An Exact Penalty Method for Locally Convergent Maximum Consensus

Huu Le, Tat-Jun Chin and David Suter
School of Computer Science, The University of Adelaide
{huu.le, tat-jun.chin, david.suter}@adelaide.edu.au

Abstract

Maximum consensus estimation plays a critically important role in computer vision. Currently, the most prevalent approach draws from the class of non-deterministic hypothesize-and-verify algorithms, which are cheap but do not guarantee solution quality. On the other extreme, there are global algorithms which are exhaustive search in nature and can be costly for practical-sized inputs. This paper aims to fill the gap between the two extremes by proposing a locally convergent maximum consensus algorithm. Our method is based on a formulating the problem with linear complementarity constraints, then defining a penalized version which is provably equivalent to the original problem. Based on the penalty problem, we develop a Frank-Wolfe algorithm that can deterministically solve the maximum consensus problem. Compared to the randomized techniques, our method is deterministic and locally convergent; relative to the global algorithms, our method is much more practical on realistic input sizes. Further, our approach is naturally applicable to problems with geometric residuals[1].

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 [8] 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 [4].

An important innovation is locally optimized RANSAC (LO-RANSAC) [6, 15]. 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 [19, 26, 7, 16, 3]. 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 [26, 16], tree search [3], or exhaustive search [19, 7]. 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. We

[1] Code and demo are available in the supplementary material.

then develop a Frank-Wolfe algorithm that can deterministically and locally improve a given maximum consensus estimate. 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 nonlinear geometric residuals commonly used in computer vision [13, 14]. As will be demonstrated, our method can consistently improve upon a rough estimate (obtained using least squares, RANSAC, etc.) while incurring only marginally higher runtimes.

The above properties make our algorithm an excellent post processor for RANSAC and its variants, which are currently dominant in the field.

1.1. M-estimators and IRLS

More broadly in statistics, M-estimators [12] 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 [25, 1], 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 [13, 14, 10].

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.

2. Problem definition

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

$$\begin{aligned} \max_{\boldsymbol{\theta}\in\mathbb{R}^d,\, \mathcal{I}\in\mathcal{P}(N)} \quad & |\mathcal{I}| \\ \text{subject to} \quad & |\mathbf{x}_j^T\boldsymbol{\theta} - y_j| \leq \epsilon \qquad \forall j \in \mathcal{I}, \end{aligned} \tag{1}$$

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

$$\mathbf{x}_j^T\boldsymbol{\theta} - y_j \leq \epsilon, \quad -\mathbf{x}_j^T\boldsymbol{\theta} + y_j \leq \epsilon, \tag{2}$$

and collecting the data into the matrices

$$\mathbf{A} = \begin{bmatrix}\mathbf{x}_1, -\mathbf{x}_1, \ldots, \mathbf{x}_N, -\mathbf{x}_N\end{bmatrix}, \tag{3}$$

$$\mathbf{b} = \begin{bmatrix}\epsilon + y_1, \epsilon - y_1, \ldots, \epsilon + y_N, \epsilon - y_N\end{bmatrix}^T, \tag{4}$$

where $\mathbf{A} \in \mathbb{R}^{d\times M}$, $\mathbf{b} \in \mathbb{R}^M$ and $M = 2N$, we can rewrite (1) equivalently as

$$\begin{aligned} \max_{\boldsymbol{\theta}\in\mathbb{R}^d,\, \mathcal{I}\in\mathcal{P}(M)} \quad & |\mathcal{I}| \\ \text{subject to} \quad & \mathbf{a}_i^T\boldsymbol{\theta} - b_i \leq 0 \qquad \forall i \in \mathcal{I}, \end{aligned} \tag{5}$$

where $\mathbf{a}_i$ is the i-th column of $\mathbf{A}$ and b_i is the i-th element of $\mathbf{b}$. Problems (1) and (5) are equivalent in the sense that they have the same maximizer, though the maximum objective value of (5) is N plus the maximum objective value of (1) since for any $\boldsymbol{\theta}$, at least one of the constraints in (2) are satisfied. Henceforth, we will be developing our maximum consensus algorithm based on (5) as our target problem.

