Guaranteed Outlier Removal with Mixed Integer Linear Programs

Tat-Jun Chin*, Yang Heng Kee*, Anders Eriksson† and Frank Neumann*
*School of Computer Science, The University of Adelaide
†School of Electrical Engineering and Computer Science, Queensland University of Technology

Abstract

The maximum consensus problem is fundamentally important to robust geometric fitting in computer vision. Solving the problem exactly is computationally demanding, and the effort required increases rapidly with the problem size. Although randomized algorithms are much more efficient, the optimality of the solution is not guaranteed. Towards the goal of solving maximum consensus exactly, we present guaranteed outlier removal as a technique to reduce the runtime of exact algorithms. Specifically, before conducting global optimization, we attempt to remove data that are provably true outliers, i.e., those that do not exist in the maximum consensus set. We propose an algorithm based on mixed integer linear programming to perform the removal. The result of our algorithm is a smaller data instance that admits a much faster solution by subsequent exact algorithms, while yielding the same globally optimal result as the original problem. We demonstrate that overall speedups of up to 80% can be achieved on common vision problems[1].

1. Introduction

Given a set of data $\mathcal{X} = \{\mathbf{x}_i, y_i\}_{i=1}^N$, a frequently occurring problem in computer vision is to remove the outliers in $\mathcal{X}$. Often this is achieved by fitting a model, parametrized by $\boldsymbol{\theta} \in \mathbb{R}^L$, that has the largest consensus set $\mathcal{I}$ within $\mathcal{X}$

$$\begin{aligned} &\underset{\boldsymbol{\theta}, \mathcal{I} \subseteq \mathcal{X}}{\text{maximize}} \quad |\mathcal{I}| \\ &\text{subject to} \quad \left|\mathbf{x}_i^T \boldsymbol{\theta} - y_i\right| \leq \epsilon \quad \forall \{\mathbf{x}_i, y_i\} \in \mathcal{I}, \end{aligned} \tag{1}$$

where $\epsilon \geq 0$ is the inlier threshold. The solution $\mathcal{I}^*$ is the maximum consensus set, with consensus size $|\mathcal{I}^*|$. In this paper, we call $\mathcal{I}^*$ as *true inliers*, and $\mathcal{X} \setminus \mathcal{I}^*$ as *true outliers*, to indicate this segmentation as our target result.

Randomized algorithms such as RANSAC [10] and its variants are often used for outlier removal by approximately solving (1). Specifically, via a hypothesize-and-test procedure, RANSAC finds an approximate solution $\tilde{\mathcal{I}}$ to (1), where $|\tilde{\mathcal{I}}| \leq |\mathcal{I}^*|$, and the subset $\mathcal{X} \setminus \tilde{\mathcal{I}}$ are removed as outliers. Usually RANSAC-type algorithms do not provide optimality bounds, i.e., $\tilde{\mathcal{I}}$ can differ arbitrarily from $\mathcal{I}^*$. Also, in general $\tilde{\mathcal{I}} \not\subseteq \mathcal{I}^*$, hence, the consensus set $\tilde{\mathcal{I}}$ of RANSAC may contain true outliers, or conversely some of the data removed by RANSAC may be true inliers.

Solving (1) exactly, however, is non-trivial. Maximum consensus is an instance of the maximum feasible subsystem (MaxFS) problem [7, Chap. 7], which is intractable in general. Exact algorithms for (1) are brute force search in nature, whose runtime increases rapidly with N. The most well known is branch-and-bound (BnB) [4, 19, 15, 27], which can take exponential time in the worst case. For a fixed dimension L of $\boldsymbol{\theta}$, the problem can be solved in time proportional to an L-th order polynomial of N [11, 16, 9, 6]. However, this is only practical for small L.

