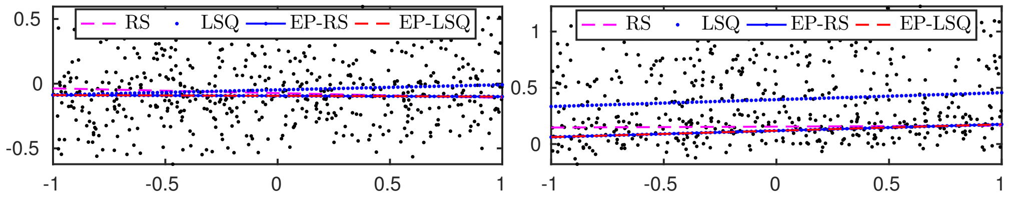
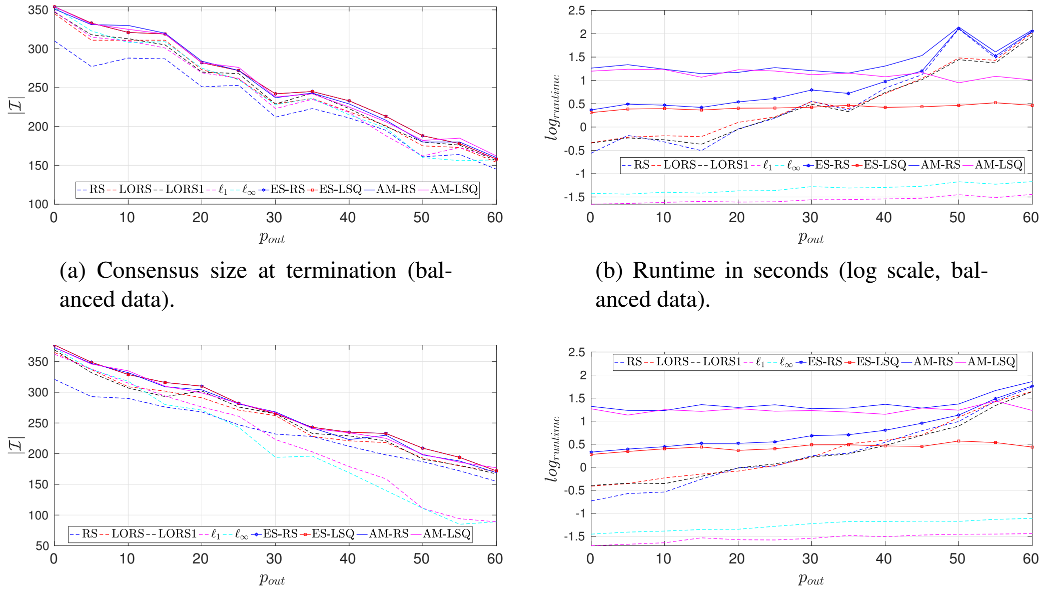


FIGURE 6.2: Two-dimensional analogy of balanced (top) and unbalanced (bottom) data generated in our experiments. The results of RANSAC, least squares, and our method initialized with the former two methods are shown. Observe that least squares is heavily biased under unbalanced data, but EP is able to recover from the bad initialization. (For clarity, the results of AM variants are not plotted as they are very close to EP-RS and EP-LSQ)



(a) Consensus size at termination (balanced data).

(b) Runtime in seconds (log scale, balanced data).

(c) Consensus size at termination (unbalanced data).

(d) Runtime in seconds (log scale, unbalanced data).

FIGURE 6.3: Results for linear regression ($d = 8$ dimensions). (a)(b) Balanced data; (c)(d) Unbalanced data.

6.6.1 Linear models

6.6.1.1 Linear regression with synthetic data

We generated $N = 500$ points $\{\mathbf{x}_j, y_j\}_{j=1}^{N}$ in $\mathbb{R}^9$ following a linear trend $y = \mathbf{x}^T\boldsymbol{\theta}$, where $\boldsymbol{\theta} \in \mathbb{R}^8$ and $\mathbf{x}_j \in [-1, 1]^8$ were randomly sampled. Each y_j was perturbed by Gaussian noise with standard deviation of $\sigma_{in} = 0.1$. To simulate outliers, $p_{out}\%$ of y_j's were randomly selected and corrupted. To test the ability of our methods to deal with bad initializations, two different outlier settings were considered:

- Balanced data: the y_j of outliers were added with Gaussian noise of $\sigma_{out} = 1$. This evenly distributed the outliers on both sides of the hyperplane.

- Unbalanced data: as above, but the sign of the additive noise was forced to be positive. Thus, outliers were distributed only on one side of the hyperplane. On such data, the least squares solution is heavily biased.

See Fig. 6.2 for a *2D analogy* of these outlier settings. We tested with $p_{out} = \{0, 5, 10 \ldots, 60\}$. The inlier threshold for maximum consensus was set to $\epsilon = 0.1$.

Our algorithms EP and ADMM were initialized respectively with RANSAC (variants EP-RS and ADMM-RS) and least squares (variants EP-LSQ and ADMM-LSQ). For EP variants, the initial α was set to 0.5 and $\kappa = 5$, while initial ρ of ADMM variants was set to 0.1 and $\sigma = 2.5$ for all the runs.

Methods		RS	PS	GMLE	LORS	LORS1	ℓ_1	ℓ_∞	EP-RS	EP-LSQ	EP-ℓ_∞	AM-RS	AM-LSQ	AM-ℓ_∞
House	$\lvert\mathcal{I}\rvert$	387	367	392	407	410	31	394	**418**	**418**	**418**	417	391	**418**
N = 556	time (s)	0.05	0.08	0.04	0.2	0.25	0.02	0.03	2.05	1.99	2.14	8.35	10.13	6.62
Aerial	$\lvert\mathcal{I}\rvert$	360	320	353	376	377	31	357	371	371	**381**	368	369	380
N = 483	time (s)	0.06	0.1	0.05	0.22	0.27	0.01	0.04	1.12	1.18	1.13	4.81	10.49	7.58
Merton	$\lvert\mathcal{I}\rvert$	499	484	500	502	502	95	501	508	508	508	508	504	**511**
N = 590	time (s)	0.06	0.06	0.06	0.25	0.39	0.03	0.03	1.17	1.08	1.1	7.44	7.02	8.82
Wadham	$\lvert\mathcal{I}\rvert$	462	409	481	527	527	3	502	**531**	**531**	**531**	504	522	502
N = 618	time (s)	0.05	0.08	0.04	0.29	0.36	0.02	0.03	1.69	1.61	1.62	6.88	16.46	6.52
Corridor	$\lvert\mathcal{I}\rvert$	385	376	385	394	396	31	376	389	**402**	**402**	392	392	397
N = 684	time (s)	0.24	0.29	0.2	0.36	0.62	0.05	0.07	4.39	3.36	3.59	10.52	10.56	10.0
Buidling 81	$\lvert\mathcal{I}\rvert$	414	384	420	440	439	441	442	**446**	**446**	**446**	431	445	442
N = 525	time (s)	0.05	0.05	0.05	0.23	0.3	0.01	0.02	1.65	1.58	1.5	15.32	8.69	2.45
Building 04	$\lvert\mathcal{I}\rvert$	182	182	172	194	192	32	171	197	197	197	**200**	122	184
N = 394	time (s)	1.26	1.36	1.58	1.03	2.2	0.04	0.05	2.18	1.22	1.2	6.33	11.36	6.61
Building 23	$\lvert\mathcal{I}\rvert$	314	315	302	326	326	20	217	330	330	330	**332**	179	323
N = 699	time (s)	1.45	1.44	1.96	1.24	2.61	0.02	0.11	4.17	3.06	2.89	9.97	13.85	10.02
Building 36	$\lvert\mathcal{I}\rvert$	397	366	381	411	410	20	353	**418**	**418**	**418**	409	344	391
N = 651	time (s)	0.13	0.31	0.19	0.32	0.42	0.03	0.05	2.61	2.42	2.6	9.39	11.46	9.71
Building 50	$\lvert\mathcal{I}\rvert$	320	307	319	322	322	42	317	**325**	**325**	**325**	322	324	323
N = 365	time (s)	0.05	0.05	0.04	0.16	0.24	0.02	0.01	0.6	0.55	0.53	3.78	4.54	3.28

TABLE 6.1: Fundamental matrix estimation results

Methods		RS	PS	GMLE	LORS	LORS1	ℓ_1	ℓ_∞	EP-RS	EP-LSQ	EP-ℓ_∞	AM-RS	AM-LSQ	AM-ℓ_∞
Notre Dame 1	$\lvert\mathcal{I}\rvert$	270	264	269	272	272	224	217	278	278	278	280	**281**	**281**
N = 310	time (s)	0.07	0.07	0.05	0.5	0.56	0.02	0.03	0.79	0.9	0.86	9.19	14.25	7.81
Notre Dame 2	$\lvert\mathcal{I}\rvert$	127	129	128	134	134	72	82	135	135	135	**136**	135	**136**
N = 254	time (s)	0.2	0.18	0.16	0.55	0.79	0.04	0.04	2.18	2.06	1.86	15.68	9.22	12.02
Notre Dame 3	$\lvert\mathcal{I}\rvert$	445	446	444	445	445	394	412	447	447	447	**448**	**448**	447
N = 521	time (s)	0.17	0.15	0.12	6.71	6.93	0.1	0.13	4.21	4.22	4.27	39.15	42.4	27.27
Notre Dame 4	$\lvert\mathcal{I}\rvert$	406	375	395	482	480	437	483	487	487	487	**488**	484	487
N = 543	time (s)	0.06	0.06	0.05	1.09	1.15	0.04	0.03	2.4	2.43	2.23	12.41	20.23	17.85
South Building 1	$\lvert\mathcal{I}\rvert$	147	142	142	153	154	117	116	149	149	149	**166**	164	164
N = 552	time (s)	0.06	0.06	0.05	0.79	0.85	0.07	0.07	11.09	11.14	8.27	26.36	38.35	35.51
South Building 2	$\lvert\mathcal{I}\rvert$	432	427	434	442	438	128	440	**462**	**462**	**462**	445	447	449
N = 507	time (s)	0.06	0.06	0.05	1.57	1.62	0.05	0.03	4.77	4.72	4.69	6.6	6.41	6.27
South Buildind 3	$\lvert\mathcal{I}\rvert$	318	309	298	372	376	10	338	**380**	**380**	**380**	371	375	371
N = 394	time (s)	0.07	0.07	0.05	0.56	0.8	0.08	0.05	9.35	9.27	8.88	26.94	30.54	24.58
Gerrard 1	$\lvert\mathcal{I}\rvert$	97	96	98	99	100	6	55	98	98	98	102	103	**105**
N = 181	time (s)	0.17	0.18	0.19	0.24	0.42	0.02	0.03	0.96	0.81	0.83	5.21	4.58	5.09
Gerrard 2	$\lvert\mathcal{I}\rvert$	157	152	151	162	162	9	103	**171**	**171**	**171**	165	166	168
N = 399	time (s)	0.62	0.7	0.66	0.72	1.33	0.07	0.05	3.25	2.81	2.74	12.26	14.47	14.42
Gerrard 3	$\lvert\mathcal{I}\rvert$	83	85	84	**96**	94	47	73	**96**	**96**	**96**	**96**	86	82
N = 208	time (s)	0.24	0.25	0.19	0.3	0.53	0.02	0.02	1.27	1.01	1.03	8.15	6.87	7.63