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) equivalently as an outlier count minimization problem

$$\min_{\mathbf{u}\in\{0,1\}^M,\, \mathbf{s}\in\mathbb{R}^M,\, \boldsymbol{\theta}\in\mathbb{R}^d} \quad \sum_i u_i \tag{6a}$$

$$\text{subject to} \quad s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i \geq 0, \tag{6b}$$

$$u_i(s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i) = 0, \tag{6c}$$

$$s_i(1 - u_i) = 0, \tag{6d}$$

$$s_i \geq 0. \tag{6e}$$

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 (6d). In turn, (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 (6c) and (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 (6d), the indicator variables can be relaxed without impacting the optimum, leading to the equivalent problem

$$\min_{\mathbf{u},\mathbf{s}\in\mathbb{R}^M,\, \boldsymbol{\theta}\in\mathbb{R}^d} \quad \sum_i u_i \tag{7a}$$

$$\text{subject to} \quad s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i \geq 0, \tag{7b}$$

$$u_i(s_i - \mathbf{a}_i^T\boldsymbol{\theta} + b_i) = 0, \tag{7c}$$

$$s_i(1 - u_i) = 0, \tag{7d}$$

$$1 - u_i \geq 0, \tag{7e}$$

$$s_i, u_i \geq 0. \tag{7f}$$

This, however, does not make (7) tractable to solve exactly, since (7c) and (7d) are bilinear in the unknowns.

To re-express (7) using only positive variables, define

$$\mathbf{v} = \begin{bmatrix} \boldsymbol{\theta} + \gamma\mathbf{1} \\ \gamma \end{bmatrix}, \quad \mathbf{c}_i = \begin{bmatrix} \mathbf{a}_i^T & -\mathbf{a}_i^T\mathbf{1} \end{bmatrix}^T, \tag{8}$$

where both are real vectors of length $(d+1)$. Problem (7) can then be reformulated equivalently as

$$\begin{aligned} \min_{\mathbf{u},\mathbf{s}\in\mathbb{R}^M,\ \mathbf{v}\in\mathbb{R}^{d+1}} \quad & \sum_i u_i \\ \text{subject to} \quad & s_i - \mathbf{c}_i^T\mathbf{v} + b_i \geq 0, \\ & u_i(s_i - \mathbf{c}_i^T\mathbf{v} + b_i) = 0, \\ & s_i(1-u_i) = 0, \\ & 1 - u_i \geq 0, \\ & s_i, u_i, v_i \geq 0. \end{aligned} \tag{9}$$

Given a solution $\hat{\mathbf{u}}$, $\hat{\mathbf{s}}$ and $\hat{\mathbf{v}}$ to (9), the corresponding solution $\hat{\boldsymbol{\theta}}$ to (7) can be obtained by simply subtracting the last element of $\hat{\mathbf{v}}$ from its first-d elements.

3. Penalty method

Incorporating the equality constraints in (9) into the cost function as a penalty term, we obtain the penalty problem

$$\begin{aligned} \min_{\mathbf{u},\mathbf{s},\mathbf{v}} \quad & \sum_i u_i + \alpha\left[u_i(s_i - \mathbf{c}_i^T\mathbf{v} + b_i) + s_i(1-u_i)\right] \\ \text{s.t.} \quad & s_i - \mathbf{c}_i^T\mathbf{v} + b_i \geq 0, \\ & 1 - u_i \geq 0, \\ & s_i, u_i, v_i \geq 0. \end{aligned} \tag{10}$$

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

$$\mathbf{z} = \begin{bmatrix} \mathbf{u}^T & \mathbf{s}^T & \mathbf{v}^T \end{bmatrix}^T. \tag{11}$$

In addition, the cost function in (10) is rewritten as

$$P(\mathbf{z} \mid \alpha) = F(\mathbf{z}) + \alpha Q(\mathbf{z}), \tag{12}$$

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

$$Q(\mathbf{z}) = \sum_i u_i(s_i - \mathbf{c}_i^T\mathbf{v} + b_i) + s_i(1-u_i) \tag{13}$$

$$= \sum_i s_i - u_i(\mathbf{c}_i^T\mathbf{v} - b_i). \tag{14}$$

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

In Sec. 3.3, we will investigate the conditions under which solving (10) is equivalent to solving (9), and devise an algorithm in Sec. 4 to exploit the equivalence for consensus maximization. First, in the next two subsections, we discuss solving the penalty problem (10) for a given α.