In this paper, we present a *guaranteed outlier removal* (GORE) approach to speed up exact solutions of maximum consensus. Rather than attempting to solve (1) directly, our technique reduces $\mathcal{X}$ to a subset $\mathcal{X}'$, under the condition

$$\mathcal{I}^* \subseteq \mathcal{X}' \subseteq \mathcal{X}, \tag{2}$$

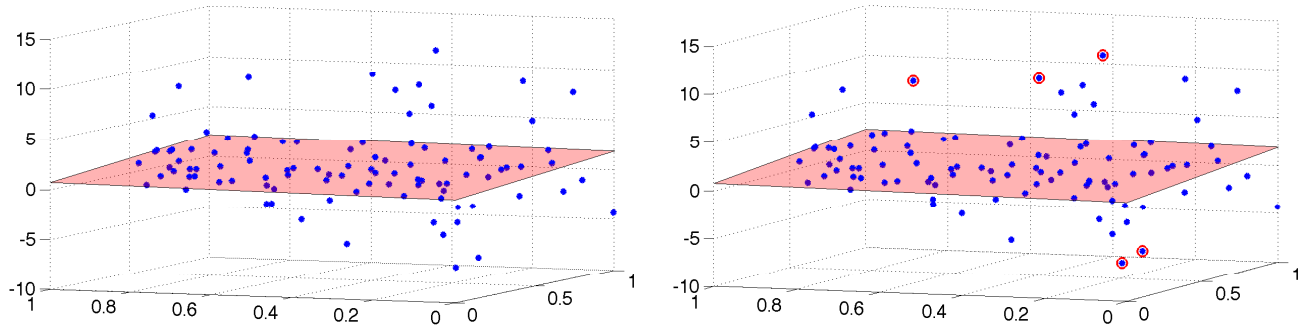
i.e., any data removed by our reduction of $\mathcal{X}$ are guaranteed to be true outliers. While $\mathcal{X}'$ may not be outlier-free, solving (1) exactly on $\mathcal{X}'$ will take less time, while yielding the same result as on the original input $\mathcal{X}$; see Fig. 1.

Note that naive heuristics used to reduce $\mathcal{X}$, e.g., by removing the most outlying data according to RANSAC, will not guarantee (2) and preserve global optimality. Instead, we propose a mixed integer linear program (MILP) algorithm to conduct GORE. For clarity our initial derivations will be based on the linear regression residual $|\mathbf{x}^T\boldsymbol{\theta} - y|$. In Sec. 3, we will extend our algorithm to handle geometric (non-linear regression) residuals [14].

We can expect that solving (1) on $\mathcal{X}'$ instead of $\mathcal{X}$ will be much faster, since any exact algorithm scales badly with problem size, and thus can be speeded up by even small reductions of $\mathcal{X}$. Of course, the time spent on conducting GORE must be included in the overall runtime. Intuitively, the effort required to remove *all* true outliers is unlikely to be less than that for solving (1) on $\mathcal{X}$ without any data re-

[1] Demo program is provided in the supplementary material.

1063-6919/16 $31.00 © 2016 IEEE
DOI 10.1109/CVPR.2016.631



(a) $|\mathcal{X}| = 100$, time to solution = $423.07s$ (global optimization) (b) $|\mathcal{X}'| = 95$, time to sol. = $50s$ (GORE) + $32.5s$ (glob. opt.) = $82.5s$

Figure 1. (a) Solving (1) exactly on $\mathcal{X}$ with $N = 100$ to robustly fit an affine plane ($d = 3$) using the Gurobi solver took $423.07s$. (b) Removing 5 true outliers (points circled in red) using the proposed GORE algorithm ($50s$) and subjecting the remaining data $\mathcal{X}'$ to Gurobi returns the same maximum consensus set in $32.5s$. This represents a reduction of 80% in the total runtime. Note that while removing such bad outliers using RANSAC may be easy, it does not guarantee that only true outliers are removed; indeed it may also remove true inliers.

moved. An immediate question is thus *how many true outliers must be removed for the proposed reduction scheme to pay off?* We will show that removing a small percentage of the most egregious true outliers using GORE is relatively cheap, and doing this is sufficient to speed up (1) substantially. As shown in Fig. 1, reducing $\mathcal{X}$ by a mere 5% using GORE decreases the *total* runtime required to find the globally optimal result by 80%. In Sec. 4, we will show similar performance gains on practical computer vision problems.