TABLE 6.2: Linearised homography estimation results

Fig. 6.3 shows the average consensus size at termination and runtime (in log scale) of the methods. Note that runtime of RS and LSQ were included in the runtime of EP-RS, ADMM-RS, EP-LSQ and ADMM-LSQ, respectively. It is clear that, in terms of solution quality, the variants

Methods		RS	PS	GMLE	LORS	LORS1	ℓ_1	ℓ_∞	EP-RS	EP-ℓ_∞	AM-RS	AM-ℓ_∞
University Library	$\lvert\mathcal{I}\rvert$	236	234	231	261	257	107	36	**268**	**268**	264	266
N = 439	time (s)	0.53	0.46	0.46	1.62	1.43	1.06	1.65	7.53	12.32	11.95	12.91
Christ Church	$\lvert\mathcal{I}\rvert$	241	234	227	253	251	245	142	**283**	**283**	278	278
N = 439	time (s)	0.49	0.52	0.5	1.44	1.63	1.35	2.09	8.95	10.93	16.82	18.16
Kapel	$\lvert\mathcal{I}\rvert$	165	168	126	168	168	28	104	**169**	**169**	166	168
N = 449	time (s)	1.18	1.11	7.87	2.31	2.68	2.7	2.07	12.44	13.32	13.17	11.61
Invalides	$\lvert\mathcal{I}\rvert$	161	161	148	174	174	13	126	**187**	**187**	177	176
N = 558	time (s)	4.29	3.92	5.93	4.31	8.01	2.9	1.42	33.92	31.51	15.33	13.44
Union House	$\lvert\mathcal{I}\rvert$	213	213	199	224	230	14	65	231	231	**232**	208
N = 520	time (s)	1.56	1.64	2.5	3.27	3.51	3.72	1.78	21.84	26.59	17.73	17.35
Old Classic Wing	$\lvert\mathcal{I}\rvert$	198	208	126	209	210	52	147	**216**	206	210	197
N = 561	time (s)	1.85	1.47	20.57	3.32	3.96	2.77	1.47	19.29	31.57	17.06	17.23
Ball Hall	$\lvert\mathcal{I}\rvert$	235	218	220	253	253	33	70	**272**	205	234	115
N = 534	time (s)	1.03	1.18	1.04	1.7	2.34	0.57	1.05	7.47	10.47	12.45	13.35
Building 64	$\lvert\mathcal{I}\rvert$	123	128	100	135	133	73	82	**142**	**142**	**142**	**142**
N = 427	time (s)	3.27	2.56	10.11	3.63	5.93	1.17	0.99	22.95	21.54	15.07	14.05
Building 10	$\lvert\mathcal{I}\rvert$	204	223	170	223	226	176	165	**229**	**229**	226	210
N = 425	time (s)	0.48	0.48	0.95	1.46	1.38	1.14	1.71	10.66	12.59	13.56	13.48
Building 15	$\lvert\mathcal{I}\rvert$	335	338	293	339	339	333	262	**345**	**345**	337	336
N = 477	time (s)	0.53	0.52	0.49	1.39	1.65	1.64	1.17	14.39	14.56	18.31	16.08

TABLE 6.3: Homography estimation results

Methods		RS	PS	GMLE	LORS	LORS1	ℓ_1	ℓ_∞	EP-RS	EP-ℓ_∞	AM-RS	AM-ℓ_∞
Bikes	$\lvert\mathcal{I}\rvert$	410	410	410	411	410	412	415	**421**	**421**	417	417
N = 518	time (s)	5.94	5.86	5.6	8.23	13.42	4.52	0.97	15.21	7.76	10.42	5.65
Tree	$\lvert\mathcal{I}\rvert$	286	288	289	287	286	301	278	**311**	**311**	305	307
N = 465	time (s)	5.94	5.86	5.6	8.23	13.42	4.52	0.97	15.21	7.76	10.42	5.65
Boat	$\lvert\mathcal{I}\rvert$	308	311	304	310	308	330	330	**340**	**340**	325	330
N = 402	time (s)	5.61	5.63	5.31	6.62	10.91	2.46	0.88	10.34	5.59	10.12	5.05
Graff	$\lvert\mathcal{I}\rvert$	140	141	142	141	140	304	308	**313**	**313**	308	308
N = 331	time (s)	4.95	4.7	4.32	5.91	9.34	1.39	0.39	10.82	6.26	17.18	11.7
Bark	$\lvert\mathcal{I}\rvert$	194	195	195	194	194	200	**203**	**203**	**203**	202	**203**
N = 219	time (s)	3.01	3.06	3.41	3.42	5.61	0.32	0.32	3.86	1.17	14.21	14.49
Building 143	$\lvert\mathcal{I}\rvert$	94	93	91	99	94	338	331	342	342	**349**	347
N = 537	time (s)	7.97	8.19	8.02	9.52	15.41	5.62	2.55	16.6	10.28	34.77	33.12
Building 152	$\lvert\mathcal{I}\rvert$	198	192	173	211	198	221	228	**281**	**281**	277	277
N = 469	time (s)	6	6	5.71	7.71	11.67	3.16	1.71	12.41	7.75	28.2	24.03
Building 163	$\lvert\mathcal{I}\rvert$	306	308	303	307	306	402	399	**437**	**437**	431	430
N = 617	time (s)	7.85	7.82	7.58	8.93	15.3	8.06	3.37	16.93	11.64	21.93	17.04
Building 170	$\lvert\mathcal{I}\rvert$	315	311	311	318	315	455	412	**538**	**538**	524	525
N = 707	time (s)	9.48	9.46	9.25	11.65	18.72	11.24	2.18	31.66	23.73	61.65	57.71
Building 174	$\lvert\mathcal{I}\rvert$	339	338	339	341	339	334	312	369	369	**375**	374
N = 580	time (s)	7.8	7.73	7.4	9.78	15.13	5.94	1.89	17.92	11.77	50.48	38.63

TABLE 6.4: Affinity estimation results

of EP and ADMM consistently outperformed the other methods. The fact that EP-LSQ could match the quality of EP-RS on unbalanced data attest to the ability of EP to recover from bad initializations. In terms of runtime, while both EP variants were slightly more expensive than the RANSAC variants, as p_{out} increased over 35%, EP-LSQ began to outperform the RANSAC variants (since EP-RS was initialized using RANSAC, its runtime also increased with p_{out}). ADMM variants were also able to obtain roughly the same quality as EP-based methods, albeit with longer runtime. This is explainable as ADMM requires solving quaratic subproblems while only LPs are required for EP variants.

Methods		RS	LORS	LORS1	ℓ_1	ℓ_∞	EP-RS	EP-ℓ_∞	AM-RS	AM-ℓ_∞
Point 1	$\|\mathcal{I}\|$	95	96	95	96	81	**97**	**97**	**97**	96
N = 167	time (s)	0.22	0.47	0.38	0.8	2.04	3.69	4.34	1.81	3.63
Point 3	$\|\mathcal{I}\|$	82	84	82	79	53	85	84	**86**	77
N = 145	time (s)	0.15	0.32	0.29	0.16	1.16	1.92	2.97	2.04	2.5
Point 9	$\|\mathcal{I}\|$	49	51	49	30	38	**52**	49	**52**	47
N = 135	time (s)	0.16	0.39	0.28	0.14	0.84	2.37	3.08	1.42	4.37
Point 15	$\|\mathcal{I}\|$	50	53	50	43	38	53	46	**55**	41
N = 140	time (s)	0.15	0.36	0.27	0.24	1.14	2.63	3.52	1.4	4.16
Point 24	$\|\mathcal{I}\|$	110	113	110	113	111	113	113	**114**	**114**
N = 155	time (s)	0.17	0.34	0.31	0.13	0.44	2.24	2.59	1.67	1.93
Point 72	$\|\mathcal{I}\|$	38	39	38	37	35	**41**	**41**	**41**	39
N = 104	time (s)	0.12	0.29	0.21	0.08	0.54	1.15	1.53	1.12	1.57
Point 82	$\|\mathcal{I}\|$	56	58	56	55	48	59	59	**60**	55
N = 118	time (s)	0.13	0.33	0.23	0.09	0.4	1.43	1.82	1.22	1.48
Point 192	$\|\mathcal{I}\|$	89	90	89	92	87	91	91	**93**	92
N = 123	time (s)	0.14	0.27	0.26	0.09	0.39	1.15	1.41	1.27	1.51
Point 193	$\|\mathcal{I}\|$	113	114	113	111	113	**117**	**117**	116	**117**
N = 132	time (s)	0.14	0.28	0.26	0.09	0.45	0.99	1.28	1.29	1.67
Point 249	$\|\mathcal{I}\|$	93	**94**	93	93	90	**94**	92	**94**	92
N = 124	time (s)	0.13	0.27	0.24	0.1	0.36	1.59	1.84	1.31	1.61

TABLE 6.5: Triangulation Results

6.6.1.2 Fundamental matrix estimation (with algebraic error)

Following [42, Chapter 11], the epipolar constraint is linearized to enable the fundamental matrix to be estimated linearly (note that the usual geometric distances for fundamental matrix estimation do not have the generalized fractional form (6.50), thus linearization is essential to enable our method. Sec. 6.6.2 will describe results for model estimation with geometric distances).

Five image pairs from the VGG dataset[3] (Corridor, House, Merton II, Wadham and Aerial View I) and five image paris from the Zurich Building data set[4] (Building 04, Building 23, Building 36, Building 50 and Building 81) were used. The images were first resized before SIFT (as implemented on VLFeat [87]) was used to extract around 500 correspondences per pair. An inlier threshold of $\epsilon = 1$ was used for all image pairs. For EP and ADMM, apart from initialization with RANSAC and least squares, we also initialised it with ℓ_∞ outlier removal (variants EP-ℓ_∞ and ADMM-ℓ_∞). For all EP variants, the initial α was set to 0.5 and $\kappa = 5$, while initial ρ for all ADMM variants was set to 0.1 and $\sigma = 2.5$ for all the runs.

