Deterministic Approximate Methods for Maximum Consensus Robust Fitting

Huu Le, Tat-Jun Chin, Anders Eriksson, Thanh-Toan Do, and David Suter

Abstract—Maximum consensus estimation plays a critically important role in several robust fitting problems in computer vision. Currently, the most prevalent algorithms for consensus maximization draw from the class of randomized hypothesize-and-verify algorithms, which are cheap but can usually deliver only rough approximate solutions. On the other extreme, there are exact algorithms which are exhaustive search in nature and can be costly for practical-sized inputs. This paper fills the gap between the two extremes by proposing *deterministic* algorithms to *approximately* optimize the maximum consensus criterion. Our work begins by reformulating consensus maximization with linear complementarity constraints. Then, we develop two novel algorithms: one based on non-smooth penalty method with a Frank-Wolfe style optimization scheme, the other based on the Alternating Direction Method of Multipliers (ADMM). Both algorithms solve convex subproblems to efficiently perform the optimization. We demonstrate the capability of our algorithms to greatly improve a rough initial estimate, such as those obtained using least squares or a randomized algorithm. Compared to the exact algorithms, our approach is much more practical on realistic input sizes. Further, our approach is naturally applicable to estimation problems with geometric residuals. *Matlab code and demo program for our methods can be downloaded from https://goo.gl/FQcxpi.*

Index Terms—Maximum consensus, robust fitting, deterministic algorithm, approximate algorithm.

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 [1] and its variants [2], [3], [4], [5], [6]. 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 [7].

An important innovation is locally optimized RANSAC (LO-RANSAC) [2], [8]. 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 provide an absolute certainty whether the obtained result is a satisfactory approximation. Moreover, when data is contaminated with a high proportion of outliers, such randomized methods tend to be computationally expensive, because the probability of randomly picking a clean minimal subset decreases exponentially with the number of outliers. 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 [9], [10], [11], [12], [13]. 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 [10], [12], tree search [13], or enumeration [9], [11]. 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 sup-

- *H. Le is with Chalmers University of Technology, Sweden. Email: huul@chalmers.se.*
T.-J. Chin is with the school of Computer Science, The University of Adelaide, Australia.
A. Eriksson is with the School of Information Technology and Electrical Engineering, University of Queensland, Brisbane, Australia.
T.-T. Do is with the Department of Computer Science, University of Liverpool.
D. Suter is with the School of Science, Edith Cowan University, Australia.

Manuscript received; Revised

ported 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 [14], [15].

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.

1.1 Deterministic robust fitting

M-estimators [16] are an 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 [17], [18], 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. 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 [14], [15], [19].

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.

1.2 Road map

The paper is structured as follows:

- Sec. 2 defines the maximum consensus problem and characterizes the solution. It then describes the crucial reformulation with complementarity constraints.
- Sec. 3 describes the non-smooth penalty method.
- Sec. 4 describes the ADMM-based algorithm.
- Sec. 5 shows the applicability of our methods to estimation problems with quasiconvex geometric residuals.
- Sec. 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 [20], which proposed only the method based on non-smooth penalization. Sec. 6 of the present paper experimentally compares the new ADMM technique with the penalty method.

2 Problem definition

We develop our algorithms in the context of fitting linear models, before extending to models with geometric residuals in Sec. 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.,

$$\max_{\boldsymbol{\theta}\in\mathbb{R}^d} \quad \Psi(\boldsymbol{\theta}) \tag{1}$$

where the objective function

$$\Psi(\boldsymbol{\theta}) = \sum_{j=1}^{N} \mathbb{I}\left(|\mathbf{x}_j^T\boldsymbol{\theta} - y_j| \le \epsilon\right) \tag{2}$$

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 $|\mathbf{x}_j^T\boldsymbol{\theta} - y_j| \le \epsilon$ equivalently using the two linear constraints

$$\mathbf{x}_j^T\boldsymbol{\theta} - y_j \le \epsilon, \quad -\mathbf{x}_j^T\boldsymbol{\theta} + y_j \le \epsilon, \tag{3}$$

and collecting the data into the matrices

$$\begin{aligned} \mathbf{A} &= \left[\mathbf{x}_1, -\mathbf{x}_1, \dots, \mathbf{x}_N, -\mathbf{x}_N\right], \\ \mathbf{b} &= \left[\epsilon + y_1, \epsilon - y_1, \dots, \epsilon + y_N, \epsilon - y_N\right]^T, \end{aligned} \tag{4}$$

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

$$\Psi(\boldsymbol{\theta}) = \sum_{i=1}^{M} \mathbb{I}\left(\mathbf{a}_i^T\boldsymbol{\theta} - b_i \le 0\right), \tag{5}$$

where $\mathbf{a}_i$ is the i-th column of $\mathbf{A}$ and b_i is the i-th element of $\mathbf{b}$. Plugging (5) instead of (2) into (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 (5) as the definition of consensus.

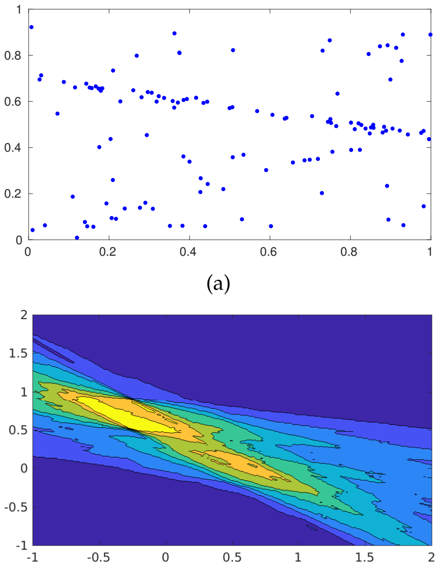
2.1 Characterizing the solution

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 (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. 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.

We define the *global* or *exact solution* to (1) as the vector $\boldsymbol{\theta}^*$ such that $\Psi(\boldsymbol{\theta}^*) \ge \Psi(\boldsymbol{\theta})$ 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

$$\max_{\boldsymbol{\theta}\in\mathbb{R}^d} \quad f(\boldsymbol{\theta}) \tag{6}$$

is a vector $\hat{\boldsymbol{\theta}}$ such that there exists a neighborhood $\mathcal{N} \subset \mathbb{R}^d$ of $\hat{\boldsymbol{\theta}}$ where $f(\hat{\boldsymbol{\theta}}) \ge f(\boldsymbol{\theta})$ for all $\boldsymbol{\theta} \in \mathcal{N}$ [21, Chap. 2]. By this definition, since Ψ is piece-wise constant, all $\boldsymbol{\theta} \in \mathbb{R}^d$ are local solutions to (1). The concept of *local optimality* is thus not



(a)

(b)

Fig. 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.

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 (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.

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 (1) equivalently as an outlier count minimization problem

$$\begin{aligned}
\min_{\mathbf{u}\in\{0,1\}^M,\, \mathbf{s}\in\mathbb{R}^M,\, \boldsymbol{\theta}\in\mathbb{R}^d} \quad & \sum_i u_i && \text{(7a)}\\
\text{subject to} \quad & s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i \geq 0, && \text{(7b)}\\
& u_i(s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i) = 0, && \text{(7c)}\\
& s_i(1-u_i) = 0, && \text{(7d)}\\
& s_i \geq 0. && \text{(7e)}
\end{aligned}$$

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 (7d). In turn, (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 (7c) and (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 (7d), the indicator variables can be relaxed without impacting the optimum, leading to the equivalent problem

$$\begin{aligned}
\min_{\mathbf{u},\mathbf{s}\in\mathbb{R}^M,\, \boldsymbol{\theta}\in\mathbb{R}^d} \quad & \sum_i u_i && \text{(8a)}\\
\text{subject to} \quad & s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i \geq 0, && \text{(8b)}\\
& u_i(s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i) = 0, && \text{(8c)}\\
& s_i(1-u_i) = 0, && \text{(8d)}\\
& 1 - u_i \geq 0, && \text{(8e)}\\
& s_i, u_i \geq 0. && \text{(8f)}
\end{aligned}$$

This, however, does not make (8) tractable, since (8c) and (8d) are bilinear in the unknowns.