1.1. Related work

A priori removing some of the outliers in a highly contaminated dataset to speed up a subsequent exact algorithm is not a new idea. For example, in [15], the proposed BnB algorithm was suggested as a means for "post validating" the result of RANSAC. However, culling the data with a fast approximate algorithm somewhat defeats the purpose of finding an exact solution afterwards, since the obtained result is not guaranteed to be optimal w.r.t. the original input. In contrast, our approach does not suffer from this weakness, since GORE removes only true outliers.

Our work is an extension of two recent techniques for GORE [22, 5], respectively specialized for estimating 2D rigid transforms and 3D rotations (both are 3DOF problems). In both works, the authors exploit the underlying geometry of the target model to quickly prune keypoint matches that do not lie in the maximum consensus set - their methods, therefore, cannot be applied to models other than 2D rigid transform and 3D rotation. In contrast, our algorithm is substantially more flexible; first, the linear regression model we use in (1) is applicable to a larger range of problems. Second, as we will show in Sec. 3, our formulation can be easily extended to geometric (non-linear regression) residuals used in computer vision [14].

Underpinning the flexibility of our method is a MILP formulation, which "abstracts away" the calculation of bounds required for GORE. Further, our MILP formulation allows the utilization of efficient commercial off-the-shelf solvers (e.g., CPLEX, Gurobi) to handle up to 6DOF models.

Beyond RANSAC and its variants, there exist other approximate methods for outlier removal. The l_∞ approach [20, 26] recursively finds the l_∞ estimate and removes the data with the largest residual, until a consensus set is obtained. While this guarantees that at least one true outlier will be removed in each step, the resulting consensus set may be smaller than the optimal consensus size since true inliers may also be removed. The l_1 approach [17] seeks an estimate that minimizes the sum of the residuals that exceed ϵ. After arriving at a solution, data whose residual exceeds ϵ are removed as outliers. Again, there is no guarantee that only true outliers will be removed, and the resulting consensus set may not be the largest it can be.

2. MILP formulation for GORE

Problem (1) can be restated as removing as few data as possible to reach a consistent subset. Following [7, Chapter 7] (see also [27]), this can be written as

$$\underset{\boldsymbol{\theta},\mathbf{z}}{\text{minimize}} \quad \sum_i z_i \tag{3a}$$

$$\text{subject to} \quad \left|\mathbf{x}_i^T\boldsymbol{\theta} - y_i\right| \leq \epsilon + z_i M, \tag{3b}$$

$$z_i \in \{0, 1\}, \tag{3c}$$

where $\mathbf{z} = \{z_i\}_{i=1}^N$ are indicator variables, and M is a large positive constant. Recall that a constraint of the form $|\mathbf{a}^T\boldsymbol{\theta} + b| \leq h$ is equivalent to two linear constraints

$$\mathbf{a}^T\boldsymbol{\theta} + b \leq h \quad \text{and} \quad -\mathbf{a}^T\boldsymbol{\theta} - b \leq h, \tag{4}$$

thus, problem (3) is a MILP [8]. Intuitively, setting $z_i = 1$ amounts to discarding $\{\mathbf{x}_i, y_i\}$ as an outlier. Given the solution $\mathbf{z}^*$ to (3), the maximum consensus set is

$$\mathcal{I}^* = \{\mathbf{x}_i, y_i \mid z_i^* = 0\}. \tag{5}$$

The constant M must be big enough to ensure correct operation. Using a big M to "ignore" constraints is very common in optimization. See [18, 7] for guidelines on selecting M.