Table 6.1 summarizes the quantitative results for all methods. Regardless of the initialization method, EP was able to find the largest consensus set. ADMM variants converge to approximately the same solution quality as EP while taking slightly longer runtime. Fig. 6.4 displays

[3] http://www.robots.ox.ac.uk/ vgg/data/

[4] http://www.vision.ee.ethz.ch/showroom/zubud/

sample qualitative results for EP; for the qualitative results for the other image pairs, please see the supplementary material.

6.6.1.3 Homography estimation (with algebraic error)

Following [42, Chapter 4], the homography constraints were linearized to investigate the performance of our algorithms. Five image pairs form the VGG dataset: University Library, Christ Church, Valbonne, Kapel and Paris's Invalides; three image pairs from the AdelaideRMF dataset [90]: Union House, Old Classic Wing, Ball Hall and three pairs from the Zurich Building dataset: Building 64, Building 10 and Building 15 were used for this experiment. Parameters for the EP and ADMM variants were reused from the fundamental matrix experiment. Quantitative results displayed in Table 6.2 show that all the EP and ADMM variants were able to achieve the highest consensus size.

6.6.2 Models with geometric distances

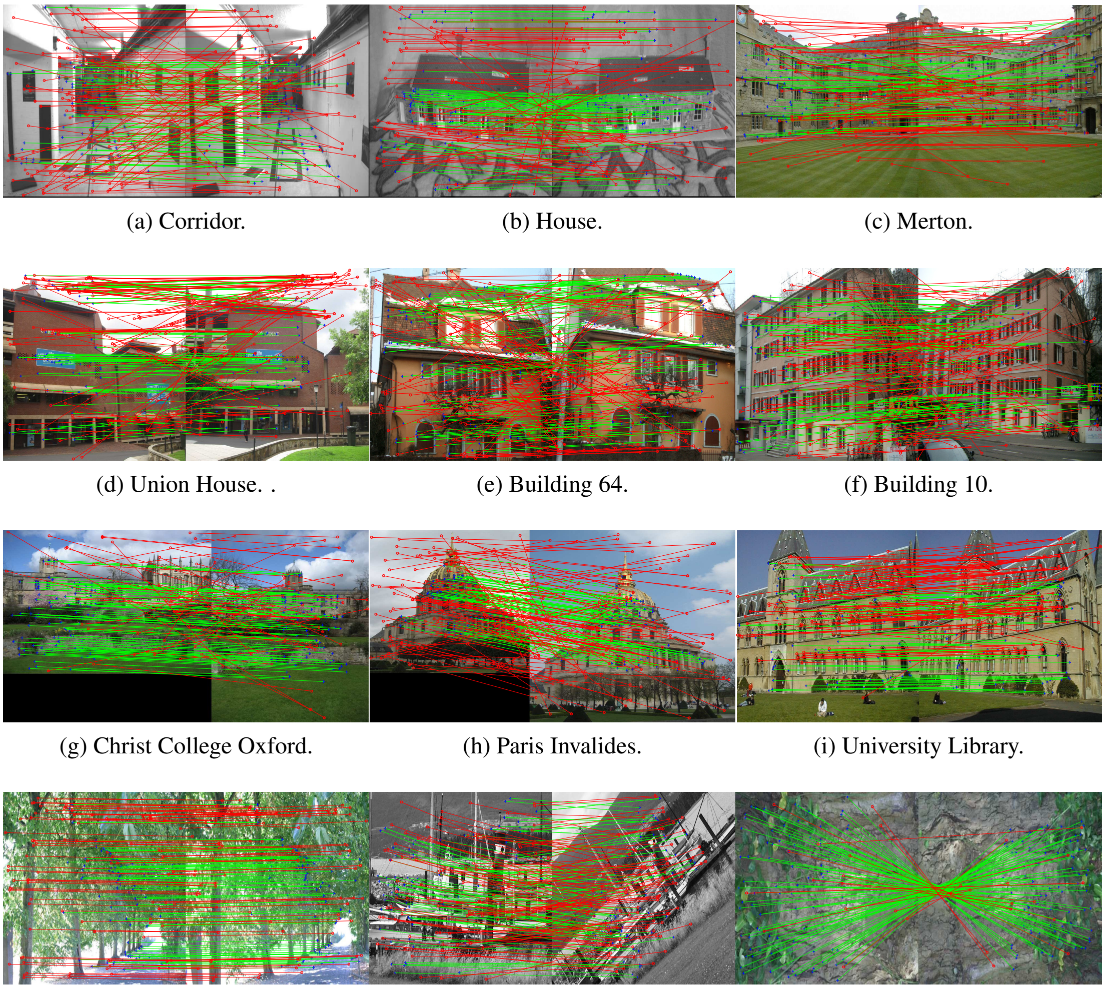
6.6.2.1 Homography estimation

We estimated 2D homographies based on the transfer error using all the methods. In the context of (6.50), the geometric residual for homography (with $p = 1$) is

$$\frac{\|(\boldsymbol{\theta}_{1:2} - \mathbf{v}_i\boldsymbol{\theta}_3)\tilde{\mathbf{u}}_i\|_1}{\boldsymbol{\theta}_3\tilde{\mathbf{u}}_i}, \tag{6.55}$$

where $\boldsymbol{\theta}_{1:2}$ and $\boldsymbol{\theta}_3$ denote the first-two rows and the last row of the homography matrix, respectively. Each pair $(\mathbf{u}_i, \mathbf{v}_i)$ represents a point match across two views, and $\tilde{\mathbf{u}}_i = [\mathbf{u}^T 1]^T$. The data used in the linearized homography experiment was reused. The inlier threshold of $\epsilon = 4$ pixels was used for all input data. Initial α was set to 10 and $\kappa = 1.5$ for all EP variants. For ADMM variants, initial ρ was set to 0.1 and the increment rate σ was set to 1.5 for all the runs.

Quantitative results are shown in Table 6.3, and a sample qualitative result for EP is shown in Fig. 6.4. Similar to the fundamental matrix case, the EP variants outperformed the other methods in terms of solution quality, but were slower though its runtime was still within the same order of magnitude. ADMM variants also attain approximately the same solution as EP with slightly longer runtimes. Note that EP-LSQ and ADMM-LSQ were not invoked here, since finding least squares estimates based on geometric distances is intractable in general [40].



(a) Corridor. (b) House. (c) Merton.

(d) Union House. . (e) Building 64. (f) Building 10.

(g) Christ College Oxford. (h) Paris Invalides. (i) University Library.

(j) Trees. (k) Boat. (l) Bark.

FIGURE 6.4: Qualitative results of local refinement methods on (a,b,c) fundamental matrix estimation, (d,e,f) linearized homography estimation (g,h,i) homography estimation with geometric distance, and (j,k,l) affinity estimation. Green and red lines represent detected inliers and outliers. For clarity, only 100 inliers/outliers are plotted. See the supplementary material for more qualititave results.

6.6.2.2 Affinity estimation

The previous experiment was repeated for affinity (6 DoF affine transformation) estimation, where the geometric matching error for the i-th correspondence can be written as:

$$\|\mathbf{u}_i - \boldsymbol{\theta}\tilde{\mathbf{v}}_i\|_1, \tag{6.56}$$

where each pair $(\mathbf{u}_i, \mathbf{v}_i)$ is a correspondence across two views, $\boldsymbol{\theta} \in \mathbb{R}^{2\times3}$ represents the affine transformation, and $\tilde{\mathbf{v}}_i = [\mathbf{v}^T \; 1]^T$. Initial α was set to 0.5, $\kappa = 5$ for EP variants and initial $\rho = 0.5$ and $\sigma = 2.5$ for ADMM variants. The inlier threshold was set to $\epsilon = 2$ pixels. Five image pairs from VGG's affine image dataset: Bikes, Graff, Bark, Tree, Boat and five pairs of building from the Zurich Building Dataset: Building 143, Building 152, Building 163, Building 170 and Building 174 were selected for the experiment. Quantitative results are given in Table 6.4, and sample qualitative result is shown in Fig. 6.4. Similar conclusions can be drawn.

6.6.2.3 Triangulation

We conducted triangulation from outlier-contaminated multiple-view observations of 3D points. For each image point $\mathbf{x}_i$ and the camera matrix $\mathbf{P}_i \in \mathbb{R}^{3\times 4}$, the following reprojection error with respect to the point estimation $\boldsymbol{\theta}$ was used in our experiments:

$$\frac{\|(\mathbf{P}^i_{1:2} - \mathbf{x}_i\mathbf{P}^i_3)\boldsymbol{\theta}'\|_1}{\mathbf{P}^i_3\tilde{\boldsymbol{\theta}}}, \tag{6.57}$$

where $\tilde{\boldsymbol{\theta}} = [\boldsymbol{\theta}^T\ 1]^T$, $\mathbf{P}^i_{1:2}$ denotes the first two rows of the camera matrix and $\mathbf{P}^i_3$ represents its third row. We selected five feature tracks from the NotreDame dataset [80] with more than $N = 150$ views each to test our algorithm. The inlier threshold for maximum consensus was set to $\epsilon = 1$ pixel. α was initially set to 0.5 and $\kappa = 1.5$ for all variants of EP. For the ADMM variants, initial ρ was set to 0.1 and $\sigma = 2.5$. Table 6.5 shows the quantitative results. Again, the variants of local refinement algorithms are better than the other methods in terms of solution quality. The runtime gap was not as significant here due to the low-dimensionality of the model ($d = 3$).

We repeated the experiments for all 11595 feature tracks in the dataset with more than 10 views. All the methods were executed with $\epsilon = 1$ pixel and the same set of parameters. Table 6.6 lists the total number of inliers and runtime for all the methods over all tested points. With RANSAC initialization, EP-RS was able to achieve the highest total number of inliers followed by ADMM-RS. The triangulated result is shown in Figure 6.5.