To re-express (8) using only positive variables, define

$$\mathbf{v} = \begin{bmatrix} \boldsymbol{\theta} + \gamma\mathbf{1} \\ \gamma \end{bmatrix}, \qquad \mathbf{c}_i = \begin{bmatrix} \mathbf{a}_i^T & -\mathbf{a}_i^T\mathbf{1} \end{bmatrix}^T, \tag{9}$$

where both are real vectors of length $(d+1)$. Problem (8) can then be reformulated equivalently as

$$\begin{aligned}
\min_{\mathbf{u},\mathbf{s}\in\mathbb{R}^M,\, \mathbf{v}\in\mathbb{R}^{d+1}} \quad & \sum_i u_i \\
\text{subject to} \quad & s_i - \mathbf{c}_i^T\mathbf{v} + b_i \geq 0, \\
& u_i(s_i - \mathbf{c}_i^T\mathbf{v} + b_i) = 0, \\
& s_i(1-u_i) = 0, \\
& 1 - u_i \geq 0, \\
& s_i, u_i, v_i \geq 0.
\end{aligned} \tag{10}$$

Given a solution $\hat{\mathbf{u}}$, $\hat{\mathbf{s}}$ and $\hat{\mathbf{v}}$ to (10), the corresponding solution $\hat{\boldsymbol{\theta}}$ to (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 (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.

3 Non-smooth penalty method

Our first deterministic refinement algorithm is based on the technique of non-smooth penalization [21, Sec. 17.2]. Incorporating the equality constraints in (10) into the cost function as a penalty term, we obtain the penalty problem

$$\begin{aligned}
\min_{\mathbf{u},\mathbf{s}\in\mathbb{R}^M,\mathbf{v}\in\mathbb{R}^{d+1}} \quad & \sum_i u_i + \alpha\left[u_i(s_i - \mathbf{c}_i^T\mathbf{v} + b_i) + s_i(1-u_i)\right] \\
\text{s.t.} \quad & s_i - \mathbf{c}_i^T\mathbf{v} + b_i \geq 0, \\
& 1 - u_i \geq 0, \\
& s_i, u_i, v_i \geq 0.
\end{aligned} \tag{11}$$

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 (11) define a convex domain.

Henceforth, to reduce clutter, we sometimes use

$$\mathbf{z} = \begin{bmatrix} \mathbf{u}^T & \mathbf{s}^T & \mathbf{v}^T \end{bmatrix}^T. \tag{12}$$

The cost function in (11) can be rewritten as

$$P(\mathbf{z} \mid \alpha) = F(\mathbf{z}) + \alpha Q(\mathbf{z}), \tag{13}$$

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

$$\begin{aligned} Q(\mathbf{z}) &= \sum_i u_i(s_i - \mathbf{c}_i^T\mathbf{v} + b_i) + s_i(1 - u_i) & (14)\\ &= \sum_i s_i - u_i(\mathbf{c}_i^T\mathbf{v} - b_i). & (15)\end{aligned}$$

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

$$\mathbf{r}(\mathbf{z}) = \begin{bmatrix} r_1(\mathbf{z}) & \dots & r_M(\mathbf{z}) \end{bmatrix}^T, \tag{16}$$

where

$$r_i(\mathbf{z}) = s_i - u_i(\mathbf{c}_i^T\mathbf{v} - b_i). \tag{17}$$

In Sec. 3.2, we will devise a consensus maximization algorithm based on solving a sequence of the penalty problem (11) with increasing values of α. Before that, in Sec. 3.1, we will discuss a method to solve the penalty problem for a given (constant) α.

3.1 Solving the penalty problem

3.1.1 Necessary optimality conditions

Although $P(\mathbf{z} \mid \alpha)$ is quadratic, problem (11) is non-convex. However, it can be shown that (11) has a vertex solution, i.e., a solution that is an extreme point of the convex set

$$\begin{aligned} \mathcal{P} = \{\mathbf{z} \in \mathbb{R}^{2M+d+1} \,|\, & s_i - \mathbf{c}_i^T\mathbf{v} + b_i \geq 0,\\ & 1 - u_i \geq 0,\\ & s_i, u_i, v_i \geq 0,\\ & i = 1, \dots, M\} \end{aligned} \tag{18}$$

To minimize clutter, rewrite

$$\mathcal{P} = \{\mathbf{z} \in \mathbb{R}^{2M+d+1} \mid \mathbf{M}\mathbf{z} + \mathbf{q} \geq \mathbf{0}, \;\; \mathbf{z} \geq \mathbf{0}\}, \tag{19}$$

where

$$\begin{aligned} \mathbf{M} &= \begin{bmatrix} \mathbf{0} & \mathbf{I} & -\mathbf{C} \\ -\mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix},\\ \mathbf{C} &= \begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \dots & \mathbf{c}_M \end{bmatrix}^T,\\ \mathbf{q} &= \begin{bmatrix} \mathbf{b}^T & \mathbf{1}^T \end{bmatrix}^T; \end{aligned} \tag{20}$$

(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 (11) obey the KKT conditions [21, Chap. 12]

$$\begin{aligned} &\mathbf{u}^T(-\alpha\mathbf{C}\mathbf{v} + \alpha\mathbf{b} + \mathbf{1} + \boldsymbol{\lambda}^{\mathcal{G}}) = 0,\\ &\mathbf{s}^T(\alpha\mathbf{1} - \boldsymbol{\lambda}^{\mathcal{H}}) = 0,\\ &\mathbf{v}^T(-\alpha\mathbf{C}^T\mathbf{u} + \mathbf{C}^T\boldsymbol{\lambda}^{\mathcal{H}}) = 0,\\ &(\boldsymbol{\lambda}^{\mathcal{H}})^T(\mathbf{s} - \mathbf{C}\mathbf{v} + \mathbf{b}) = 0,\\ &(\boldsymbol{\lambda}^{\mathcal{G}})^T(\mathbf{1} - \mathbf{u}) = 0,\\ &\mathbf{s} - \mathbf{C}\mathbf{v} + \mathbf{b} \geq \mathbf{0},\\ &\mathbf{1} - \mathbf{u} \geq \mathbf{0},\\ &\boldsymbol{\lambda}^{\mathcal{H}}, \boldsymbol{\lambda}^{\mathcal{G}}, \mathbf{u}, \mathbf{v}, \mathbf{s} \geq \mathbf{0}, \end{aligned} \tag{21}$$

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 (11); see the supplementary material (Section 2) for details.

By rearranging, the KKT conditions (21) can be summarized by the following relations

$$\mathbf{M}'\mathbf{z}' + \mathbf{q}' \geq \mathbf{0}, \;\; \mathbf{z}' \geq \mathbf{0}, \;\; (\mathbf{z}')^T(\mathbf{M}'\mathbf{z}' + \mathbf{q}') = 0, \tag{22}$$

where

$$\begin{aligned} \mathbf{z}' &= \begin{bmatrix} \mathbf{z}^T & (\boldsymbol{\lambda}^{\mathcal{H}})^T & (\boldsymbol{\lambda}^{\mathcal{G}})^T \end{bmatrix}^T,\\ \mathbf{M}' &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & -\alpha\mathbf{C} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ -\alpha\mathbf{C}^T & \mathbf{0} & \mathbf{0} & \mathbf{C}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & -\mathbf{C} & \mathbf{0} & \mathbf{0} \\ -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix},\\ \mathbf{q}' &= \begin{bmatrix} (\alpha\mathbf{b} + \mathbf{1})^T & \alpha\mathbf{1}^T & \mathbf{0}^T & \mathbf{b}^T & \mathbf{1}^T \end{bmatrix}^T. \end{aligned} \tag{23}$$

Finding a feasible $\mathbf{z}'$ for (22) is an instance of a *linear complementarity problem (LCP)* [22]. Define the convex set

$$\mathcal{P}' = \{\mathbf{z}' \in \mathbb{R}^{4M+d+1} \mid \mathbf{M}'\mathbf{z}' + \mathbf{q}' \geq \mathbf{0}, \;\; \mathbf{z}' \geq \mathbf{0}\}. \tag{24}$$

We invoke the following result from [22, Lemma 2].

Theorem 1. If the LCP defined by the constraints (22) has a solution, then it has a solution at a vertex of $\mathcal{P}'$.