The idea of GORE begins by rewriting (3) equivalently as the following "nested" problem

$$\underset{k=1,\ldots,N}{\text{minimize}} \quad \beta^k, \tag{6}$$

where we define β^k as the optimal objective value of the following subproblem

$$\begin{aligned} \underset{\boldsymbol{\theta},\mathbf{z}}{\text{minimize}} \quad & \sum_{i\neq k} z_i && \text{(7a)} \\ \text{subject to} \quad & \left|\mathbf{x}_i^T\boldsymbol{\theta} - y_i\right| \leq \epsilon + z_i M, && \text{(7b)} \\ & z_i \in \{0,1\}, && \text{(7c)} \\ & \left|\mathbf{x}_k^T\boldsymbol{\theta} - y_k\right| \leq \epsilon. && \text{(7d)} \end{aligned}$$

In words, problem (7) seeks to remove as few data as possible to achieve a consistent subset, given that $\{\mathbf{x}_k, y_k\}$ cannot be removed. Note that (7) remains a MILP, and (6) is not any easier to solve than (1); its utility derives from showing how a bound on (7) allows to identify true outliers.

Let $(\hat{\theta}, \hat{\mathbf{z}})$ indicate a suboptimal solution to (3), and

$$\hat{u} = \|\hat{\mathbf{z}}\|_1 \geq \|\mathbf{z}^*\|_1 \tag{8}$$

be its value. Let α^k be a lower bound value to (7), i.e.,

$$\alpha^k \leq \beta^k. \tag{9}$$

Given $\hat{u}$ and α^k, we can perform a test according to the following lemma to decide whether $\{\mathbf{x}_k, y_k\}$ is a true outlier.

Lemma 1 *If $\alpha^k > \hat{u}$, then $\{\mathbf{x}_k, y_k\}$ is a true outlier.*

Proof The lemma can be established via contradiction. The k-th datum $\{\mathbf{x}_k, y_k\}$ is a true inlier if and only if

$$\beta^k = \|\mathbf{z}^*\|_1 \leq \hat{u}. \tag{10}$$

In other words, if $\{\mathbf{x}_k, y_k\}$ is a true inlier, insisting it to be an inlier in (7) does not change the fact that removing $\|\mathbf{z}^*\|_1$ data is sufficient to achieve consensus. However, if we are given that $\alpha^k > \hat{u}$, then from (9)

$$\hat{u} < \alpha^k \leq \beta^k \tag{11}$$

and the necessary and sufficient condition (10) cannot hold. Thus $\{\mathbf{x}_k, y_k\}$ must be a true outlier. ∎

The following result shows that there are $\{\mathbf{x}_k, y_k\}$ where the test above will never give an affirmative answer.

Lemma 2 *If $\hat{z}_k = 0$, then $\alpha^k \leq \hat{u}$.*

Proof If $\hat{z}_k = 0$, then, fixing $\{\mathbf{x}_k, y_k\}$ as an inlier, $(\hat{\theta}, \hat{\mathbf{z}})$ is also a suboptimal solution to (7). Thus, $\hat{u} \geq \beta^k \geq \alpha^k$, and the condition in Lemma 1 will never be met. ∎

Our method applies Lemma 1 iteratively to remove true outliers. Critical to the operation of GORE is the calculation of $\hat{u}$ and α^k. The former can be obtained from approximate algorithms to (1) or (3), e.g., RANSAC and its variants. Specifically, given a suboptimal solution $\tilde{\mathcal{I}}$, we compute the upper bound as

$$\hat{u} = N - |\tilde{\mathcal{I}}|. \tag{12}$$

The main difficulty lies in computing a tight lower bound α^k. In the following subsection (Sec. 2.1), we will describe our method for obtaining α^k.

Lemma 2, however, establishes that there are data (those with $\hat{z}_i = 0$) that cannot be removed by the test in Lemma 1. Our main GORE algorithm, to be described in Sec. 2.2, prioritizes the test for data with the largest errors with respect to the suboptimal parameter $\hat{\theta}$, i.e., those with $\hat{z}_i = 1$.

2.1. Lower bound calculation

The standard approach for lower bounding MILPs is via a linear program (LP) relaxation [8]. In the context of (7), α^k is obtained as the optimal value of the LP

$$\begin{aligned} \underset{\boldsymbol{\theta},\mathbf{z}}{\text{minimize}} \quad & \sum_{i\neq k} z_i && \text{(13a)} \\ \text{subject to} \quad & \left|\mathbf{x}_i^T\boldsymbol{\theta} - y_i\right| \leq \epsilon + z_i M, && \text{(13b)} \\ & z_i \in [0,1], \quad \text{/*continuous*/} && \text{(13c)} \\ & \left|\mathbf{x}_k^T\boldsymbol{\theta} - y_k\right| \leq \epsilon, && \text{(13d)} \end{aligned}$$

where the binary constraints (7c) are relaxed to become continuous. By the simple argument that $[0,1]^{N-1}$ is a superset of $\{0,1\}^{N-1}$, α^k cannot exceed β^k.