Methods	Total inliers	Time (minutes)
RANSAC	91888	12.10
LO-RANSAC	94387	23.09
New LORANSAC	91555	20.84
ℓ_1 approximation	40669	11.16
ℓ_∞ outlier removal	43869	45.18
EP with RANSAC intialization	**99232**	49.52
EP with ℓ_∞ initialization	59996	71.86
ADMM with RANSAC initialization	97453	86.14
ADMM with ℓ_∞ initialization	49760	125.74

TABLE 6.6: Total inliers and runtime of triangulation for 11595 selected points with more than 10 views

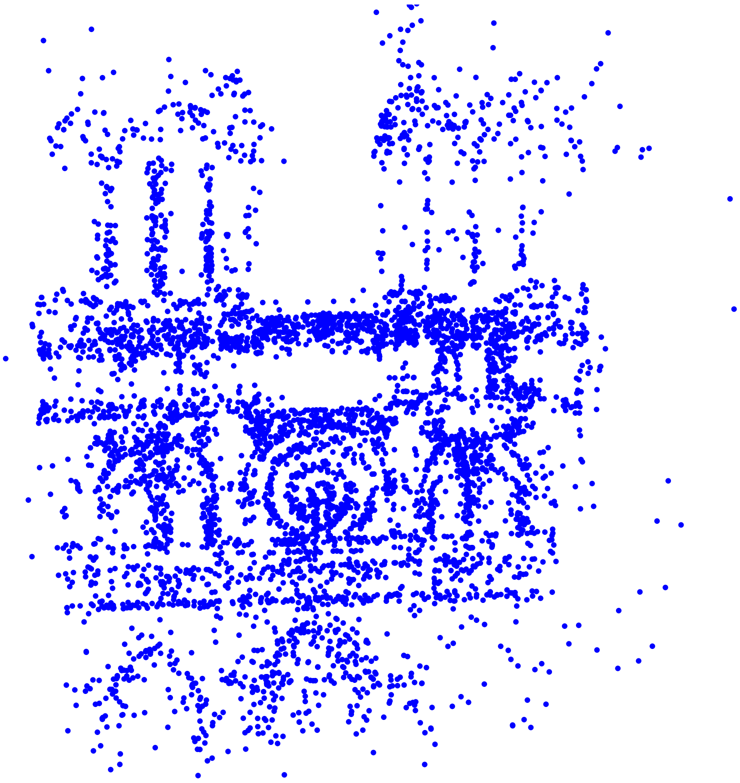


FIGURE 6.5: Qualitative results of EP-RS on triangulation.

6.7 Conclusions

We introduced two novel deterministic approximate algorithms for maximum consensus, based on non-smooth penalized method and ADMM. In terms of solution quality, our algorithms outperform other heuristic and approximate methods—this was demonstrated particularly by our methods being able to improve upon the solution of RANSAC. Even when presented with bad initializations (i.e., when using least squares to initialize on unbalanced data), our methods was able to recover and attain good solutions. Though our methods can be slower, their runtimes are still well within practical range (seconds to tens of seconds). In fact, at high outlier rates, our methods is actually faster than the RANSAC variants, while yielding higher-quality results.

Overall, the experiments illustrate that the proposed method can serve well in settings where slight additional runtime is a worthwhile expense for guaranteed convergence to an improved maximum consensus solution.

6.8 Supplementary Material

6.8.1 Convergence proof for the ADMM-based algorithm

Let $\beta = 2\rho$ and $\boldsymbol{\gamma} = \beta\boldsymbol{\lambda}$. The augmented Lagrangian (eq. (36) in the main paper) can be rewritten in the un-scaled ADMM form as

$$\begin{aligned}\mathcal{L}_\beta(\mathbf{z}, \{\mathbf{z}'_i\}, \mathbf{z}_C, \boldsymbol{\gamma}) = \; & \sum_i \left[u'_i + \mathbb{I}_B(\mathbf{z}'_i)\right] + \mathbb{I}_C(\mathbf{z}_C) + \|\mathbf{u}\|^2 \\ & + (\boldsymbol{\gamma}^{\mathbf{u}})^T(\mathbf{u}' - \mathbf{u}) + \frac{\beta}{2}\|\mathbf{u}' - \mathbf{u}\|_2^2 \\ & + (\boldsymbol{\gamma}^{\mathbf{s}})^T(\mathbf{s}' - \mathbf{s}) + \frac{\beta}{2}\|\mathbf{s}' - \mathbf{s}\|_2^2 \\ & + (\boldsymbol{\gamma}^{\mathbf{s}}_C)^T(\mathbf{s}_C - \mathbf{s}) + \frac{\beta}{2}\|\mathbf{s}_C - \mathbf{s}\|_2^2 \\ & + (\boldsymbol{\gamma}^{\mathbf{v}}_C)^T(\mathbf{v}_C - \mathbf{v}) + \frac{\beta}{2}\|\mathbf{v}_C - \mathbf{v}\|_2^2 \\ & + \sum_i \left[(\boldsymbol{\gamma}^{\mathbf{v}}_i)^T(\mathbf{v}'_i - \mathbf{v}) + \frac{\beta}{2}\|\mathbf{v}'_i - \mathbf{v}\|_2^2\right].\end{aligned} \tag{6.58}$$

For the ease of notation, the auxiliary variables $\{\mathbf{z}'_i\}$ and $\mathbf{z}_C$ can be collected into the vector $\mathbf{x}$, where

$$\mathbf{x} = [\{\mathbf{z}'^T_i\} \; \mathbf{z}^T_C]^T. \tag{6.59}$$

Also, define the functions $h(\mathbf{z})$ and $f(\mathbf{x})$ as

$$h(\mathbf{z}) = \|\mathbf{u}\|^2, \tag{6.60}$$

$$f(\mathbf{x}) = \sum_i u'_i. \tag{6.61}$$

With the definition of $\mathbf{x}$ in (6.59), the coupling constraints can be written in the following form

$$\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{0}, \tag{6.62}$$

where the matrix $\mathbf{B}$ is defined as:

$$\mathbf{B} = [\{(\mathbf{B}_i)^T\} \; (\mathbf{B}_C)^T]^T.$$

The matrix $\mathbf{B}$ is a collection of the selection sub-matrices, where each of the $\mathbf{B}_i$ can be defined as

$$\mathbf{B}_i \in \mathbb{R}^{(d+3)\times 2M+d+1} = \begin{bmatrix} \mathbf{0}_{1\times(i-1)} \; -1 \; \mathbf{0}_{1\times(M-i)} & \mathbf{0}_{1\times M} & \mathbf{0}_{1\times(d+1)} \\ \mathbf{0}_{1\times M} & \mathbf{0}_{1\times(i-1)} \; -1 \; \mathbf{0}_{1\times(M-i)} & \mathbf{0}_{1\times(d+1)} \\ \mathbf{0}_{(d+1)\times M} & \mathbf{0}_{(d+1)\times M} & -\mathbf{I}_{(d+1)\times(d+1)} \end{bmatrix},$$

and $\mathbf{B}_C$ is defined as:

$$\mathbf{B}_C \in \mathbb{R}^{(d+3)\times M+d+1} = \begin{bmatrix} \mathbf{0}_{N\times N} & \mathbf{0}_{N\times(N+d+1)} \\ \mathbf{0}_{N\times N} & -\mathbf{I}_{N\times(N+d+1)} \end{bmatrix}.$$

where $\mathbf{0}_{m\times n}$ and $\mathbf{I}_{m\times n}$ represent a zero matrix and an identity matrix of size $m\times n$, respectively.

Intuitively, the matrices $\mathbf{B}_i$ are to select u_i, s_i and $\mathbf{v}$, and $\mathbf{B}_C$ is to select $[\mathbf{s}^T \quad \mathbf{v}^T]^T$ from the variable $\mathbf{z}$. In other words,

$$\begin{aligned} \mathbf{B}_i\mathbf{z} = & \quad -[u_i \; s_i \; \mathbf{v}^T], \\ \mathbf{B}_C\mathbf{z} = & \quad -[\mathbf{s}^T \; \mathbf{v}^T]. \end{aligned} \tag{6.63}$$

Recall that since $\boldsymbol{\gamma} = \beta\boldsymbol{\lambda}$, the vector $\boldsymbol{\gamma}$ has the form of

$$\boldsymbol{\gamma} = \left[\{(\boldsymbol{\gamma}_i)^T\} \;\; (\boldsymbol{\gamma}_C)^T\right]^T.$$

In addition, note that with the changes of variables discussed above, the value of $\boldsymbol{\gamma}$ is updated at each iteration by the equation

$$\boldsymbol{\gamma}^+ = \boldsymbol{\gamma}^t + \beta(\mathbf{x}^+ + \mathbf{B}\mathbf{z}^+). \tag{6.64}$$

Thus, the augmented Lagangian function can now be written as

$$\mathcal{L}_\beta(\mathbf{x},\mathbf{z},\boldsymbol{\gamma}) \quad = f(\mathbf{x}) + h(\mathbf{z}) + \sum_i \mathbb{I}_B(\mathbf{z}'_i) + \mathbb{I}_C(\mathbf{z}_C) + \boldsymbol{\gamma}^T(\mathbf{x}+\mathbf{B}\mathbf{z}) + \frac{\beta}{2}\|\mathbf{x}+\mathbf{B}\mathbf{z}\|^2. \tag{6.65}$$

Monotonicity of the Lagrangian function Consider the $(t+1)$-th update cycle of Algorithm 2. Let $\mathbf{x}^t$, $\mathbf{z}^t$, $\boldsymbol{\gamma}^t$ denote the variables carried from the t-th iteration and $\mathbf{x}^+$, $\mathbf{z}^+$, $\boldsymbol{\gamma}^+$ represent the updated variables, i.e., $\mathbf{x}^{(t+1)}$, $\mathbf{z}^{(t+1)}$ and $\boldsymbol{\gamma}^{(t+1)}$, respectively.

As the update steps for the auxiliary variables, which involves minimizing (6.65) with respect to $\mathbf{x}$, can be solved optimally, the following inequailty holds:

$$\mathcal{L}_\beta(\mathbf{x}^t,\mathbf{z}^t,\boldsymbol{\gamma}^t) \geq \mathcal{L}_\beta(\mathbf{x}^+,\mathbf{z}^t,\boldsymbol{\gamma}^t). \tag{6.66}$$

After the original variable $\mathbf{z}$ and the Lagrangian multipliers $\boldsymbol{\gamma}$ are updated, consider the difference between the two Lagrangian functions,

$$D_\mathcal{L} = \mathcal{L}_\beta(\mathbf{x}^+,\mathbf{z}^t,\boldsymbol{\gamma}^t) - \mathcal{L}_\beta(\mathbf{x}^+,\mathbf{z}^+,\boldsymbol{\gamma}^+). \tag{6.67}$$

In the following, we will prove that with a sufficiently large β, $D_\mathcal{L} \geq 0$.