Theorem 1 implies that the KKT points of (11) (including the solutions of the problem) occur at the vertices of $\mathcal{P}'$. This also implies that (11) has a vertex solution, viz.:

Theorem 2. For any vertex

$$\mathbf{z}'_v = [\mathbf{z}_v^T \;\; (\boldsymbol{\lambda}_v^{\mathcal{H}})^T \;\; (\boldsymbol{\lambda}_v^{\mathcal{G}})^T)]^T \tag{25}$$

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

$$\mathbf{M}'\mathbf{E}\mathbf{z}'_v + \mathbf{q}' - \boldsymbol{\gamma}' = \mathbf{0}, \tag{26}$$

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,

$$\mathbf{M}'_J\mathbf{E}\mathbf{z}'_v + \begin{bmatrix} \mathbf{b}^T & \mathbf{1}^T \end{bmatrix}^T - \boldsymbol{\gamma}'_J = \mathbf{0}, \tag{27}$$

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 (23)), then

$$\mathbf{M}'_J\mathbf{E}_K\mathbf{z}_v + \begin{bmatrix} \mathbf{b}^T & \mathbf{1}^T \end{bmatrix}^T - \boldsymbol{\gamma}'_J = \mathbf{0}, \tag{28}$$

where $\mathbf{E}_K$ is the first-$(2M + d + 1)$ columns of $\mathbf{E}$. Since $\mathbf{M}'_J\mathbf{E}_K = \mathbf{M}$, then (28) implies that $\mathbf{z}_v$ is a vertex of $\mathcal{P}$. ∎

3.1.2 Frank-Wolfe algorithm

Theorem 2 suggests an approach to solve (11) by searching for a vertex solution. Further, note that for a fixed $\mathbf{u}$, (11) reduces to an LP. Conversely, for fixed s and $\mathbf{v}$, (11) is also an LP. This advocates alternating between optimizing subsets of the variables using LPs. Algorithm 1 summarizes the method, which is in fact a special case of the Frank-Wolfe method [23] for non-convex quadratic minimization.