The lower bound obtained solely via (13) tends to be loose. Observe that since M is very large, each continuous z_i in (13) need only be turned on slightly to attain sufficient slack, i.e., the optimized $\mathbf{z}$ tends to be small and fractional, leading to a large gap between α^k and β^k.

To obtain a more useful lower bound, we leverage on existing BnB algorithms for solving MILPs [8]. In the context of solving (7), BnB maintains a pair of lower and upper bound values α^k and γ^k over time, where

$$\alpha^k \leq \beta^k \leq \gamma^k. \tag{14}$$

The lower bound α^k is progressively raised by solving (13) on recursive subdivisions of the parameter space. If the exact solution to (7) is desired, BnB is executed until $\alpha^k = \gamma^k$. For the purpose of GORE, however, we simply execute BnB until one of the following is satisfied

$$\alpha^k > \hat{u} \text{ (Condition 1)} \quad \text{or} \quad \gamma^k \leq \hat{u} \text{ (Condition 2)}, \tag{15}$$

or until the time budget is exhausted. Satisfying Condition 1 implies that $\{\mathbf{x}_k, y_k\}$ is a true outlier, while satisfying Condition 2 means that Condition 1 will never be met. Meeting

Algorithm 1 MILP-based GORE

Require: Data $\mathcal{X} = \{\mathbf{x}_i, y_i\}_{i=1}^N$, inlier threshold ϵ, number of rejection tests T, maximum duration per test c.

```
Run approx. algo. to obtain an upper bound û (12).
Order X increasingly based on residuals on approx. sol.
for k = N, N − 1, . . . , N − T + 1 do
    Run BnB to solve (7) on X until one of the following is satisfied:
    • α^k > û (Condition 1);
    • γ^k ≤ û (Condition 2);
    • c seconds have elapsed.
    if Condition 1 was satisfied then
        X ← X \ {x_k, y_k}
    end if
    if Condition 2 was satisfied then
        û ← γ^k
    end if
end for
return Reduced data X.
```

Condition 2 also indicates that a better suboptimal solution (one that involves identifying $\{\mathbf{x}_k, y_k\}$ as an inlier) has been discovered, thus $\hat{u}$ should be updated.

Many state-of-the-art MILP solvers such as CPLEX and Gurobi are based on BnB, which we use as a "black box". Hence, we do not provide further details of BnB here; the interested reader is referred to [8].

2.2. Main algorithm

Algorithm 1 summarizes our method. An approximate algorithm is used to obtain the upper bound $\hat{u}$, and to reorder $\mathcal{X}$ such that the data that are more likely to be true outliers are first tested for removal. In our experiments, we apply the BnB MILP solver of Gurobi to derive the lower bound α^k. The crucial parameters are T and c, where the former is the number of data to attempt to reject, and the latter is the maximum duration devoted to attempt to reject a particular datum. The total runtime of GORE is therefore $T \cdot c$. As we will show in Sec. 4, on many real-life data, setting T and c to small values (e.g., $T \approx 0.1N$, $c \in [5s, 15s]$) is sufficient to reduce $\mathcal{X}$ to a subset $\mathcal{X}'$ that significantly speeds up global solution.