3.1. Necessary optimality conditions

Although $P(\mathbf{z} \mid \alpha)$ is quadratic, problem (10) is non-convex. However, it can be shown that (10) has a vertex solution, i.e., is an extreme point of the convex set

$$\mathcal{P} = \{\mathbf{z} \in \mathbb{R}^{2M+d+1} \mid \mathbf{M}\mathbf{z} + \mathbf{q} \geq \mathbf{0},\ \ \mathbf{z} \geq \mathbf{0}\}, \tag{15}$$

where

$$\begin{aligned} \mathbf{M} &= \begin{bmatrix} \mathbf{0} & \mathbf{I} & -\mathbf{C} \\ -\mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \\ \mathbf{C} &= \begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \dots & \mathbf{c}_M \end{bmatrix}^T, \\ \mathbf{q} &= \begin{bmatrix} \mathbf{b}^T & \mathbf{1}^T \end{bmatrix}^T; \end{aligned} \tag{16}$$

(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 (10) obey the KKT conditions [18, Chap. 12]

$$\begin{aligned} & \mathbf{u}^T(-\alpha\mathbf{C}\mathbf{v} + \alpha\mathbf{b} + \mathbf{1} + \boldsymbol{\lambda}^{\mathcal{G}}) = 0, \\ & \mathbf{s}^T(\alpha\mathbf{1} - \boldsymbol{\lambda}^{\mathcal{H}}) = 0, \\ & \mathbf{v}^T(-\alpha\mathbf{C}^T\mathbf{u} + \mathbf{C}^T\boldsymbol{\lambda}^{\mathcal{H}}) = 0, \\ & (\boldsymbol{\lambda}^{\mathcal{H}})^T(\mathbf{s} - \mathbf{C}\mathbf{v} + \mathbf{b}) = 0, \\ & (\boldsymbol{\lambda}^{\mathcal{G}})^T(\mathbf{1} - \mathbf{u}) = 0, \\ & \mathbf{s} - \mathbf{C}\mathbf{v} + \mathbf{b} \geq \mathbf{0}, \\ & \mathbf{1} - \mathbf{u} \geq \mathbf{0}, \\ & \boldsymbol{\lambda}^{\mathcal{H}}, \boldsymbol{\lambda}^{\mathcal{G}}, \mathbf{u}, \mathbf{v}, \mathbf{s} \geq \mathbf{0}, \end{aligned} \tag{17}$$

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 (10); see supplementary material for details.

By rearranging, the KKT conditions (17) can be summarized by the following relations

$$\mathbf{M}'\mathbf{z}' + \mathbf{q}' \geq \mathbf{0}, \ \ \mathbf{z}' \geq \mathbf{0}, \ \ (\mathbf{z}')^T(\mathbf{M}'\mathbf{z}' + \mathbf{q}') = 0, \tag{18}$$

where

$$\begin{aligned} \mathbf{z}' &= \begin{bmatrix} \mathbf{z}^T & (\boldsymbol{\lambda}^{\mathcal{H}})^T & (\boldsymbol{\lambda}^{\mathcal{G}})^T \end{bmatrix}^T, \\ \mathbf{M}' &= \begin{bmatrix} \mathbf{0} & \mathbf{0} & -\alpha\mathbf{C} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ -\alpha\mathbf{C}^T & \mathbf{0} & \mathbf{0} & \mathbf{C}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & -\mathbf{C} & \mathbf{0} & \mathbf{0} \\ -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \\ \mathbf{q}' &= \begin{bmatrix} (\alpha\mathbf{b} + \mathbf{1})^T & \alpha\mathbf{1}^T & \mathbf{0}^T & \mathbf{b}^T & \mathbf{1}^T \end{bmatrix}^T. \end{aligned} \tag{19}$$

Finding a feasible $\mathbf{z}'$ for (18) is an instance of a *linear complementarity problem (LCP)* [17]. Define the convex set

$$\mathcal{P}' = \{\mathbf{z}' \in \mathbb{R}^{4M+d+1} \mid \mathbf{M}'\mathbf{z}' + \mathbf{q}' \geq \mathbf{0},\ \ \mathbf{z}' \geq \mathbf{0}\}. \tag{20}$$

We invoke the following result from [17, Lemma 2].