Algorithm 1 Frank-Wolfe method for (11).

```
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** 3.* In a finite number of steps, Algorithm 1 converges to a KKT point of (11).

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

$$\mathcal{P} = \mathcal{P}_1 \times \mathcal{P}_2, \tag{29}$$

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 (11). 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 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 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 1 alternates between residual minimization (LP1) and inlier-outlier dichotomization (LP2).

3.2 Main algorithm

Intuitively, if the penalty parameter α is small, Algorithm 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 (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 payoff, thus preserving the feasibility of $\mathbf{z}$— Section 3.2.1 will formally establish this condition.

The above observations argue for a deterministic consensus maximization algorithm based on solving (11) for progressively larger α's; see Algorithm 2. For each α, our method solves (11) using Algorithm 1. The solution $\hat{\mathbf{z}}$ for a particular α is then used to initialize Algorithm 1 for the next larger α. The sequence terminates when the complementarity residual $Q(\mathbf{z})$ vanishes or becomes insignificant.

Algorithm 2 Non-smooth penalty method for solving (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. 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 (11) (Sec. 3.1.1) to enable efficient minimization.

3.2.1 Convergence

***Theorem** 4.* If α is sufficiently large, Algorithm 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 (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. ■

3.2.2 Initialization

Algorithm 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 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 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, these settings enable Algorithm 2 to find very good solutions at competitive runtimes.

4 ADMM-BASED ALGORITHM

Our second technique derives from the class of proximal splitting algorithms [24]. Specifically, we apply the ADMM to construct a deterministic approximate algorithm for our target problem (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 [25], [26]. While ADMM has recently found usage in several geometric vision problems, e.g., bundle adjustment [27], [28], triangulation [29], its application to robust fitting is relatively unexplored.

4.1 ADMM formulation

The specific version of ADMM used in our work is *consensus ADMM* [24], 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 (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

$$\mathbf{z}'_i = [u'_i\ s'_i\ (\mathbf{v}'_i)^T]^T \in \mathbb{R}^{d+3}, \tag{30}$$

as well as the "coupling" parameter vector

$$\mathbf{z}_C = [\mathbf{s}_C^T\ \mathbf{v}_C^T]^T \in \mathbb{R}^{M+d}. \tag{31}$$

Then, rewrite (10) as

$$\begin{aligned} \min_{\mathbf{z}, \{\mathbf{z}'_i\}, \mathbf{z}_C} \quad & \sum_i [u'_i + \mathbb{I}_B(\mathbf{z}'_i)] + \mathbb{I}_C(\mathbf{z}_C) && \text{(32a)} \\ \text{s.t.} \quad & \mathbf{u} = \mathbf{u}', && \text{(32b)} \\ & \mathbf{s} = \mathbf{s}' = \mathbf{s}_C, && \text{(32c)} \\ & \mathbf{v} = \mathbf{v}'_i = \mathbf{v}_C, && \text{(32d)} \end{aligned}$$

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

$$\mathbb{I}_B(\mathbf{z}'_i) = \begin{cases} 0 & \text{if } \begin{cases} u'_i(s'_i - \mathbf{c}_i^T \mathbf{v}'_i + b_i) = 0, \\ s'_i(1 - u'_i) = 0, \\ u'_i \in \{0, 1\}, \end{cases} \\ \infty & \text{otherwise,} \end{cases} \tag{33}$$

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

$$\mathbb{I}_C(\mathbf{z}_C) = \begin{cases} 0 & \text{if } \begin{cases} \mathbf{s}_C - \mathbf{C}\mathbf{v}_C + \mathbf{b} \geq \mathbf{0}, \\ \mathbf{s}_C, \mathbf{v}_C \geq \mathbf{0}, \end{cases} \\ \infty & \text{otherwise.} \end{cases} \tag{34}$$

Note that the objective function (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 (32b), (32c), and (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 (32) is identical to problem (10), in that solving (32) results in the same optimum as (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 (32) does not change if the term $\|\mathbf{u}\|^2$ is added to the cost function (32a). Thus, to aid the convergence of our proposed algorithm (refer to the supplementary material (Section 1) for more details), the solution of (32) can be obtained by solving the following problem:

$$\begin{aligned} \min_{\mathbf{z}, \{\mathbf{z}'_i\}, \mathbf{z}_C} \quad & \sum_i [u'_i + \mathbb{I}_B(\mathbf{z}'_i)] + \mathbb{I}_C(\mathbf{z}_C) + \|\mathbf{u}\|^2 && \text{(35a)} \\ \text{s.t.} \quad & \mathbf{u} = \mathbf{u}', && \text{(35b)} \\ & \mathbf{s} = \mathbf{s}' = \mathbf{s}_C, && \text{(35c)} \\ & \mathbf{v} = \mathbf{v}'_i = \mathbf{v}_C, && \text{(35d)} \end{aligned}$$

4.1.1 Augmented Lagrangian

Now consider the augmented Lagrangian of (35)

$$\begin{aligned} \mathcal{L}_\rho(\mathbf{z}, \{\mathbf{z}'_i\}, \mathbf{z}_C, \boldsymbol{\lambda}) = \; & \sum_i [u'_i + \mathbb{I}_B(\mathbf{z}'_i)] + \mathbb{I}_C(\mathbf{z}_C) + \|\mathbf{u}\|^2 \\ & + \rho(\|\mathbf{u}' - \mathbf{u} + \boldsymbol{\lambda}^\mathbf{u}\|_2^2 - \|\boldsymbol{\lambda}^\mathbf{u}\|_2^2) \\ & + \rho(\|\mathbf{s}' - \mathbf{s} + \boldsymbol{\lambda}^\mathbf{s}\|_2^2 - \|\boldsymbol{\lambda}^\mathbf{s}\|_2^2) \\ & + \rho(\|\mathbf{s}_C - \mathbf{s} + \boldsymbol{\lambda}^\mathbf{s}_C\|_2^2 - \|\boldsymbol{\lambda}^\mathbf{s}_C\|_2^2) \\ & + \rho(\|\mathbf{v}_C - \mathbf{v} + \boldsymbol{\lambda}^\mathbf{v}_C\|_2^2 - \|\boldsymbol{\lambda}^\mathbf{v}_C\|_2^2) \\ & + \rho \sum_i (\|\mathbf{v}'_i - \mathbf{v} + \boldsymbol{\lambda}^\mathbf{v}_i\|_2^2 - \|\boldsymbol{\lambda}^\mathbf{v}_i\|_2^2), \end{aligned} \tag{36}$$

where

$$\mathbf{u}' = \begin{bmatrix} u'_1 & \dots u'_M \end{bmatrix}^T, \;\; \mathbf{s}' = \begin{bmatrix} s'_1 & \dots s'_M \end{bmatrix}^T, \tag{37}$$

and ρ is the penalty parameter. The vector

$$\boldsymbol{\lambda} = [(\boldsymbol{\lambda}^\mathbf{u})^T\ (\boldsymbol{\lambda}^\mathbf{s})^T\ (\boldsymbol{\lambda}^\mathbf{s}_C)^T\ (\boldsymbol{\lambda}^\mathbf{v}_C)^T\ \{(\boldsymbol{\lambda}^\mathbf{v}_i)^T\}_{i=1}^M]^T \tag{38}$$

contains all the scaled dual variables associated with the constraints in (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 [24]. Sec. 4.3 will elaborate on the overall algorithm and the associated convergence guarantee. Next in Sec. 4.2 we will first examine in detail the individual update steps.

[1] Consensus ADMM is a version commonly used for distributed optimization [24]. For brevity, we do not explore distributed optimization in our work, though our algorithm is amenable to such a scheme.

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

$$\mathbf{z}_i' \leftarrow \underset{\mathbf{z}_i'}{\arg\min}\ \mathcal{L}_\rho(\mathbf{z}, \{\mathbf{z}_i'\}, \mathbf{z}_C, \boldsymbol{\lambda}),\ \forall i, \tag{39a}$$

$$\mathbf{z}_C \leftarrow \underset{\mathbf{z}_C}{\arg\min}\ \mathcal{L}_\rho(\mathbf{z}, \{\mathbf{z}_i'\}, \mathbf{z}_C, \boldsymbol{\lambda}), \tag{39b}$$

$$\mathbf{z} \leftarrow \underset{\mathbf{z}}{\arg\min}\ \mathcal{L}_\rho(\mathbf{z}, \{\mathbf{z}_i'\}, \mathbf{z}_C, \boldsymbol{\lambda}), \tag{39c}$$

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

$$\begin{aligned}
\boldsymbol{\lambda}^{\mathbf{u}} &\leftarrow \boldsymbol{\lambda}^{\mathbf{u}} + \mathbf{u}' - \mathbf{u},\\
\boldsymbol{\lambda}^{\mathbf{s}} &\leftarrow \boldsymbol{\lambda}^{\mathbf{s}} + \mathbf{s}' - \mathbf{s},\\
\boldsymbol{\lambda}^{\mathbf{s}}_C &\leftarrow \boldsymbol{\lambda}^{\mathbf{s}}_C + \mathbf{s}_C - \mathbf{s},\\
\boldsymbol{\lambda}^{\mathbf{v}}_C &\leftarrow \boldsymbol{\lambda}^{\mathbf{v}}_C + \mathbf{v}_C - \mathbf{v},\\
\boldsymbol{\lambda}^{\mathbf{v}}_i &\leftarrow \boldsymbol{\lambda}^{\mathbf{v}}_i + \mathbf{v}_i' - \mathbf{v},\ \forall i.
\end{aligned} \tag{40}$$

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 [24].

In the following, we take a deeper look into the subproblems in (39).

4.2.1 Updating $\mathbf{z}_i'$

Due to the decomposable nature of the augmented Lagrangian (36), the problem in (39a) can be reduced to

$$\begin{aligned}
\underset{\mathbf{z}_i'}{\arg\min}\quad & u_i' + \rho(u_i' - u_i + \lambda_i^{\mathbf{u}})^2\\
& + \rho(s_i' - s_i + \lambda_i^{\mathbf{s}})^2 + \rho\|\mathbf{v}_i' - \mathbf{v} + \boldsymbol{\lambda}_i^{\mathbf{v}}\|_2^2 && (41a)\\
\text{s.t.}\quad & u_i'(s_i' - \mathbf{c}_i^T\mathbf{v}_i' + b_i) = 0, && (41b)\\
& s_i'(1 - u_i') = 0, && (41c)\\
& u_i' \in \{0, 1\}. && (41d)
\end{aligned}$$

where terms not affected by $\mathbf{z}_i'$ have also been ignored. Due to the complementarity constraints (41b) and (41c), and the binary restriction (41d) on u_i' , (41) 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 $\mathbf{v} - \boldsymbol{\lambda}_i^{\mathbf{v}}$ to minimize (41a).
- $u_i' = 1$: To satisfy (41b), s_i' must be equal to $\mathbf{c}_i^T\mathbf{v}_i' - b_i$ Then problem (41) becomes the unconstrained convex quadratic program (QP)

$$\min_{\mathbf{v}_i'}\ (\mathbf{c}_i^T\mathbf{v}_i' - b_i - s_i + \lambda_i^{\mathbf{s}})^2 + \|\mathbf{v}_i' - \mathbf{v} + \boldsymbol{\lambda}_i^{\mathbf{v}}\|_2^2. \tag{42}$$

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 (41). Note that the value of ρ would affect the chosen $\mathbf{z}_i'$.

4.2.2 Updating $\mathbf{z}_C$

Ignoring terms unrelated to $\mathbf{z}_C$, the problem in (39b) can be re-expressed as a convex QP

$$\begin{aligned}
\min_{\mathbf{z}_C}\quad & \|\mathbf{s}_C - \mathbf{s} + \boldsymbol{\lambda}_C^{\mathbf{s}}\|_2^2 + \|\mathbf{v}_C - \mathbf{v} + \boldsymbol{\lambda}_C^{\mathbf{v}}\|_2^2,\\
\text{s.t.}\quad & \mathbf{s}_C - \mathbf{C}\mathbf{v}_C + \mathbf{b} \geq \mathbf{0},\\
& \mathbf{s}_C, \mathbf{v}_C \geq \mathbf{0},
\end{aligned} \tag{43}$$

which can be solved efficiently up to global optimality.

4.2.3 Updating $\mathbf{z}$

Again ignoring terms unrelated to the variables of interest, the problem in (39c) reduces to

$$\begin{aligned}
\underset{\mathbf{z}}{\arg\min}\quad & \rho(\|\mathbf{u}' - \mathbf{u} + \boldsymbol{\lambda}^{\mathbf{u}}\|_2^2 + \|\mathbf{s}' - \mathbf{s} + \boldsymbol{\lambda}^{\mathbf{s}}\|_2^2\\
& + \|\mathbf{s}_C - \mathbf{s} + \boldsymbol{\lambda}_C^{\mathbf{s}}\|_2^2 + \|\mathbf{v}_C - \mathbf{v} + \boldsymbol{\lambda}_C^{\mathbf{v}}\|_2^2\\
& + \sum_i \|\mathbf{v}_i' - \mathbf{v} + \boldsymbol{\lambda}_i^{\mathbf{v}}\|_2^2) + \|\mathbf{u}\|^2.
\end{aligned} \tag{44}$$

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"

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

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

4.3 Main algorithm

Similar to the non-smooth penalty algorithm discussed in Sec. 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 3 summarizes the overall procedure.

4.3.1 Convergence

Theorem 5. For a sufficiently large ρ, the ADMM update iterations in (39) converge to a stationary point of the augmented Lagrangian (36), which is also a feasible solution of (10), after a finite number of steps.

Proof The detailed proof for this theorem can be found in the supplementary material (Section 1). For completeness, an outline of the proof is provided in this section.

Consider the $(t + 1)$-th update cycle of Algorithm. 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.

During the update steps of $\{\mathbf{z}_i\}$ and $\mathbf{z}_C$, since (41) and (43) can be solved optimally, it follows that:

$$\mathcal{L}_\rho(\mathbf{z}, \{\mathbf{z}_i'\}, \mathbf{z}_C, \boldsymbol{\lambda}) \geq \mathcal{L}_\rho(\mathbf{z}, \{\mathbf{z}_i'\}^+, \mathbf{z}_C^+, \boldsymbol{\lambda}) \tag{45}$$

Then, after $\mathbf{z}$ and $\boldsymbol{\lambda}$ are updated, with a sufficiently large ρ, it can be proven that:

$$\mathcal{L}_\rho(\mathbf{z}, \{\mathbf{z}_i'\}^+, \mathbf{z}_C^+, \boldsymbol{\lambda}) \geq \mathcal{L}_\rho(\mathbf{z}^+, \{\mathbf{z}_i'\}^+, \mathbf{z}_C^+, \boldsymbol{\lambda}^+) \tag{46}$$

Algorithm 3 ADMM-based method for solving (10).