3. GORE with geometric residuals

So far, our method has been derived based on the linear regression residual $|\mathbf{x}^T\boldsymbol{\theta} - y|$. Here, we generalize our method to handle geometric residuals, in particular, quasiconvex residuals which are functions of the form

$$\frac{\|\mathbf{A}\boldsymbol{\theta} + \mathbf{b}\|_p}{\mathbf{c}^T\boldsymbol{\theta} + d} \quad \text{with} \quad \mathbf{c}^T\boldsymbol{\theta} + d > 0, \tag{16}$$

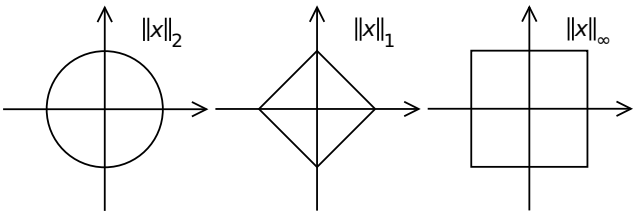


Figure 2. Unit circles for 3 different p-norms (adapted from [1]).

where $\boldsymbol{\theta} \in \mathbb{R}^L$ consists of the unknown parameters, and

$$\mathbf{A} \in \mathbb{R}^{2\times L},\ \mathbf{b} \in \mathbb{R}^2,\ \mathbf{c} \in \mathbb{R}^L,\ d \in \mathbb{R} \tag{17}$$

are constants. Note that (16) is quasiconvex for all $p \geq 1$. Many computer vision problems involve quasiconvex residuals, such as triangulation, homography estimation and camera resectioning; see [14] for details.

3.1. Maximum consensus

Thresholding (16) with ϵ and rearranging yields

$$\|\mathbf{A}\boldsymbol{\theta} + \mathbf{b}\|_p \leq \epsilon(\mathbf{c}^T\boldsymbol{\theta} + d), \tag{18}$$

where the condition $\mathbf{c}^T\boldsymbol{\theta} + d > 0$ is no longer required since it is implied above for any $\epsilon \geq 0$; see [14, Sec. 4].

In practice, actual measurements such as feature coordinates and camera matrices are used to derive the input data $\mathcal{X} = \{\mathbf{A}_i, \mathbf{b}_i, \mathbf{c}_i, d_i\}_{i=1}^N$. Given $\mathcal{X}$, the maximum consensus problem (3) can be extended for geometric residuals

$$\underset{\boldsymbol{\theta},\mathbf{z}}{\text{minimize}} \quad \sum_i z_i \tag{19a}$$

$$\text{subject to} \quad \|\mathbf{A}_i\boldsymbol{\theta} + \mathbf{b}_i\|_p \leq \epsilon(\mathbf{c}_i^T\boldsymbol{\theta} + d_i) + z_i M, \tag{19b}$$

$$z_i \in \{0, 1\}. \tag{19c}$$

Historically, quasiconvex residuals have been investigated for the case $p = 2$ [13, 14]. However, using the 2-norm makes (19b) a second order cone constraint, thus our MILP-based GORE is not immediately applicable.

Nonetheless, the 2-norm is not the only one possible for outlier removal. While the 2-norm is justified for i.i.d. Normal *inlier* residuals, for the purpose of separating inliers from *gross* outliers, most p-norms are effective. Fig. 2 shows unit circles for commonly used p-norms.

The general form of the p-norm is

$$\|\mathbf{x}\|_p = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{\frac{1}{p}}. \tag{20}$$

We show that for $p = 1$ and $p = \infty$, the thresholding constraint (19b) can be expressed using linear inequalities. Let

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{a}_{i,1}^T \\ \mathbf{a}_{i,2}^T \end{bmatrix}, \quad \mathbf{b}_i = \begin{bmatrix} b_{i,1} \\ b_{i,2} \end{bmatrix} \tag{21}$$

with $\mathbf{a}_{i,j}^T$ the j-th row of $\mathbf{A}_i$ and $b_{i,j}$ the j-th value of $\mathbf{b}_i$.