Theorem 1 *If the LCP defined by the constraints* (18) *has a solution, then it has a solution at a vertex of* $\mathcal{P}'$.

Theorem 1 implies that the KKT points of (10) (including the solutions of the problem) occur at the vertices of $\mathcal{P}'$. This also implies that (10) has a vertex solution, viz.:

Theorem 2 *For any vertex*

$$\mathbf{z}'_v = [\mathbf{z}_v^T \;\; (\boldsymbol{\lambda}_v^{\mathcal{H}})^T \;\; (\boldsymbol{\lambda}_v^{\mathcal{G}})^T)]^T \tag{21}$$

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

$$\mathbf{M}'\mathbf{E}\mathbf{z}'_v + \mathbf{q}' - \boldsymbol{\gamma}' = \mathbf{0}, \tag{22}$$

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,

$$\mathbf{M}'_J\mathbf{E}\mathbf{z}'_v + \begin{bmatrix}\mathbf{b}^T & \mathbf{1}^T\end{bmatrix}^T - \boldsymbol{\gamma}'_J = \mathbf{0}, \tag{23}$$

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 (19)), then

$$\mathbf{M}'_J\mathbf{E}_K\mathbf{z}_v + \begin{bmatrix}\mathbf{b}^T & \mathbf{1}^T\end{bmatrix}^T - \boldsymbol{\gamma}'_J = \mathbf{0}, \tag{24}$$

where $\mathbf{E}_K$ is the first-$(2M + d + 1)$ columns of $\mathbf{E}$. Since $\mathbf{M}'_J\mathbf{E}_K = \mathbf{M}$, then (24) implies that $\mathbf{z}_v$ is a vertex of $\mathcal{P}$. ∎

3.2. Frank-Wolfe algorithm

Theorem 2 suggests an approach to solve (10) by searching for a vertex solution. Further, note that for a fixed $\mathbf{u}$, (10) reduces to an LP. Conversely, for fixed $\mathbf{s}$ and $\mathbf{v}$, (10) 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 [9] for non-convex quadratic minimization.

Theorem 3 *In a finite number of steps, Algorithm 1 converges to a KKT point of* (10).

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

$$\mathcal{P} = \mathcal{P}_1 \times \mathcal{P}_2, \tag{25}$$

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 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 (10). Since each LP must reduce or maintain $P(\mathbf{z} \mid \alpha)$ which is bounded below, the process terminates in finite steps. ∎

Algorithm 1 Frank-Wolfe method for (10).

Require: Data $\{\mathbf{c}_i, b_i\}_{i=1}^M$, penalty value α, initial solution $\mathbf{u}^{(0)}$, $\mathbf{v}^{(0)}$, $\mathbf{s}^{(0)}$, threshold δ.

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: $\quad t \leftarrow t + 1$.
5: $\quad \mathbf{s}^{(t)}, \mathbf{v}^{(t)} \leftarrow \arg\min_{\mathbf{s},\mathbf{v}} P(\mathbf{u}^{(t-1)}, \mathbf{s}, \mathbf{v} \mid \alpha)$ s.t. $\mathcal{P}$.
6: $\quad \mathbf{u}^{(t)} \leftarrow \arg\min_{\mathbf{u}} P(\mathbf{u}, \mathbf{s}^{(t)}, \mathbf{v}^{(t)} \mid \alpha)$ s.t. $\mathcal{P}$.
7: $\quad P^{(t)} \leftarrow P(\mathbf{u}^{(t)}, \mathbf{s}^{(t)}, \mathbf{v}^{(t)} \mid \alpha)$.
8: $\quad$ **if** $|P^{(t-1)} - P^{(t)}| \leq \delta$ **then**
9: $\quad\quad$ Break.
10: $\quad$ **end if**
11: **end while**
12: **return** $\mathbf{u}^{(t)}, \mathbf{v}^{(t)}, \mathbf{s}^{(t)}$.