Since $\mathbf{z}^+$ minimizes $\mathcal{L}_\beta(\mathbf{x}^+, \mathbf{z}, \gamma^t)$ after the $\mathbf{z}$ update step, the following optimality condition holds:

$$\nabla\mathcal{L}(\mathbf{z}^+) = \nabla h(\mathbf{z}^+) + \mathbf{B}^T\gamma^t + \mathbf{B}^T\beta(\mathbf{x}^+ + \mathbf{B}\mathbf{z}^+) = \mathbf{0}. \tag{6.68}$$

Due to the fact that γ^+ is updated by $\gamma^+ = \gamma^t + \beta(\mathbf{x}^+ + \mathbf{B}\mathbf{z}^+)$, (6.68) is equivalent to

$$\mathbf{B}^T\gamma^+ = -\nabla h(\mathbf{z}^+). \tag{6.69}$$

Henceforth, let $\mathbf{B}_{M+1} = \mathbf{B}_C$ and $\gamma_{M+1} = \gamma_C$. Then $\mathbf{B} = [\{(\mathbf{B}_i)^T\}_{i=1}^{M+1}]^T$ and $\gamma = [\{\gamma_i\}_{i=1}^{M+1}]^T$. Observe that $\mathbf{B}^T\gamma^+$ can be written as the summation

$$\mathbf{B}^T\gamma^+ = \sum_{i=1}^{M+1} \mathbf{B}_i^T\gamma_i^+. \tag{6.70}$$

Combining (6.69) and (6.70) results in

$$\mathbf{B}_i^T\gamma_i^+ = -\nabla h(\mathbf{z}^+) - \sum_{j=1, j\neq i}^{M+1} \mathbf{B}_j^T\gamma_j^+, \quad \forall i = 1\ldots(M+1). \tag{6.71}$$

From (6.65) and (6.67), after some manipulations, $D_\mathcal{L}$ can be written as

$$\begin{aligned} D_\mathcal{L} &= h(\mathbf{z}^t) - h(\mathbf{z}^+) + (\gamma^+)^T(\mathbf{B}\mathbf{z}^t - \mathbf{B}\mathbf{z}^+) + \frac{\beta}{2}\|\mathbf{B}\mathbf{z}^+ - \mathbf{B}\mathbf{z}^t\|^2 - \frac{1}{\beta}\|\gamma^+ - \gamma^t\|^2 \\ &= h(\mathbf{z}^t) - h(\mathbf{z}^+) + (\gamma^+)^T(\mathbf{B}\mathbf{z}^t - \mathbf{B}\mathbf{z}^+) + \frac{\beta}{2}\|\mathbf{B}\mathbf{z}^+ - \mathbf{B}\mathbf{z}^t\|^2 - \frac{1}{\beta}\sum_{i=1}^{M+1}\|\gamma_i^+ - \gamma_i^t\|^2. \end{aligned} \tag{6.72}$$

For each matrix $\mathbf{B}_i$, let κ_i denotes the smallest strictly positive eigenvalue of $\mathbf{B}_i^T\mathbf{B}_i$. Since $\gamma_i^+ = \gamma_i^t + \beta((\mathbf{z}_i')^+ + \mathbf{B}_i\mathbf{z}^+) \in \text{Im}(\mathbf{B}_i)$, following [63, Lemma 2],

$$\|\gamma_i^+ - \gamma_i^t\| \quad \leq \kappa_i\|\mathbf{B}_i^T(\gamma_i^+ - \gamma_i^t)\|. \tag{6.73}$$

Making use of (6.71), then apply the triangle inequality,

$$\begin{aligned} \|\mathbf{B}_i^T(\gamma_i^+ - \gamma_i)\| \quad &= \|\nabla h(\mathbf{z}^+) - \nabla h(\mathbf{z}^t) - \sum_{j\neq i}\mathbf{B}_j^T(\gamma_j^+ - \gamma_j^t)\| \\ &\geq |\|\nabla h(\mathbf{z}^+) - \nabla h(\mathbf{z}^t)\| - \sum_{j\neq i}\|\mathbf{B}_j^T(\gamma_j^+ - \gamma_j^t)\||. \end{aligned} \tag{6.74}$$

It then follows that

$$-\|\mathbf{B}_i^T(\gamma_i^+ - \gamma_i^t)\| \leq \|\nabla h(\mathbf{z}^+) - \nabla h(\mathbf{z}^t)\| - \sum_{j\neq i}\|\mathbf{B}_j^T(\gamma_j^+ - \gamma_j^t)\|. \tag{6.75}$$

Summing the LHS and RHS over all $i = 1 \dots (M+1)$,

$$\begin{aligned}
-\sum_i \|\mathbf{B}_i^T(\boldsymbol{\gamma}_i^+ - \boldsymbol{\gamma}_i^t)\| &\leq (M+1)\|\nabla h(\mathbf{z}^+) - \nabla h(\mathbf{z}^t)\| - M\sum_i \|\mathbf{B}_j^T(\boldsymbol{\gamma}_j^+ - \boldsymbol{\gamma}_j^t)\|, \\
(M-1)\sum_i \|\mathbf{B}_i^T(\boldsymbol{\gamma}_i^+ - \boldsymbol{\gamma}_i^t)\| &\leq (M+1)\|\nabla h(\mathbf{z}^+) - \nabla h(\mathbf{z}^t)\|, \\
\sum_i \|\mathbf{B}_i^T(\boldsymbol{\gamma}_i^+ - \boldsymbol{\gamma}_i^t)\| &\leq \frac{M+1}{M-1}\|\nabla h(\mathbf{z}^+) - \nabla h(\mathbf{z}^t)\|.
\end{aligned} \tag{6.76}$$

Therefore, from (6.73),

$$\|\boldsymbol{\gamma}_i^+ - \boldsymbol{\gamma}_i^t\| \quad \leq \kappa_i\|\mathbf{B}_i^T(\boldsymbol{\gamma}_i^+ - \boldsymbol{\gamma}_i^t)\| \leq \kappa_i\frac{M+1}{M-1}\|\nabla h(\mathbf{z}^+) - \nabla h(\mathbf{z}^t)\| \leq C_i\|\mathbf{u}^+ - \mathbf{u}^t\|, \tag{6.77}$$

where $C_i = 2\kappa_i\frac{M+1}{M-1}$. The last inequality holds due to $\nabla h(\mathbf{z}) = 2\mathbf{u}$.

The inequality (6.77) results in:

$$\begin{aligned}
\|\boldsymbol{\gamma}_i^+ - \boldsymbol{\gamma}_i^t\|^2 &\leq C_i^2\|\mathbf{u}^+ - \mathbf{u}^t\|^2, \\
\sum_i \|\boldsymbol{\gamma}_i^+ - \boldsymbol{\gamma}_i^t\|^2 &\leq \left(\sum_i C_i^2\right)\|\mathbf{u}^+ - \mathbf{u}^t\|^2.
\end{aligned} \tag{6.78}$$

Let $C = \left(\sum_i C_i^2\right)$, we now have

$$\begin{aligned}
\|\boldsymbol{\gamma}^+ - \boldsymbol{\gamma}^t\|^2 = \sum_i \|\boldsymbol{\gamma}_i^+ - \boldsymbol{\gamma}_i^t\|^2 &\leq C\|\mathbf{u}^+ - \mathbf{u}^t\|^2 \\
-\frac{1}{\beta}\|\boldsymbol{\gamma}^+ - \boldsymbol{\gamma}^t\|^2 &\geq -\frac{C}{\beta}\|\mathbf{u}^+ - \mathbf{u}^t\|^2
\end{aligned} \tag{6.79}$$

Therefore, it follows that

$$\begin{aligned}
D_{\mathcal{L}} &= h(\mathbf{z}^t) - h(\mathbf{z}^+) + (\mathbf{B}^T\boldsymbol{\gamma}^+)^T(\mathbf{z}^t - \mathbf{z}^+) + \frac{\beta}{2}\|\mathbf{B}\mathbf{z}^+ - \mathbf{B}\mathbf{z}^t\|^2 - \frac{1}{\beta}\|\boldsymbol{\gamma}^+ - \boldsymbol{\gamma}^t\|^2 \\
&= h(\mathbf{z}^t) - h(\mathbf{z}^+) - (\nabla h(\mathbf{z}^+))^T(\mathbf{z}^t - \mathbf{z}^+) + \frac{\beta}{2}\|\mathbf{B}\mathbf{z}^+ - \mathbf{B}\mathbf{z}^t\|^2 - \frac{1}{\beta}\|\boldsymbol{\gamma}^+ - \boldsymbol{\gamma}^t\|^2 \\
&\geq h(\mathbf{z}^t) - h(\mathbf{z}^+) - (\nabla h(\mathbf{z}^+))^T(\mathbf{z}^t - \mathbf{z}^+) + \frac{\beta}{2}\|\mathbf{B}\mathbf{z}^+ - \mathbf{B}\mathbf{z}^t\|^2 - \frac{C}{\beta}\|\mathbf{z}^+ - \mathbf{z}^t\|^2.
\end{aligned} \tag{6.80}$$

Based on Taylor expansion of the funcion $h(\mathbf{z})$ around $\mathbf{z}^+$

$$h(\mathbf{z}^t) \geq h(\mathbf{z}^+) + (\nabla h(\mathbf{z}^+))^T(\mathbf{z}^t - \mathbf{z}^+) + (\mathbf{z}^t - \mathbf{z}^+)^T\nabla^2 h(\mathbf{z}^+)(\mathbf{z}^t - \mathbf{z}^+) \tag{6.81}$$

Thus,

$$\begin{aligned}
h(\mathbf{z}^t) - h(\mathbf{z}^+) - (\nabla h(\mathbf{z}^+))^T(\mathbf{z}^t - \mathbf{z}^+) &\geq (\mathbf{z}^t - \mathbf{z}^+)^T\nabla^2 h(\mathbf{z}^+)(\mathbf{z}^t - \mathbf{z}^+) \\
&\geq \|\mathbf{u}^+ - \mathbf{u}^t\|^2
\end{aligned} \tag{6.82}$$

Also, note that

$$\|\mathbf{B}\mathbf{z}^{+}-\mathbf{B}\mathbf{z}^{t}\|^2 \quad = \|\mathbf{u}^{+}-\mathbf{u}^{t}\|^2+2\|\mathbf{s}^{+}-\mathbf{s}^{t}\|^2+(M+1)\|\mathbf{s}^{+}-\mathbf{s}^{t}\|^2 \geq \|\mathbf{u}^{+}-\mathbf{u}^{t}\|^2. \tag{6.83}$$

From (6.80), (6.82) and (6.83),

$$D_{\mathcal{L}} \geq \left(\frac{\beta}{2}+1-\frac{C}{\beta}\right)\|\mathbf{u}^{+}-\mathbf{u}^{t}\|^2. \tag{6.84}$$

Thus, with a sufficiently large β such that

$$\beta^2+2-2C \geq 0, \tag{6.85}$$

$D_{\mathcal{L}}$ is monotonically non-increasing, i.e.,

$$D^t_{\mathcal{L}} \geq 0, \quad \forall t \tag{6.86}$$

Then, from (6.66) and (6.67), with $D_L \geq 0$, it follows that:

$$\mathcal{L}_\beta(\mathbf{x}^{+},\mathbf{z}^{+},\boldsymbol{\gamma}^{+}) \geq \mathcal{L}_\beta(\mathbf{x}^{t},\mathbf{z}^{t},\boldsymbol{\gamma}^{t}), \quad \forall t. \tag{6.87}$$