```
Require: Data $\{\mathbf{c}_i, b_i\}_{i=1}^{M}$, initial model parameter $\boldsymbol{\theta}$, initial penalty value $\rho$, increment rate $\sigma$, threshold $\delta$.
$t \leftarrow 0$
$\mathbf{v}^{(t)} \leftarrow \left[(\boldsymbol{\theta} + |\min_j(\theta_j)|\mathbf{1})^T \;\; |\min_j(\theta_j)|\right]^T$.
$\mathbf{u}^{(t)} \leftarrow \mathbb{I}(\mathbf{Cv} - \mathbf{b} > 0)$.
$\mathbf{s}^{(t)} \leftarrow \mathbf{u} \odot (\mathbf{Cv} - \mathbf{b})$.
$\mathbf{z_i}^{(t)} = \mathbf{z}^{(t)}$ ; $\mathbf{z}_C{}^{(t)} = [\mathbf{s}^{(t)}; \mathbf{v}^{(t)}]$ ; $\boldsymbol{\lambda}^{(t)} = \mathbf{0}$
while true do
    $t \leftarrow t + 1$
    Update $\mathbf{z}_i{}^{(t)}$ by solving (41) $\forall\ i = 1..N$
    Update $\mathbf{z}_C{}^{(t)}$ by solving (43)
    Update $\mathbf{z}^{(t)}$ by solving (44)
    if $\|\mathbf{z}^{(t)} - \mathbf{z}^{(t-1)}\| \leq \delta$ then
        Break.
    end if
    $\rho^{(t)} \leftarrow \sigma \cdot \rho^{(t-1)}$.
end while
return $\mathbf{u}, \mathbf{s}, \mathbf{v}$.
```

(detailed proof is provided in the supplementary material – Section 1). From (45) and (46), the following inequality holds:

$$\mathcal{L}_\rho(\mathbf{z}, \{\mathbf{z}'_i\}, \mathbf{z}_C, \boldsymbol{\lambda}) \geq \mathcal{L}_\rho(\mathbf{z}^+, \{\mathbf{z}'_i\}^+, \mathbf{z}_C^+, \boldsymbol{\lambda}^+) \tag{47}$$

given that ρ is large enough.

The inequality (47) states that, with a sufficiently large ρ, the augmented Lagrangian (36) is monotonically non-increasing after every ADMM update cycle. As this function is bounded below with a sufficiently large ρ (detailed proof is given in the supplementary material–Section 1), its convergence to a point $\mathbf{z}^*$ is guaranteed. At convergence, all the constraints (32b), (32c) and (32d) are satisfied and $\mathbf{z}^*$ is also a feasible solution of (10). ■

4.3.2 Initialization

Similar to Alg. 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 that with this choice of the parameters, the algorithm was able to significantly improve the solution from an initial starting point.

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 [9], [14], [30], however, that many geometric residuals have the following *generalized fractional* form

$$\frac{\|\mathbf{G}\boldsymbol{\theta} + \mathbf{h}\|_p}{\mathbf{r}^T\boldsymbol{\theta} + q} \quad \text{with} \quad \mathbf{r}^T\boldsymbol{\theta} + q > 0, \tag{48}$$

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 (48). The associated maximum consensus problem is

$$\max_{\boldsymbol{\theta}\in\mathbb{R}^d} \quad \Psi(\boldsymbol{\theta}), \tag{49}$$

where

$$\Psi(\boldsymbol{\theta}) = \sum_{j=1}^{N} \mathbb{I}\left(\|\mathbf{G}_j\boldsymbol{\theta} + h_j\|_p \leq \epsilon(\mathbf{r}_j^T\boldsymbol{\theta} + q_j)\right). \tag{50}$$

In (50), we have moved the denominator of (48) to the RHS since ϵ is non-negative (see [14] for details). We show that for $p = 1$, our method can be easily adapted to solve maximum consensus for geometric residuals (49)[2]. Define

$$\mathbf{G}_j = \begin{bmatrix} \mathbf{g}_{j,1}^T \\ \mathbf{g}_{j,2}^T \end{bmatrix} \quad \mathbf{h}_j = \begin{bmatrix} h_{j,1} \\ h_{j,2} \end{bmatrix}. \tag{51}$$

Now, for $p = 1$, the constraint in (49) becomes

$$\left|\mathbf{g}_{j,1}^T\boldsymbol{\theta} + h_{j,1}\right| + \left|\mathbf{g}_{j,2}^T\boldsymbol{\theta} + h_{j,2}\right| \leq \epsilon(\mathbf{r}_j^T\boldsymbol{\theta} + q_j), \tag{52}$$

which in turn can be equivalently implemented using four linear constraints (see [30] for details). We can then manipulate (50) into the form (5), and the rest of our theory and algorithms will be immediately applicable.

6 Results

We tested our method (Algorithm 2 and Algorithm 3, henceforth abbreviated as EP and AM, respectively) on common parameter estimation problems. We compared EP and AM against the following well-known methods:

- RANSAC (RS) [1]: 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) [2]: 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) [8]: Following [8], 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) [31]: This method is equivalent to introducing slack variables to problem (2) and minimizing the ℓ_1-norm of the slack variables to yield an approximate solution to maximum consensus.
- ℓ_∞ outlier removal (l_∞) [32]: Again, in the context of (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 fundamental matrix estimation and linearized homography, we also compare our methods with Cov-RANSAC (CRS) [33], in which the uncertainties of the measurements and the homography matrix are incorporated to improve RANSAC.

[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.

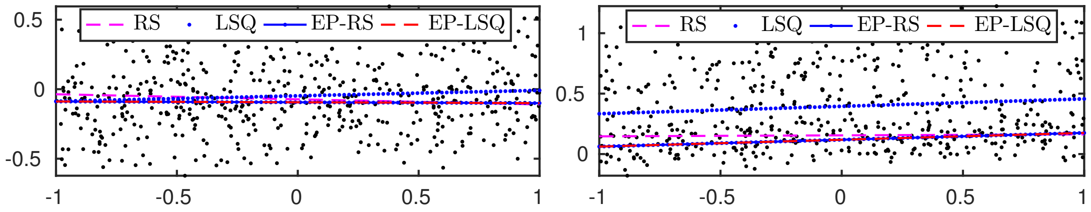
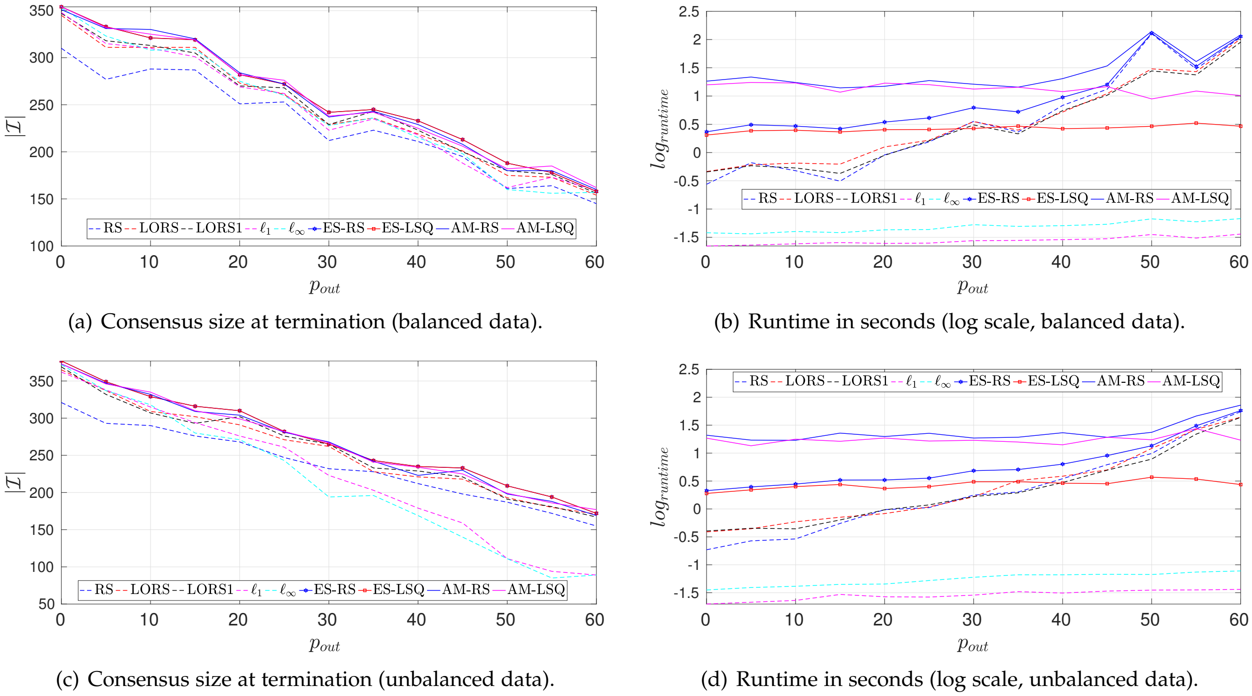


Fig. 2. Two-dimensional analogy of balanced (left) and unbalanced (right) data generated in our experiments. The results of RANSAC, least squares, and our method initialized with the former two methods are shown. Observe that least squares is heavily biased under unbalanced data, but EP is able to recover 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).

Fig. 3. Results for linear regression ($d = 8$ dimensions). (a)(b) Balanced data; (c)(d) Unbalanced data.

- For the experiments with image data where key-point matching scores are available as inlier priors, we executed two state-of-the-art RANSAC variants: PROSAC (PS) [3] and Guided MLESAC (GMLE) [4].

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, AM, ℓ_1 and ℓ_∞, Gurobi was employed as the LP and QP solver.

6.1 Linear models

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. 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 AM were initialized respectively with RANSAC (variants EP-RS and AM-RS) and least squares (variants EP-LSQ and AM-LSQ). For EP variants, the initial α was set to 0.5 and $\kappa = 5$, while initial ρ of AM variants was set to 0.1 and $\sigma = 2.5$ for all the runs.

Fig. 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, AM-RS, EP-LSQ and AM-LSQ, respectively. It is clear that, in terms of solution quality, the variants of EP and AM 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}). AM variants