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).

3.3. Exactness of penalization

The penalty problem (10) is an instance of a non-smooth exact penalty method [18, Sec. 17.2]. Observe that $Q(\mathbf{z})$ is the ℓ_1-norm of the LHS of the equality constraints in (9). The exactness of the penalization is exhibited in the following theorems (rephrased in the context of our problem).

Theorem 4 (based on Theorem 17.3 in [18]) *If* $\mathbf{z}^*$ *is a local solution of the original problem* (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 (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 (based on Theorem 17.4 in [18]) *Let $\hat{\mathbf{z}}$ be a KKT point of the penalized problem (10) for α greater than α^*. Then, $Q(\hat{\mathbf{z}}) = 0$, and $\hat{\mathbf{z}}$ is also a KKT point of (9).*

A "one shot" approach that sets α to a very large value and solves a single instance of (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.

4. Main algorithm

Based on the above results, we propose our main algorithm for solving the maximum consensus problem (9); see Algorithm 2. Our method solves (10) using Algorithm 1 for successively larger α, where the solution $\hat{\mathbf{z}}$ for a particular α is used to initialize Algorithm 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 (10), which we can provably achieve thanks to Theorem 3, Theorem 5 guarantees that Algorithm 2 will converge to a solution for (9) that satisfies the first-order necessary conditions for optimality.

Algorithm 2 Main algorithm for solving (9).

Require: Data $\{\mathbf{c}_i, b_i\}_{i=1}^M$, initial model parameter $\boldsymbol{\theta}$, initial penalty value α, increment rate κ, threshold δ.

$\mathbf{v} \leftarrow \left[(\boldsymbol{\theta} + |\min_j(\theta_j)|\mathbf{1})^T \;\; |\min_j(\theta_j)|\right]^T$.
$\mathbf{u} \leftarrow \mathbb{I}(\mathbf{C}\mathbf{v} - \mathbf{b} > 0)$.
$\mathbf{s} \leftarrow \mathbf{u} \odot (\mathbf{C}\mathbf{v} - \mathbf{b})$.
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 (10) (Sec. 3.1) to enable efficient minimization.

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

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 [13, 19, 2], however, that many geometric residuals have the following *generalized fractional* form

$$\frac{\|\mathbf{G}\boldsymbol{\theta} + \mathbf{h}\|_p}{\mathbf{r}^T\boldsymbol{\theta} + q} \quad \text{with} \quad \mathbf{r}^T\boldsymbol{\theta} + q > 0, \tag{26}$$

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 (26). The associated maximum consensus problem is

$$\begin{aligned} \max_{\boldsymbol{\theta}\in\mathbb{R}^d,\ \mathcal{I}\in\mathcal{P}(N)} \quad & |\mathcal{I}| \\ \text{subject to} \quad & \|\mathbf{G}_j\boldsymbol{\theta} + h_j\|_p \leq \epsilon(\mathbf{r}_j^T\boldsymbol{\theta} + q_j) \quad \forall j \in \mathcal{I}, \end{aligned} \tag{27}$$

where the denominator of (26) can be moved to the RHS since ϵ is non-negative (see [13] for details). We show that for $p = 1$, our method can be easily adapted to solve (27) up to local optimality[2]. Define

$$\mathbf{G}_j = \begin{bmatrix} \mathbf{g}_{j,1}^T \\ \mathbf{g}_{j,2}^T \end{bmatrix} \quad \mathbf{h}_j = \begin{bmatrix} h_{j,1} \\ h_{j,2} \end{bmatrix}. \tag{28}$$

Now, for $p = 1$, the constraint in (27) becomes

$$\left|\mathbf{g}_{j,1}^T\boldsymbol{\theta} + h_{j,1}\right| + \left|\mathbf{g}_{j,2}^T\boldsymbol{\theta} + h_{j,2}\right| \leq \epsilon(\mathbf{r}_j^T\boldsymbol{\theta} + q_j), \tag{29}$$

which in turn can be equivalently implemented using four linear constraints (see [2] for details). We can then manipulate (27) into the form (5), and the rest of our theory and algorithms will be immediately applicable.