Boundedness of $\mathcal{L}_\beta$ Besides the monotonicity of the Lagrangian function, we now prove that the Lagrangian function $\mathcal{L}_\beta$ will be lower bounded by a proper choice of β. Indeed, $\mathcal{L}_\beta$ can be written as

$$\begin{aligned}
\mathcal{L}_\beta \quad &= f(\mathbf{x}^t)+h(\mathbf{z}^t)+\boldsymbol{\gamma}^T(\mathbf{x}^t+\mathbf{B}\mathbf{z}^t)+\frac{\beta}{2}\|\mathbf{x}^t+\mathbf{B}\mathbf{z}^t\|^2\\
&= f(\mathbf{x}^t)+h(\mathbf{z}^t)+\frac{1}{\beta}(\boldsymbol{\gamma}^t)^T(\boldsymbol{\gamma}^t-\boldsymbol{\gamma}^{(t-1)})+\frac{1}{2\beta}\|\boldsymbol{\gamma}^t-\boldsymbol{\gamma}^{(t-1)}\|^2\\
\text{(Cauchy inequality)} \qquad &\geq f(\mathbf{x}^t)+h(\mathbf{z}^t)-\frac{1}{\beta}\|\boldsymbol{\gamma}^t\|\|\boldsymbol{\gamma}^t-\boldsymbol{\gamma}^{(t-1)}\|+\frac{1}{2\beta}\|\boldsymbol{\gamma}^t-\boldsymbol{\gamma}^{(t-1)}\|^2.
\end{aligned} \tag{6.88}$$

From (6.79), it follows that

$$\|\boldsymbol{\gamma}^t-\boldsymbol{\gamma}^{(t-1)}\|^2 \quad \leq C\|\mathbf{u}^t-\mathbf{u}^{(t-1)}\|^2 \leq KC, \tag{6.89}$$

where K is a positive number. The last inequality holds due to the fact that $\|\mathbf{u}^{+}-\mathbf{u}\|^2$ is upper bounded. Thus,

$$\|\boldsymbol{\gamma}^t-\boldsymbol{\gamma}^{(t-1)}\| \quad \leq \sqrt{KC}. \tag{6.90}$$

Denote $\sqrt{KC}$ by C', (6.88) now becomes

$$\mathcal{L}_\beta \quad \geq f(\mathbf{x})+h(\mathbf{z})-\frac{C'}{\beta}\|\boldsymbol{\gamma}^t\|+\frac{1}{2\beta}\|\boldsymbol{\gamma}^t-\boldsymbol{\gamma}^{(t-1)}\|^2. \tag{6.91}$$

Since $f(\mathbf{x}) = \sum_i u'_i$ and $h(\mathbf{z}) = \|\mathbf{u}\|^2$, $f(\mathbf{x}) + h(\mathbf{z})$ is lower-bounded by 0. As β is allowed to increase at each iteration of the algorithm, in order for $\mathcal{L}_\beta$ to be lower-bounded, we would like to select β at iteration $(t+1)$-th such that following inequality holds

$$-\frac{C'}{\beta}\|\boldsymbol{\gamma}^+\| + \frac{1}{2\beta}\|\boldsymbol{\gamma}^+ + \boldsymbol{\gamma}^{(t)}\|^2 \geq L_0. \tag{6.92}$$

where $L_0 > -\infty$ is a large negative value to prevent $\mathcal{L}_\beta$ to drift to $-\infty$. Note that at the iteration $(t+1)$-th, the value of $\boldsymbol{\gamma}^+$ is not known in advance, but by applying (6.90), we have $-C' \leq \|\boldsymbol{\gamma}^+\| - \|\boldsymbol{\gamma}^t\| \leq C'$, then we can bound $\|\boldsymbol{\gamma}^+\|$ by writing

$$-\|\boldsymbol{\gamma}^+\| \geq -C' - \|\boldsymbol{\gamma}^t\|, \tag{6.93a}$$

$$\|\boldsymbol{\gamma}^+\| \geq -C' + \|\boldsymbol{\gamma}^t\|. \tag{6.93b}$$

Let D denotes $-C' - \|\boldsymbol{\gamma}^t\|$ and E denotes $-C' + \|\boldsymbol{\gamma}^t\|$, (6.91) can be rewritten as

$$\begin{aligned}
-\frac{C'}{\beta}\|\boldsymbol{\gamma}^+\| + \frac{1}{2\beta}\|\boldsymbol{\gamma}^+ - \boldsymbol{\gamma}^t\|^2 \quad &\geq \frac{C'}{\beta}D + \frac{1}{2\beta}(\|\boldsymbol{\gamma}^+\| - \|\boldsymbol{\gamma}^t\|)^2 \\
&\geq \frac{C'}{\beta}D + \frac{1}{2\beta}(\|\boldsymbol{\gamma}^+\|^2 - 2\|\boldsymbol{\gamma}^+\|\|\boldsymbol{\gamma}^t\| + \|\boldsymbol{\gamma}^t\|^2) \\
&\geq \frac{C'}{\beta}D + \frac{1}{2\beta}(E^2 - 2\|\boldsymbol{\gamma}^+\|\|\boldsymbol{\gamma}^t\| + \|\boldsymbol{\gamma}^t\|^2) \\
&\geq \frac{C'}{\beta}D + \frac{1}{2\beta}(E^2 - 2D\|\boldsymbol{\gamma}^t\| + \|\boldsymbol{\gamma}^t\|^2) \\
&\geq \frac{C'D - D\|\boldsymbol{\gamma}^t\|}{\beta}.
\end{aligned} \tag{6.94}$$

In other for (6.92) to hold, the RHS of (6.94) must be greater than L_0, or:

$$\frac{C'D - D\|\boldsymbol{\gamma}^t\|}{\beta} \geq L_0. \tag{6.95}$$

Then, β must be chosen such that:

$$\beta L_0 \leq C'D - D\|\boldsymbol{\gamma}^t\|. \tag{6.96}$$

Divide both sides of (6.96) by L_0 (note that $L_0 < 0$), we get the condition for β

$$\beta \geq \frac{C'D - D\|\boldsymbol{\gamma}^t\|}{L_0}. \tag{6.97}$$

Therefore, with a sufficiently large β such that (6.85) and (6.97) hold, the Lagrangian function $\mathcal{L}_\beta$ is non-increasing and bounded from below. Theferefore, it converges after a finite number of iterations:

$$\lim_{k\to\infty} \|\mathbf{z}^{k+1} - \mathbf{z}^k\| = 0. \tag{6.98}$$

Also, due to (6.90)

$$\lim_{k\to\infty} \|\gamma^{k+1} - \gamma^{k}\| = 0, \tag{6.99}$$

which also leads to

$$\lim_{k\to\infty} \|\mathbf{x}^{k} + \mathbf{B}\mathbf{z}^{k}\| = 0. \tag{6.100}$$

Equation (6.100) states that the auxiliary variables and the original variables converge to the same solution. The equations (6.98), (6.99) and (6.100) then complete the proof.

6.8.2 Derivation of KKT conditions

The penalty problem derived in the main paper:

$$\begin{aligned}\min_{\mathbf{u},\mathbf{s},\mathbf{v}} \quad & \sum_i u_i + \alpha \left[u_i(s_i - \mathbf{c}_i^T\mathbf{v} + b_i) + s_i(1-u_i)\right] \\ \text{s.t.} \quad & s_i - \mathbf{c}_i^T\mathbf{v} + b_i \geq 0, \\ & 1 - u_i \geq 0, \\ & s_i, u_i, v_i \geq 0.\end{aligned} \tag{6.101}$$

Define the functions correspond to the set of constraints of the penalty problem (6.101):

$$\begin{aligned}\mathcal{H}_i &= s_i - \mathbf{c}_i^T\mathbf{v} + b_i \\ \mathcal{G}_i &= 1 - u_i \\ \mathcal{S}_i &= s_i \\ \mathcal{U}_i &= u_i \\ \mathcal{V}_i &= v_i\end{aligned} \tag{6.102}$$

Also, define $\boldsymbol{\lambda}^{\mathcal{H}}$, $\boldsymbol{\lambda}^{\mathcal{G}}$, $\boldsymbol{\lambda}^{\mathcal{S}}$, $\boldsymbol{\lambda}^{\mathcal{U}}$, $\boldsymbol{\lambda}^{\mathcal{V}} \in \mathbb{R}^M$ be the Larange multipliers for the constraints in (6.101).

Derivaties of the cost function in (6.101) with respect to $\mathbf{u}$, $\mathbf{v}$, $\mathbf{s}$ respectively:

$$\begin{aligned}\nabla_{\mathbf{u}} P(\mathbf{u},\mathbf{s},\mathbf{v}|\alpha) &= -\alpha\mathbf{C}\mathbf{v} + \alpha\mathbf{b} + \mathbf{1} \\ \nabla_{\mathbf{s}} P(\mathbf{u},\mathbf{s},\mathbf{v}|\alpha) &= \alpha\mathbf{1} \\ \nabla_{\mathbf{v}} P(\mathbf{u},\mathbf{s},\mathbf{v}|\alpha) &= -\alpha\mathbf{C}^T\mathbf{u}\end{aligned}$$

$\mathbf{u}, \mathbf{s}, \mathbf{v}$ is a stationary point if the KKT condition is satisfied:

$$
\begin{aligned}
&\nabla_{\mathbf{u}} P - \sum_i \lambda_i^{\mathcal{G}} \nabla_u \mathcal{G}_i + \lambda_i^{\mathcal{U}} \nabla_u \mathcal{U}_i = \mathbf{0} \\
&\nabla_{\mathbf{s}} P - \sum_i \lambda_i^{\mathcal{H}} \nabla_s \mathcal{H}_i + \lambda_i^{\mathcal{S}} \nabla_s \mathcal{S}_i = \mathbf{0} \\
&\nabla_{\mathbf{v}} P - \sum_i \lambda_i^{\mathcal{H}} \nabla_v \mathcal{H}_i + \lambda_i^{\mathcal{V}} \nabla_v \mathcal{V}_i = \mathbf{0} \\
&\lambda_i^{\mathcal{H}} \mathcal{H}_i = 0 \\
&\lambda_i^{\mathcal{G}} \mathcal{G}_i = 0 \\
&\lambda_i^{\mathcal{S}} \mathcal{S}_i = 0 \\
&\lambda_i^{\mathcal{U}} \mathcal{U}_i = 0 \\
&\lambda_i^{\mathcal{V}} \mathcal{V}_i = 0 \\
&s_i - c_i^T v + b_i \geq 0 \\
&1 - u_i \geq 0 \\
&u_i, v_i, s_i \geq 0 \\
&\lambda_i^{\mathcal{H}}, \lambda_i^{\mathcal{G}}, \lambda_i^{\mathcal{U}}, \lambda_i^{\mathcal{V}}, \lambda_i^{\mathcal{S}} \geq 0
\end{aligned}
\tag{6.103}
$$

which is equivalent to:

$$
\begin{aligned}
-\alpha \mathbf{C}\mathbf{v} + \alpha \mathbf{b} + \mathbf{1} + \boldsymbol{\lambda}^{\mathcal{G}} - \boldsymbol{\lambda}^{\mathcal{U}} &= \mathbf{0} \\
\alpha \mathbf{1} - \boldsymbol{\lambda}^{\mathcal{H}} - \boldsymbol{\lambda}^{\mathcal{S}} &= \mathbf{0} \\
-\alpha \mathbf{C}^T \mathbf{u} + \mathbf{C}^T \boldsymbol{\lambda}^{\mathcal{H}} - \boldsymbol{\lambda}^{\mathcal{V}} &= \mathbf{0} \\
\lambda_i^{\mathcal{H}} (s_i - c_i^T v + b_i) &= 0 \\
\lambda_i^{\mathcal{G}} (1 - u_i) &= 0 \\
\lambda_i^{\mathcal{V}} v_i &= 0 \\
\lambda_i^{\mathcal{U}} u_i &= 0 \\
\lambda_i^{\mathcal{S}} s_i &= 0 \\
\lambda_i^{\mathcal{H}}, \lambda_i^{\mathcal{G}}, \lambda_i^{\mathcal{U}}, \lambda_i^{\mathcal{V}}, \lambda_i^{\mathcal{S}} &\geq 0 \\
u_i, v_i, s_i &\geq 0 \\
s_i - c_i^T v + b_i &\geq 0 \\
1 - u_i &\geq 0
\end{aligned}
\tag{6.104}
$$