TABLE 1
Fundamental matrix estimation results (with algebraic error)

Methods		RS	PS	GMLE	LORS	CRS	ℓ_1	ℓ_∞	EP-RS	EP-LSQ	EP-ℓ_∞	AM-RS	AM-LSQ	AM-ℓ_∞
House	$\lvert\mathcal{I}\rvert$	240	245	252	265	267	115	175	**275**	**275**	**275**	**275**	267	**275**
N = 556	time (s)	1.33	1.07	1.01	0.99	0.75	0.2	0.1	2.05	1.75	2.32	6.35	7.13	6.15
Aerial	$\lvert\mathcal{I}\rvert$	264	265	260	280	287	213	221	290	290	290	295	295	**300**
N = 483	time (s)	0.53	0.46	0.55	0.35	0.37	0.13	0.15	1.15	0.95	1.13	4.75	6.25	7.12
Merton	$\lvert\mathcal{I}\rvert$	295	295	301	306	306	147	227	**321**	**321**	**321**	307	305	302
N = 590	time (s)	0.65	0.25	0.30	0.25	0.30	0.25	0.13	1.18	0.95	1.05	5.15	5.73	5.78
Wadham	$\lvert\mathcal{I}\rvert$	305	307	315	320	325	271	290	**330**	**330**	**330**	310	**330**	315
N = 618	time (s)	1.52	1.35	1.15	1.05	1.08	0.15	0.27	2.25	1.41	1.42	8.88	7.51	6.52
Corridor	$\lvert\mathcal{I}\rvert$	310	310	315	327	330	251	300	375	**390**	**390**	388	375	**390**
N = 684	time (s)	0.95	1.12	0.97	0.65	0.75	0.15	0.27	2.35	1.17	1.26	6.52	5.56	7.0
Building 81	$\lvert\mathcal{I}\rvert$	421	429	417	444	439	443	441	**450**	**450**	**450**	**450**	449	441
N = 1041	time (s)	6.72	7.85	6.35	6.19	5.92	1.18	1.15	9.75	3.52	2.25	12.32	5.69	5.45
Building 04	$\lvert\mathcal{I}\rvert$	91	102	105	131	135	57	141	156	156	156	**177**	175	175
N = 1116	time (s)	12.95	10.71	11.70	9.30	3.50	1.15	0.02	17.15	6.15	7.15	15.67	10.89	11.2
Building 23	$\lvert\mathcal{I}\rvert$	120	116	110	223	225	70	110	**270**	**270**	**270**	265	267	261
N = 1163	time (s)	12.85	9.97	9.77	9.51	9.95	1.32	1.19	19.56	8.65	8.59	17.97	15.85	11.02
Building 36	$\lvert\mathcal{I}\rvert$	241	255	239	252	249	150	353	**380**	**380**	**380**	375	375	358
N = 923	time (s)	13.64	10.24	9.87	10.07	10.15	1.15	1.52	17.56	8.65	7.56	11.15	12.46	10.71

TABLE 2
Homography estimation results (with algebraic error)

Methods		RS	PS	GMLE	LORS	CRS	ℓ_1	ℓ_∞	EP-RS	EP-LSQ	EP-ℓ_∞	AM-RS	AM-LSQ	AM-ℓ_∞
University Library	$\lvert\mathcal{I}\rvert$	220	221	215	230	229	157	191	**295**	**295**	**295**	280	290	**295**
N = 439	time (s)	1.15	1.27	1.05	1.02	0.97	0.15	0.25	2.79	1.09	0.97	9.19	14.25	7.81
Christ Church	$\lvert\mathcal{I}\rvert$	259	262	265	273	277	267	251	315	315	315	**317**	311	**315**
N = 524	time (s)	1.15	1.12	1.01	1.19	1.05	0.09	0.15	2.99	1.78	1.91	9.79	8.46	15.21
Kapel	$\lvert\mathcal{I}\rvert$	156	155	162	165	160	95	115	**210**	**210**	**210**	200	201	205
N = 449	time (s)	1.18	1.12	1.18	1.44	1.65	0.11	0.07	2.22	1.32	1.29	10.41	9.74	11.01
Invalides	$\lvert\mathcal{I}\rvert$	178	170	169	180	185	117	107	230	230	230	**231**	229	229
N = 558	time (s)	2.01	2.76	1.79	1.85	1.55	0.09	0.07	3.35	3.01	4.15	10.2	9.81	10.47
Union House	$\lvert\mathcal{I}\rvert$	221	225	227	220	230	185	210	**290**	**290**	**290**	**290**	**290**	287
N = 520	time (s)	1.16	1.16	1.05	1.09	1.08	0.07	0.05	2.4	1.43	1.23	7.41	8.23	8.85
Old Classic Wing	$\lvert\mathcal{I}\rvert$	206	206	211	215	214	181	187	**250**	**250**	**250**	229	**250**	**250**
N = 561	time (s)	1.95	1.86	1.88	1.15	1.10	0.07	0.07	2.19	1.14	1.27	6.36	3.35	5.51
Ball Hall	$\lvert\mathcal{I}\rvert$	170	177	175	188	182	110	187	**215**	**215**	**215**	209	202	200
N = 538	time (s)	1.85	1.77	1.16	1.53	1.43	0.04	0.06	3.39	2.27	2.78	9.64	7.47	10.74
Building 64	$\lvert\mathcal{I}\rvert$	185	187	184	190	197	100	112	**233**	**233**	**233**	216	211	215
N = 529	time (s)	1.75	1.56	1.22	1.56	0.99	0.09	0.05	2.86	1.49	2.01	6.44	8.61	6.87
Building 10	$\lvert\mathcal{I}\rvert$	210	215	217	222	227	191	178	250	250	250	**251**	250	250
N = 546	time (s)	0.09	0.12	0.1	0.31	0.43	0.06	0.05	4.14	4.08	4.15	8.56	8.1	8.92

were also able to obtain roughly the same quality as EP-based methods, albeit with longer runtime. This is explainable as AM requires solving quaratic subproblems while only LPs are required for EP variants.

6.1.2 Fundamental matrix estimation (with algebraic error)

Following [34, 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 (48), thus linearization is essential to enable our method. Sec. 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 four image pairs 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 [35]) was used to extract putative correspondences. For images in the VGG dataset, we extract around 500 correspondences per pair. To simulate settings with high outlier rates, around 1000 putative correspondences are extracted per image pair in the Zurich dataset. An inlier threshold of $\epsilon = 1$ was used for all image pairs. For EP and AM, apart from initialization with RANSAC and least squares, we also initialised it with ℓ_∞ outlier removal (variants EP-ℓ_∞ and AM-ℓ_∞). For all EP variants, the initial α was set to 0.5 and $\kappa = 5$, while initial ρ for all AM variants was set to 0.1 and $\sigma = 2.5$ for all the runs.

Table 1 summarizes the quantitative results for all methods. Regardless of the initialization method, EP was able to find the largest consensus set. AM variants converge to approximately the same solution quality as EP while taking slightly longer runtime. Observe that for problem settings with high outlier rates (image pairs in the Zurich building dataset), EP-LSQ and EP-ℓ_∞ run much faster than all RANSAC variants, while consistently achieving better solution quality. Fig. 4 displays sample qualitative results for EP.

6.1.3 Homography estimation (with algebraic error)

Following [34, Chapter 4], the homography constraints were linearized to investigate the performance of our algorithms. Five image pairs form the VGG dataset: University Library,

[3] http://www.robots.ox.ac.uk/~vgg/data/
[4] http://www.vision.ee.ethz.ch/showroom/zubud/

TABLE 3
Homography estimation results (with geometric transfer error)

Methods		RS	PS	GMLE	LORS	LORS1	ℓ_1	ℓ_∞	EP-RS	EP-ℓ_∞	AM-RS	AM-ℓ_∞
University Library	$\lvert\mathcal{I}\rvert$	136	150	149	155	157	97	86	**210**	**210**	195	205
N = 439	time (s)	2.53	2.451	2.41	2.52	2.41	1.06	1.65	7.53	5.32	10.95	9.85
Christ Church	$\lvert\mathcal{I}\rvert$	125	127	130	125	129	101	120	**186**	**186**	175	**186**