5. Results

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

- RANSAC (RS) [8]: 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.

[2]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.

- LO-RANSAC (LORS) [6]: The maximum number of iterations for the inner sampling over the updated consensus set was set to 100.
- Improved LO-RANSAC (LORS1) [15]: 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) [20]: This method is equivalent to introducing slack variables to problem (5) and minimizing the ℓ_1-norm of the slack variables to yield an approximate solution to maximum consensus.
- ℓ_∞ outlier removal (l_∞) [21]: Again, in the context of (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) [5] and Guided MLESAC (GMLE) [23].

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.

5.1. 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. 1 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.

Fig. 2 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}).

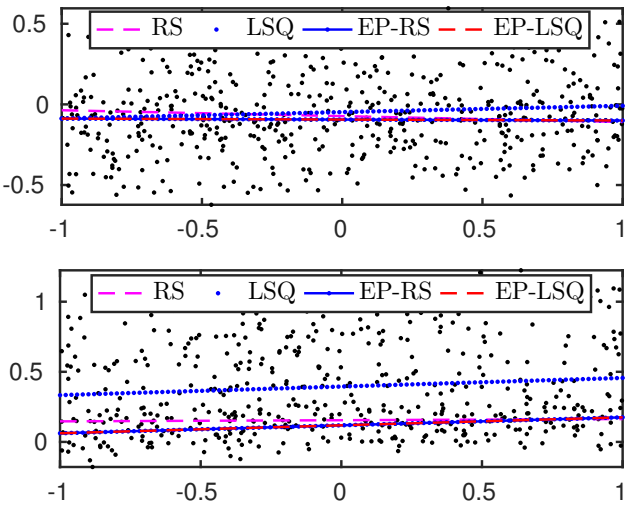
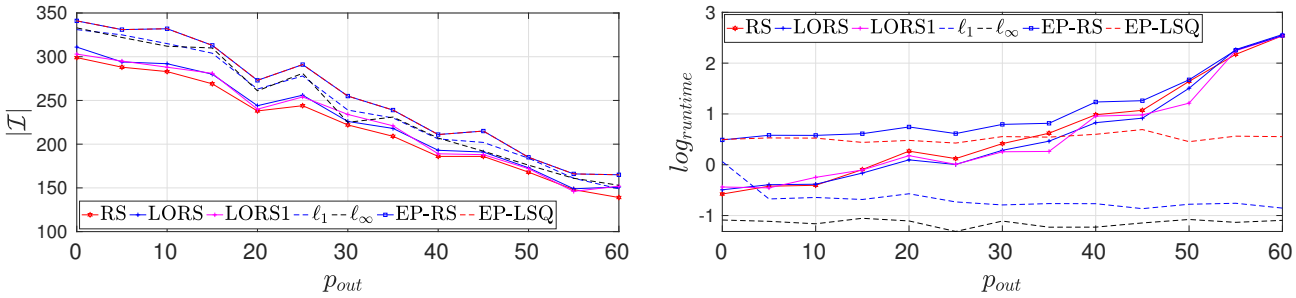


Figure 1. 2D analogy of balanced (top) and unbalanced (bottom) data generated in our experiments. The results of RANSAC, least squares, and our method initialized with the former two methods are shown. Observe that least squares is heavily biased under unbalanced data, but EP is able to recover from the bad initialization.

Fundamental matrix estimation Following [11, 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 (26), thus linearization is essential to enable our method. Sec. 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 [24]) 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 1 (top) 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. 3(a) displays a sample qualitative result for EP; for the qualitative results on the other image