By rearranging and substitution, (6.104) can be reduced to

$$
\begin{aligned}
\mathbf{u}^T(-\alpha\mathbf{C}\mathbf{v} + \alpha\mathbf{b} + \mathbf{1} + \boldsymbol{\lambda}^{\mathcal{G}}) &= 0 \\
\mathbf{s}^T(\alpha\mathbf{1} - \boldsymbol{\lambda}^{\mathcal{H}}) &= 0 \\
\mathbf{v}^T(-\alpha\mathbf{C}^T\mathbf{u} + \mathbf{C}^T\boldsymbol{\lambda}^{\mathcal{H}}) &= 0 \\
(\boldsymbol{\lambda}^{\mathcal{H}})^T(\mathbf{s} - \mathbf{C}\mathbf{v} + \mathbf{b}) &= 0 \\
(\boldsymbol{\lambda}^{\mathcal{G}})^T(\mathbf{1} - \mathbf{u}) &= 0 \\
\mathbf{s} - \mathbf{C}\mathbf{v} + \mathbf{b} &\geq \mathbf{0} \\
\mathbf{1} - \mathbf{u} &\geq \mathbf{0} \\
\boldsymbol{\lambda}^{\mathcal{H}}, \boldsymbol{\lambda}^{\mathcal{G}}, \mathbf{u}, \mathbf{s}, \mathbf{v} &\geq \mathbf{0}
\end{aligned}
\tag{6.105}
$$

Chapter 7

Conclusion and Future Work

Consensus maximization is an important problem that underpins a large number of computer vision applications. This thesis contributes several algorithmic developments for the maximum consensus problem. This chapter summarizes the contributions of this thesis (Section 7.1) and discusses future research directions (Section 7.2)

7.1 Summary of Contributions

The contributions of this thesis can be summarized as follows

- A new guided sampling algorithm is proposed (Chapter 3) that takes advantage of the underlying tree structure of consensus maximization to guide the sampling process. The new method works under the framework of LP-type problem and Monte-Carlo Tree Search and does not require domain knowledge. Empirical results show that the new algorithm outperforms RANSAC and its variants for many popular robust model fitting problems in computer vision.
- A globally optimal algorithm (based on Branch-and-Bound strategy) to estimate the Möbius transformation is proposed in Chapter 4 to robustly align non-rigid shapes with disc topology. The proposed method is much more efficient than existing methods for aligning isometric shapes.
- Two deterministic approximate methods for consensus maximization were proposed. One is based on Frank-Wolfe algorithm (Chapter 5) and another is based on ADMM algorithm (Chapter 6). These two algorithms bridge the gap between the class of randomized methods and exact methods and can consistently upgrade an initial rough estimate to a higher quality solution with slight increase in runtime.

7.2 Future research directions

7.2.1 Integration of domain knowledge into the random tree search algorithm

Chapter 3 describes a new guided sampling scheme (RATSAC) that relies solely on the underlying tree structure of the consensus maximization problem. Meanwhile, based on the results of previous guided sampling approaches such as PROSAC [22] or Guided-MLESAC [84], the use of domain knowledge (if available) has been shown to achieve relatively good results. Thus, a promising research topic that can be explored is to integrate the prior domain knowledge information into RATSAC to develop a better guided sampling strategy that utilizes both the domain knowledge and the underlying problem structure.

7.2.2 Optimal Möbius search for shapes with spherical topology

The Möbius alignment has been shown to be an effective method for the task of non-rigid isometric shape alignment. Chapter 4 discusses a globally optimal algorithm to estimate a Möbius transformation between two hyperbolic discs. So far, the proposed algorithm can only be used to handle shapes with disc topology. On the other hand, in order to apply the conformal alignment approach for shapes with spherical topology, it is required that the shapes be conformally mapped onto hyperbolic spheres. In such scenario, the Möbius search contains 6 degrees of freedom (DoF). One of the research direction is to extend the existing Möbius search algorithm developed in Chapter 4 to handle data on the unit spheres.

7.2.3 Improving the convergence rate for the approximate methods

Chapter 5 and Chapter 6 provide two algorithms to approximate the maximum consensus problem with provably convergence guarantee. Though efficient, their convergent rates are yet to be carefully analyzed. Thus, the future work is to investigate their convergence rate to theoretically validate the effectiveness of each algorithm. Furthermore, from the understanding of their convergence rate, other optimization schemes can be applied to improve the solution quality and runtime.

Bibliography

[1] http://mathworld.wolfram.com/Circle-CircleIntersection.html.

[2] http://mathworld.wolfram.com/ComplexNumber.html.

[3] https://www.cs.princeton.edu/~vk/projects/CorrsCode/.

[4] https://en.wikipedia.org/wiki/Quadratic_assignment_problem.

[5] K. Aftab and R. Hartley. "Convergence of iteratively re-weighted least squares to robust M-estimators". In: *2015 IEEE Winter Conference on Applications of Computer Vision*. IEEE. 2015, pp. 480–487.

[6] S. Agarwal, N. Snavely, and S. M. Seitz. "Fast algorithms for L-inifity problems in multiview geometry". In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, pp. 1–8.

[7] S. Angenent, S. Haker, A. Tannenbaum, and R. Kikinis. "Conformal geometry and brain flattening". In: *MICCAI*. 1999.

[8] P. Auer, N. Cesa-Bianchi, and P. Fischer. "Finite-time analysis of the multiarmed bandit problem". In: *Machine learning* 47.2 (2002), pp. 235–256.

[9] L. Bers. "Uniformization, moduli and Kleinian groups". In: *Bull. London Math. Soc.* Vol. 4. 1972, pp. 257–300.

[10] P. J. Besl and N. D. MacKay. "A method for registration of 3-d shapes". In: *IEEE TPAMI* 14.2 (1992), pp. 239–256.

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. "Distributed optimization and statistical learning via the alternating direction method of multipliers". In: *Foundations and Trends® in Machine Learning* 3.1 (2011), pp. 1–122.

[12] D. Boyer, Y. Lipman, E. StClair, J. Puente, B. Patel, T. Funkhouser, J. Jernvall, and I. Daubechies. "Algorithms to automatically quantify the geometry similarity of anatomical surfaces". In: *Proc. Nat'l Academy of Sciences*. 2011.

[13] T. Breuel. "Implementation techniques for geometric branch-and-bound matching methods". In: *CVIU* 90.3 (2003), pp. 258–294.

[14] A. M. Bronstein, M. M. Bronstein, A. M. Bruckstein, and R. Kimmel. “Analysis of two-dimensional non-rigid shapes”. In: *IJCV* 78.1 (2008), pp. 67–88.

[15] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton. “A survey of monte carlo tree search methods”. In: *IEEE Transactions on Computational Intelligence and AI in games* 4.1 (2012), pp. 1–43.

[16] E. Cheney. “Introduction to approximation theory. 1966”. In: *Chelsea, New York* ().

[17] T.-J. Chin, Y. Heng Kee, A. Eriksson, and F. Neumann. “Guaranteed outlier removal with mixed integer linear programs”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 5858–5866.

[18] T.-J. Chin, P. Purkait, A. Eriksson, and D. Suter. “Efficient globally optimal consensus maximisation with tree search”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 2413–2421.

[19] T.-J. Chin and D. Suter. “The Maximum Consensus Problem: Recent Algorithmic Advances”. In: *Synthesis Lectures on Computer Vision* 7.2 (2017), pp. 1–194.

[20] J. W. Chinneck. *Feasibility and Infeasibility in Optimization:Algorithms and Computational Methods*. Vol. 118. Springer Science & Business Media, 2007.

[21] S. Choi, T. Kim, and W. Yu. “Performance evaluation of RANSAC family”. In: *British Machine Vision Conference (BMVC)*. 2009.

[22] O. Chum and J. Matas. “Matching with PROSAC-progressive sample consensus”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 1. IEEE. 2005, pp. 220–226.

[23] O. Chum, J. Matas, and J. Kittler. “Locally optimized RANSAC”. In: *DAGM*. Springer. 2003.

[24] M. De Berg, M. Van Kreveld, M. Overmars, and O. C. Schwarzkopf. *Computational geometry*. Springer, 2000.

[25] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi. “The farthest point strategy for progressive image sampling”. In: *Image Processing, IEEE Transactions on* 6.9 (1997), pp. 1305–1315.

[26] O. Enqvist, E. Ask, F. Kahl, and K. Åström. “Robust fitting for multiple view geometry”. In: *European Conference on Computer Vision*. Springer. 2012, pp. 738–751.

[27] D. Eppstein. “Quasiconvex programming”. In: *Combinatorial and Computational Geometry* 52.287-331 (2005), p. 3.

[28] A. Eriksson, J. Bastian, T.-J. Chin, and M. Isaksson. “A Consensus-Based Framework for Distributed Bundle Adjustment”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 1754–1762.

[29] A. Eriksson and M. Isaksson. “Pseudoconvex Proximal Splitting for L-infinity Problems in Multiview Geometry”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 4066–4073.

[30] A. Eriksson, M. Isaksson, and T.-J. Chin. “High breakdown bundle adjustment”. In: *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*. IEEE. 2015, pp. 310–317.

[31] M. A. Fischler and R. C. Bolles. “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”. In: *Communications of the ACM* 24.6 (1981), pp. 381–395.

[32] M. Frank and P. Wolfe. “An algorithm for quadratic programming”. In: *Naval Research Logistics Quaterly* 3.95 (1956).

[33] J. Fredriksson, V. Larsson, and C. Olsson. “Practical robust two-view translation estimation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 2684–2690.