N = 539	time (s)	2.79	2.52	2.5	2.44	2.53	1.35	2.09	8.95	6.93	16.82	18.16
Kapel	$\lvert\mathcal{I}\rvert$	160	167	160	160	157	110	104	**175**	**175**	169	168
N = 543	time (s)	2.84	2.11	3.87	2.31	2.68	2.7	2.07	7.44	9.32	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	7.92	5.51	12.33	11.44
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	2.84	3.59	7.73	7.35
Old Classic Wing	$\lvert\mathcal{I}\rvert$	198	208	126	209	210	52	147	**216**	206	210	197
N = 557	time (s)	1.85	1.47	2.57	3.32	3.96	2.77	1.47	5.29	7.57	9.06	10.23
Ball Hall	$\lvert\mathcal{I}\rvert$	225	227	221	227	230	195	186	**250**	**250**	247	247
N = 534	time (s)	1.35	1.37	1.29	1.33	1.34	0.57	1.05	3.47	2.95	6.45	7.35
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	6.95	7.54	10.07	9.05
Building 10	$\lvert\mathcal{I}\rvert$	201	225	210	215	226	176	165	**229**	**229**	226	210
N = 525	time (s)	1.48	1.48	0.95	1.46	1.38	1.14	1.71	6.66	7.59	12.56	9.48
Building 15	$\lvert\mathcal{I}\rvert$	215	217	221	225	232	240	245	**260**	**260**	**260**	**260**
N = 596	time (s)	1.94	1.82	1.78	1.65	1.67	1.62	1.17	5.39	4.56	9.31	11.08

TABLE 4
Affinity estimation results (with geometric transfer error)

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

Christ Church, Valbonne, Kapel and Paris's Invalides; three image pairs from the AdelaideRMF dataset [36]: 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 AM variants were reused from the fundamental matrix experiment. Quantitative results displayed in Table 2 show that all the EP and AM variants were able to achieve the highest consensus size.

6.2 Models with geometric distances

6.2.1 Homography estimation

We estimated 2D homographies based on the transfer error using all the methods. In the context of (48), the geometric residual for homography (with $p = 1$) is

$$\frac{\|(\boldsymbol{\theta}_{1:2} - \mathbf{v}_i\boldsymbol{\theta}_3)\tilde{\mathbf{u}_i}\|_1}{\boldsymbol{\theta}_3\tilde{\mathbf{u}_i}}, \tag{53}$$

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 AM 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 3, and a sample qualitative result for EP is shown in Fig. 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. AM variants also attain approximately the same solution as EP with slightly longer runtimes. Note that EP-LSQ and AM-LSQ were not invoked here, since finding least squares estimates based on geometric distances is intractable in general [19].

TABLE 5
Triangulation results (with geometric transfer error)

Methods		RS	LORS	LORS1	ℓ_1	ℓ_∞	EP-RS	EP-ℓ_∞	AM-RS	AM-ℓ_∞
Point 1	$\lvert\mathcal{I}\rvert$	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	$\lvert\mathcal{I}\rvert$	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	$\lvert\mathcal{I}\rvert$	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	$\lvert\mathcal{I}\rvert$	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	$\lvert\mathcal{I}\rvert$	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	$\lvert\mathcal{I}\rvert$	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	$\lvert\mathcal{I}\rvert$	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	$\lvert\mathcal{I}\rvert$	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	$\lvert\mathcal{I}\rvert$	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	$\lvert\mathcal{I}\rvert$	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

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:

$$\|\mathbf{u}_i - \boldsymbol{\theta}\tilde{\mathbf{v}}_i\|_1, \tag{54}$$

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 AM 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 4, and sample qualitative result is shown in Fig. 4. Similar conclusions can be drawn.

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 $\mathbf{P}_i \in \mathbb{R}^{3\times4}$, the following re-projection error with respect to the point estimation $\boldsymbol{\theta}$ was used in our experiments:

$$\frac{\|(\mathbf{P}^i_{1:2} - \mathbf{x}_i\mathbf{P}^i_3)\boldsymbol{\theta}'\|_1}{\mathbf{P}^i_3\tilde{\boldsymbol{\theta}}}, \tag{55}$$

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 [37] 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 AM variants, initial ρ was set to 0.1 and $\sigma = 2.5$. Table 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 ($d = 3$).

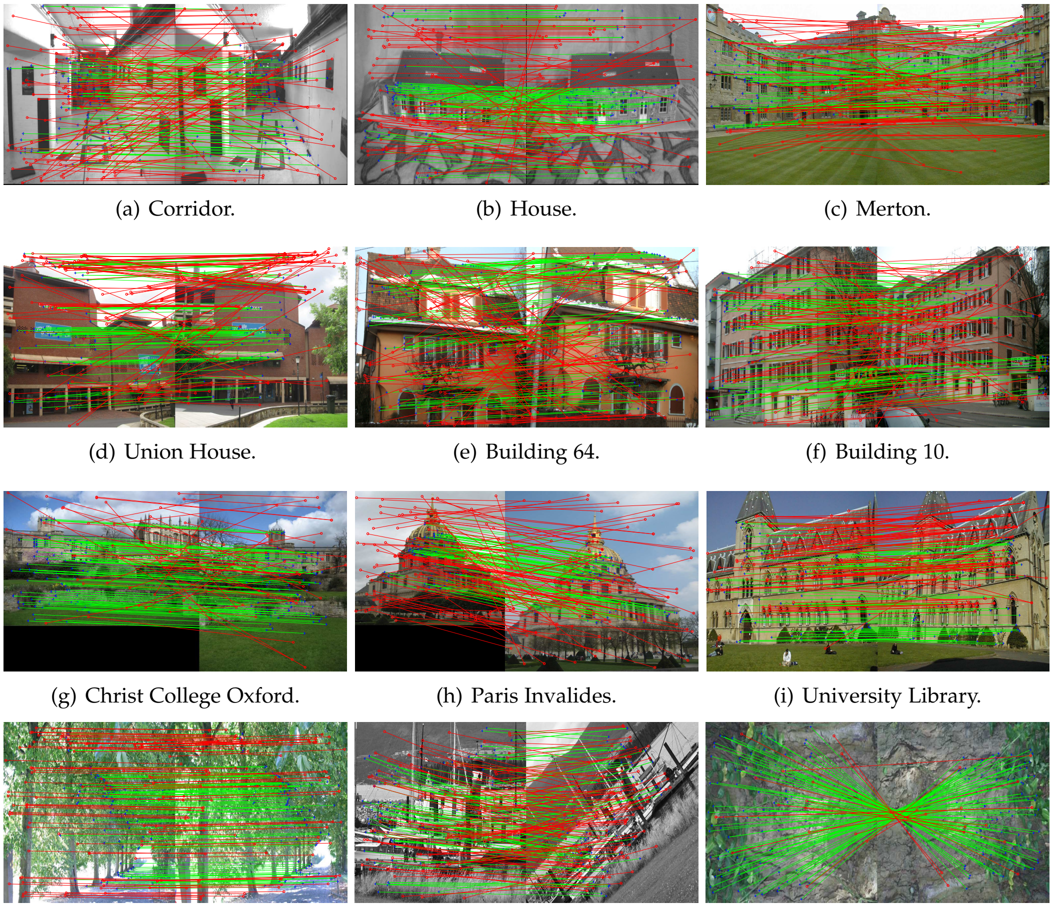
We repeated the experiments for all 11595 feature tracks in the dataset with more than 10 views. All the methods were executed with $\epsilon = 1$ pixel and the same set of parameters. Table 6 lists the total number of inliers and runtime for all the methods over all tested points. With RANSAC initialization, EP-RS was able to achieve the highest total number of inliers followed by AM-RS. The triangulated result is shown in Figure 5.

TABLE 6
Total inliers and runtime of triangulation for 11595 selected points with more than 10 views

Methods	Total inliers	Time (minutes)
RS	91888	12.10
LORS	94387	23.09
LORS1	91555	20.84
ℓ_1	40669	11.16
ℓ_∞	43869	45.18
EP-RS	**99232**	49.52
EP-ℓ_∞	59996	71.86
AM-RS	97453	86.14
AM-ℓ_∞	49760	125.74

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



(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.

Fig. 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.

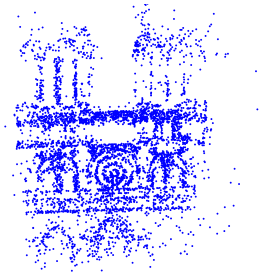


Fig. 5. Qualitative results of EP-RS on triangulation.

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.

Acknowledgments

Chin and Suter were supported by DP160103490. Eriksson was supported by FT170100072. The authors are also grateful to CE140100016 which funded this collaborative work.

References

[1] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[2] O. Chum, J. Matas, and J. Kittler, "Locally optimized ransac," in *DAGM*. Springer, 2003.

[3] 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.

[4] B. J. Tordoff and D. W. Murray, "Guided-mlesac: Faster image transform estimation by using matching priors," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 10, pp. 1523–1535, 2005.

[5] H. Le, T.-J. Chin, and D. Suter, "Ratsac-random tree sampling for maximum consensus estimation," in *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2017, pp. 1–8.