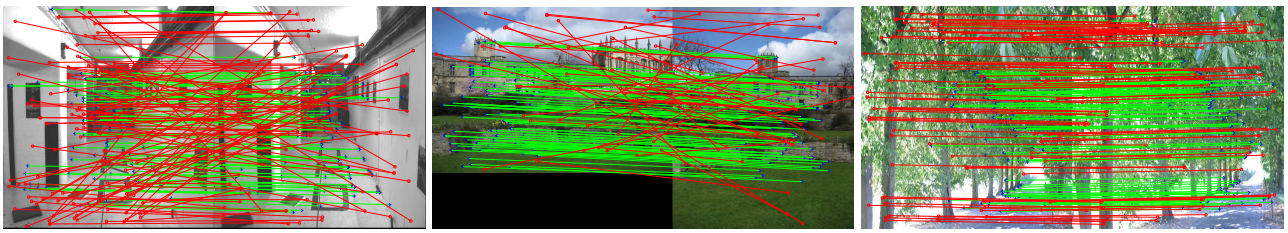
(a) Consensus size at termination (balanced data).

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

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

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

Figure 2. Results for linear regression ($d = 8$ dimensions). (a)(b) Balanced data; (c)(d) Unbalanced data.



(a) Corridor.

(b) Christ Church.

(c) Trees.

Figure 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.

pairs, please see the supplementary material.

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 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 1 (middle), and a sample qualitative result for EP is shown in Fig. 3(b). 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 [10].

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 1 (middle), and sample qualitative result is shown in Fig. 3(c). 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 [22] 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

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

	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

	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 1. (top) Fundamental matrix estimation results. (middle) Homography estimation and affinity estimation results. (bottom) 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.

$\kappa = 1.5$ for all variants of EP.

Table 1 (bottom) 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 ($d = 3$).

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.

Acknowledgements This work was supported by ARC grant DP160103490.

References

[1] 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*, pages 480–487. IEEE, 2015. 2

[2] 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*, pages 1–8. IEEE, 2008. 5

[3] 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*, pages 2413–2421, 2015. 1

[4] S. Choi, T. Kim, and W. Yu. Performance evaluation of RANSAC family. In *British Machine Vision Conference (BMVC)*, 2009. 1

[5] 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)*, volume 1, pages 220–226. IEEE, 2005. 6

[6] O. Chum, J. Matas, and J. Kittler. Locally optimized ransac. In *DAGM*. Springer, 2003. 1, 6

[7] O. Enqvist, E. Ask, F. Kahl, and K. Åström. Robust fitting for multiple view geometry. In *European Conference on Computer Vision*, pages 738–751. Springer, 2012. 1

[8] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 1, 5

[9] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quaterly*, 3(95), 1956. 4

[10] R. Hartley and F. Kahl. Optimal algorithms in multiview geometry. In *Asian conference on computer vision*, pages 13–34. Springer, 2007. 2, 7

[11] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 6

[12] P. J. Huber et al. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964. 2

[13] F. Kahl. Multiple view geometry and the l-norm. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1002–1009. IEEE, 2005. 2, 5

[14] Q. Ke and T. Kanade. Quasiconvex optimization for robust geometric reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1834–1847, 2007. 2

[15] K. Lebeda, J. Matas, and O. Chum. Fixing the locally optimized ransac–full experimental evaluation. In *British Machine Vision Conference*, pages 1–11. Citeseer, 2012. 1, 6

[16] H. Li. Consensus set maximization with guaranteed global optimality for robust geometry estimation. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1074–1080. IEEE, 2009. 1

[17] O. Mangasarian. Characterization of linear complementarity problems as linear programs. In *Complementarity and Fixed Point Problems*, pages 74–87. Springer, 1978. 3

[18] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006. 3, 4, 5

[19] 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*, pages 1–8. IEEE, 2008. 1, 5

[20] C. Olsson, A. P. Eriksson, and R. Hartley. Outlier removal using duality. In *IEEE Int. Conf. on Copmuter Vision and Pattern Recognition*, pages 1450–1457. IEEE Computer Society, 2010. 6

[21] 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)*, volume 1, pages 485–494. IEEE, 2006. 6

[22] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM transactions on graphics (TOG)*, volume 25, pages 835–846. ACM, 2006. 7

[23] 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*, 27(10):1523–1535, 2005. 6

[24] 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*, pages 1469–1472. ACM, 2010. 6

[25] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *International journal of computer vision*, 27(2):161–195, 1998. 2

[26] 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*, pages 1825–1832. IEEE, 2011. 1