[34] J. Fredriksson, V. Larsson, C. Olsson, and F. Kahl. “Optimal Relative Pose with Unknown Correspondences.” In: *CVPR*. 2016, pp. 1728–1736.

[35] X. Gu, Y. Wang, T. Chan, P. Thompson, and S.-T. Yau. “Genus zero surface conformal mapping and its application to brain surface mapping”. In: *IEEE Trans. Medical Imaging* 23.8 (2004), pp. 949–958.

[36] X. Gu and S.-T. Yau. “Global conformal surface parametrization”. In: *Proc. Eurographics Symp. Geometry Processing*. 2003.

[37] X. D. Gu and B. C. Vemuri. “Matching 3D shapes using 2D conformal representations”. In: *MICCAI*. 2004.

[38] X. D. Gu and S.-T. Yau. *Computational conformal geometry*. Advanced Lectures in Mathematics. Higher Education Press, 2008.

[39] R. I. Hartley and F. Kahl. “Global optimization through rotation space search”. In: *IJCV* 82 (2009), pp. 64–79.

[40] R. Hartley and F. Kahl. “Optimal algorithms in multiview geometry”. In: *Asian conference on computer vision*. Springer. 2007, pp. 13–34.

[41] R. Hartley and F. Schaffalitzky. “L/sub/spl infin//minimization in geometric reconstruction problems”. In: *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. Vol. 1. IEEE, pp. I–I.

[42] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[43] M. Hong, Z.-Q. Luo, and M. Razaviyayn. “Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems”. In: *SIAM Journal on Optimization* 26.1 (2016), pp. 337–364.

[44] R. Horst and H. Tuy. *Global optimization*. Springer, 1996.

[45] R. Horst and H. Tuy. *Global optimization: Deterministic approaches*. Springer Science & Business Media, 2013.

[46] P. J. Huber et al. “Robust estimation of a location parameter”. In: *The Annals of Mathematical Statistics* 35.1 (1964), pp. 73–101.

[47] M. Hurdal, P. Bowers, K. Stephenson, D. Sumners, K. Rehm, K. Shaper, and D. Rotenberg. “Quasiconformally flat mapping the human cerebellum”. In: *MICCAI*. 1999.

[48] M. K. Hurdal and K. Stephenson. “Discrete conformal methods for cortical brain flattening”. In: *Neuroimage* 45.1 (2009), S86–S98.

[49] F. Kahl. “Multiple view geometry and the Linfty-norm”. In: *Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1*. Vol. 2. IEEE. 2005, pp. 1002–1009.

[50] F. Kahl and R. Hartley. “Multiple-View Geometry Under the L infinity-Norm”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.9 (2008), pp. 1603–1617.

[51] Q. Ke and T. Kanade. “Quasiconvex optimization for robust geometric reconstruction”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.10 (2007), pp. 1834–1847.

[52] P. Koehl and J. Hass. “Automatic alignment of genus-zero surfaces”. In: *IEEE TPAMI* 36.3 (2014), pp. 466–478.

[53] N. Krivulin. “An analysis of the Least Median of Squares regression problem”. In: *Computational Statistics*. Springer, 1992, pp. 471–476.

[54] L. J. Latecki, R. Lakämper, and U. Eckhardt. “Shape descriptors for non-rigid shapes with a single closed contour”. In: *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*. Vol. 1. IEEE. 2000, pp. 424–429.

[55] H. Le, T.-J. Chin, and D. Suter. “An Exact Penalty Method for Locally Convergent Maximum Consensus”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 1888–1896.

[56] K. Lebeda, J. Matas, and O. Chum. “Fixing the locally optimized ransac–full experimental evaluation”. In: *British Machine Vision Conference*. Citeseer. 2012, pp. 1–11.

[57] M. Leordeanu, M. Hebert, and R. Sukthankar. “An integer projected fixed point method for graph matching and map inference”. In: *Advances in neural information processing systems*. 2009, pp. 1114–1122.

[58] H. Li. “A practical algorithm for L triangulation with outliers”. In: *Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on*. IEEE. 2007, pp. 1–8.

[59] H. Li. “Consensus set maximization with guaranteed global optimality for robust geometry estimation”. In: *2009 IEEE 12th International Conference on Computer Vision*. IEEE. 2009, pp. 1074–1080.

[60] H. Li. “Efficient reduction of l-infinity geometry problems”. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE. 2009, pp. 2695–2702.

[61] Y. Lipman, R. Al-Aifari, and I. Daubechies. “The continuous Procrustes distance between two surfaces”. In: *Communication on Pure and Applied Mathematics* 66.6 (2011), pp. 934–964.

[62] Y. Lipman and T. Funkhouser. “Möbius voting for surface correspondence”. In: *ACM Transactions on Graphics (TOG)*. Vol. 28. 3. ACM. 2009, p. 72.

[63] Q. Liu, X. Shen, and Y. Gu. “Linearized ADMM for Non-convex Non-smooth Optimization with Convergence Analysis”. In: *arXiv preprint arXiv:1705.02502* (2017).

[64] D. G. Lowe. “Object recognition from local scale-invariant features”. In: *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Vol. 2. Ieee. 1999, pp. 1150–1157.

[65] H. Lu, L.-P. Nolte, and M. Reyes. “Interest points location for brain image using landmark-annotated atlas”. In: *Int’l J. Imaging Systems Technology* 22 (2012), pp. 145–152.

[66] O. Mangasarian. “Characterization of linear complementarity problems as linear programs”. In: *Complementarity and Fixed Point Problems*. Springer, 1978, pp. 74–87.

[67] J. Matoušek. “On geometric optimization with few violated constraints”. In: *Discrete & Computational Geometry* 14.4 (1995), pp. 365–384.

[68] D. Mumford, C. Series, and D. Wright. *Indra’s pearls: the vision of Felix Klein*. Cambridge University Press, 2002.

[69] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[70] C. F. Olson. “An approximation algorithm for least median of squares regression”. In: *Information Processing Letters* 63.5 (1997), pp. 237–241.

[71] C. Olsson, F. Kahl, and M. Oskarsson. “Branch-and-bound methods for Euclidean registration problems”. In: *IEEE TPAMI* 31.5 (2009), pp. 783–794.

[72] C. Olsson, O. Enqvist, and F. Kahl. “A polynomial-time bound for matching and registration with outliers”. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, pp. 1–8.

[73] C. Olsson, O. Enqvist, and F. Kahl. "A polynomial-time bound for matching and registration with outliers". In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, pp. 1–8.

[74] C. Olsson, A. P. Eriksson, and R. Hartley. "Outlier removal using duality". In: *IEEE Int. Conf. on Copmuter Vision and Pattern Recognition*. IEEE Computer Society. 2010, pp. 1450–1457.

[75] A. Parra Bustos, T.-J. Chin, and D. Suter. "Fast rotation search with stereographic projections for 3D registration". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 3930–3937.

[76] D. Pooley, M. Brooks, and A. van den Hengel. "RATSAC: An adaptive method for accelerated robust estimation and its application to video synchronisation". In: *Digital Image Computing Techniques and Applications, 9th Biennial Conference of the Australian Pattern Recognition Society on*. IEEE. 2007, pp. 294–300.

[77] P. J. Rousseeuw. "Least median of squares regression". In: *Journal of the American statistical association* 79.388 (1984), pp. 871–880.

[78] M. Sharir and E. Welzl. "A combinatorial bound for linear programming and related problems". In: *STACS 92* (1992), pp. 567–579.

[79] K. Sim and R. Hartley. "Removing outliers using the Linfty norm". In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 1. IEEE. 2006, pp. 485–494.

[80] N. Snavely, S. M. Seitz, and R. Szeliski. "Photo tourism: exploring photo collections in 3D". In: *ACM transactions on graphics (TOG)*. Vol. 25. 3. ACM. 2006, pp. 835–846.

[81] B. Springborn, P. Schröder, and U. Pinkall. "Conformal equivalence of triangle meshes". In: *SIGGRAPH Asia*. 2008.

[82] K. Stephenson. *Introduction to circle packing: The theory of discrete analytic functions*. Cambridge University Press, 2005.

[83] C. V. Stewart. "Robust parameter estimation in computer vision". In: *SIAM review* 41.3 (1999), pp. 513–537.

[84] B. J. Tordoff and D. W. Murray. "Guided-MLESAC: Faster image transform estimation by using matching priors". In: *IEEE transactions on pattern analysis and machine intelligence* 27.10 (2005), pp. 1523–1535.

[85] P. H. Torr and A. Zisserman. "MLESAC: A new robust estimator with application to estimating image geometry". In: *Computer Vision and Image Understanding* 78.1 (2000), pp. 138–156.

[86] O. Van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. "A survey on shape correspondence". In: *Computer Graphics Forum* 30.6 (2011), pp. 1681–1707.

[87] A. Vedaldi and B. Fulkerson. “VLFeat: An open and portable library of computer vision algorithms”. In: *Proceedings of the 18th ACM international conference on Multimedia*. ACM. 2010, pp. 1469–1472.

[88] S. Wang, Y. Wang, M. Jin, X. D. Gu, and G. Samaras. “Conformal geometry and its applications on 3D shape matching, recognition, and stitching”. In: *IEEE TPAMI* 29.7 (2007), pp. 1209–1220.

[89] Y. Wang, W. Yin, and J. Zeng. “Global convergence of ADMM in nonconvex nonsmooth optimization”. In: *arXiv preprint arXiv:1511.06324* (2015).

[90] H. S. Wong, T.-J. Chin, J. Yu, and D. Suter. “Dynamic and hierarchical multi-structure geometric model fitting”. In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE. 2011, pp. 1044–1051.

[91] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato. “A 3D facial expression database for facial behavior research”. In: *Automatic face and gesture recognition, 2006. FGR 2006. 7th international conference on*. IEEE. 2006, pp. 211–216.

[92] W. Zeng and D. X. Gu. “Conformal geometric methods in computer vision”. In: *CEWIT*. 2011.

[93] W. Zeng, H. Li, L. Chen, J.-M. Morvan, and X. D. Gu. “An automatic 3D expression recognition framework based on sparse representation of conformal images”. In: *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*. IEEE. 2013, pp. 1–8.

[94] Y. Zeng, C. Wang, Y. Wang, X. Gu, D. Samaras, and N. Paragios. “Dense non-rigid surface registration using high-order graph matching”. In: *CVPR*. IEEE. 2010, pp. 382–389.

[95] Z. Zhang. “Determining the epipolar geometry and its uncertainty: A review”. In: *International journal of computer vision* 27.2 (1998), pp. 161–195.

[96] Y. Zheng, S. Sugimoto, and M. Okutomi. “Deterministically maximizing feasible subsystem for robust model fitting with unit norm constraint”. In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE. 2011, pp. 1825–1832.