[6] H. M. Le, T.-T. Do, T. Hoang, and N.-M. Cheung, "Sdrsac: Semidefinite-based randomized approach for robust point cloud registration without correspondences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 124–133.

[7] S. Choi, T. Kim, and W. Yu, "Performance evaluation of RANSAC family," in *British Machine Vision Conference (BMVC)*, 2009.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] F. Kahl, “Multiple view geometry and the l-norm,” in *Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1*, vol. 2. IEEE, 2005, pp. 1002–1009.

[15] Q. Ke and T. Kanade, “Quasiconvex optimization for robust geometric reconstruction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 10, pp. 1834–1847, 2007.

[16] P. J. Huber *et al.*, “Robust estimation of a location parameter,” *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964.

[17] Z. Zhang, “Determining the epipolar geometry and its uncertainty: A review,” *International journal of computer vision*, vol. 27, no. 2, pp. 161–195, 1998.

[18] 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.

[19] R. Hartley and F. Kahl, “Optimal algorithms in multiview geometry,” in *Asian conference on computer vision*. Springer, 2007, pp. 13–34.

[20] H. Le, T.-J. Chin, and D. Suter, “An exact penalty method for locally convergent maximum consensus,” in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017.

[21] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[22] O. Mangasarian, “Characterization of linear complementarity problems as linear programs,” in *Complementarity and Fixed Point Problems*. Springer, 1978, pp. 74–87.

[23] M. Frank and P. Wolfe, “An algorithm for quadratic programming,” *Naval Research Logistics Quaterly*, vol. 3, no. 95, 1956.

[24] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[25] M. Hong, Z.-Q. Luo, and M. Razaviyayn, “Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems,” *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 337–364, 2016.

[26] Y. Wang, W. Yin, and J. Zeng, “Global convergence of admm in nonconvex nonsmooth optimization,” *Journal of Scientific Computing*, vol. 78, no. 1, pp. 29–63, 2019.

[27] 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.

[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-infty problems in multiview geometry,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 4066–4073.

[30] S. Agarwal, N. Snavely, and S. M. Seitz, “Fast algorithms for l problems in multiview geometry,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[31] 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.

[32] 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.

[33] R. Raguram, J.-M. Frahm, and M. Pollefeys, “Exploiting uncertainty in random sample consensus,” in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 2074–2081.

[34] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[35] 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.

[36] 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.

[37] N. Snavely, S. M. Seitz, and R. Szeliski, “Photo tourism: exploring photo collections in 3d,” in *ACM transactions on graphics (TOG)*, vol. 25, no. 3. ACM, 2006, pp. 835–846.

Huu Le received the B.S. degree in Electrical Engineering from Portland State University, Oregon, USA, in 2011. He obtained his PhD degree in Computer Science from University of Adelaide in 2018. He is currently a Postdoctoral Researcher at the Electrical Engineering Department, Chalmers University of Technology, Sweden. His research interests include robust estimation, computational geometry and numerical optimization methods applied to the fields of computer vision.

Tat-Jun Chin received the BEng degree in mechatronics engineering from Universiti Teknologi Malaysia (UTM) in 2003 and the PhD degree in computer systems engineering from Monash University, Victoria, Australia, in 2007. He was a research fellow at the Institute for Infocomm Research (I2R) in Singapore from 2007 to 2008. Since 2008, he has been at The University of Adelaide, South Australia, and is now an Associate Professor. He is an Associate Editor of IPSJ Transactions on Computer Vision and Applications (CVA), and Journal of Imaging. His research interests include robust estimation and geometric optimisation.

PLACE
PHOTO
HERE

Anders Eriksson received the M.Sc. degree in electrical engineering and the PhD degree in mathematics from Lund University, Sweden, in 2000 and 2008, respectively. Currently, he is an Associate Professor at University of Queensland, Australia. His research interests include optimization theory and numerical methods applied to the fields of computer vision and machine learning.

Thanh-Toan Do is currently a Lecturer at the Department of Computer Science, University of Liverpool (UoL), United Kingdom. He obtained a Ph.D. in Computer Science from INRIA, Rennes, France in 2012. Before joining UoL, he was a Research Fellow at the Singapore University of Technology and Design, Singapore (2013 - 2016) and the University of Adelaide, Australia (2016 - 2018). His research interests include Computer Vision and Machine Learning.

David Suter received the BSc degree in applied mathematics and physics from The Flinders University of South Australia in 1977, the Graduate Diploma in Computing from the Royal Melbourne Institute of Technology in 1984, and the PhD degree in computer science from La Trobe University in 1991. He was a lecturer at La Trobe from 1988 to 1991; and a senior lecturer in 1992, associate professor in 2001, and professor from 2006 to 2008 at Monash University, Melbourne, Australia. From 2008 to 2017, he was a professor at The University of Adelaide. Since 2018, he has been a Professor at Edith Cowan University, Australia. He served on the ARC College of Experts from 2008 to 2010. He has previously served on the editorial boards of the Machine Vision and Applications, the International Journal of Computer Vision, and the Journal of Mathematical Imaging and Vision. He was general co-chair of ACCV 2002 and ICIP 2013.