Using the 1-norm in (19b) and applying (20) yields

$$|\mathbf{a}_{i,1}^T\boldsymbol{\theta} + b_{i,1}| + |\mathbf{a}_{i,2}^T\boldsymbol{\theta} + b_{i,2}| \leq \epsilon(\mathbf{c}_i^T\boldsymbol{\theta} + d_i) + z_i M.$$

By recursively using the rule (4) to the absolute terms, the constraint above is equivalent to the four linear constraints

$$\begin{aligned}
(\mathbf{a}_{i,1}^T + \mathbf{a}_{i,2}^T)\boldsymbol{\theta} + b_{i,1} + b_{i,2} &\leq \epsilon(\mathbf{c}_i^T\boldsymbol{\theta} + d_i) + z_i M,\\
(\mathbf{a}_{i,2}^T - \mathbf{a}_{i,1}^T)\boldsymbol{\theta} + b_{i,2} - b_{i,1} &\leq \epsilon(\mathbf{c}_i^T\boldsymbol{\theta} + d_i) + z_i M,\\
(\mathbf{a}_{i,1}^T - \mathbf{a}_{i,2}^T)\boldsymbol{\theta} + b_{i,1} - b_{i,2} &\leq \epsilon(\mathbf{c}_i^T\boldsymbol{\theta} + d_i) + z_i M,\\
-(\mathbf{a}_{i,1}^T + \mathbf{a}_{i,2}^T)\boldsymbol{\theta} - b_{i,1} - b_{i,2} &\leq \epsilon(\mathbf{c}_i^T\boldsymbol{\theta} + d_i) + z_i M.
\end{aligned}$$

Note that the single indicator variable z_i "chains" the four constraints to the same datum $\{\mathbf{A}_i, \mathbf{b}_i, \mathbf{c}_i, d_i\}$.

For $p = \infty$, the p-norm reduces to

$$\|\mathbf{x}\|_\infty = \max\{|x_1|, |x_2|, \ldots, |x_n|\}. \tag{22}$$

Changing to the ∞-norm in (19b) thus yields

$$\max\{|\mathbf{a}_{i,1}^T\boldsymbol{\theta} + b_{i,1}|, |\mathbf{a}_{i,2}^T\boldsymbol{\theta} + b_{i,2}|\} \leq \epsilon(\mathbf{c}_i^T\boldsymbol{\theta} + d_i) + z_i M,$$

which is equivalent to simultaneously imposing

$$\begin{aligned}
|\mathbf{a}_{i,1}^T\boldsymbol{\theta} + b_{i,1}| &\leq \epsilon(\mathbf{c}_i^T\boldsymbol{\theta} + d_i) + z_i M,\\
|\mathbf{a}_{i,2}^T\boldsymbol{\theta} + b_{i,2}| &\leq \epsilon(\mathbf{c}_i^T\boldsymbol{\theta} + d_i) + z_i M.
\end{aligned}$$

Applying rule (4) again on each of the above yields

$$\begin{aligned}
\mathbf{a}_{i,1}^T\boldsymbol{\theta} + b_{i,1} &\leq \epsilon(\mathbf{c}_i^T\boldsymbol{\theta} + d_i) + z_i M,\\
-\mathbf{a}_{i,1}^T\boldsymbol{\theta} - b_{i,1} &\leq \epsilon(\mathbf{c}_i^T\boldsymbol{\theta} + d_i) + z_i M,\\
\mathbf{a}_{i,2}^T\boldsymbol{\theta} + b_{i,2} &\leq \epsilon(\mathbf{c}_i^T\boldsymbol{\theta} + d_i) + z_i M,\\
-\mathbf{a}_{i,2}^T\boldsymbol{\theta} - b_{i,2} &\leq \epsilon(\mathbf{c}_i^T\boldsymbol{\theta} + d_i) + z_i M.
\end{aligned}$$

Again, the single indicator variable z_i connects the four constraints to the i-th datum.

Therefore, by using the 1-norm or ∞-norm in the quasiconvex residual (16), the corresponding maximum consensus problem (19) can be solved as a MILP.

3.2. MILP formulation

For completeness, the subproblem analogous to (7) is

$$\begin{aligned}
\underset{\boldsymbol{\theta},\mathbf{z}}{\text{minimize}} \quad & \sum_{i\neq k} z_i && \text{(23a)}\\
\text{subject to} \quad & \|\mathbf{A}_i\boldsymbol{\theta} + \mathbf{b}_i\|_p \leq \epsilon(\mathbf{c}_i^T\boldsymbol{\theta} + d_i) + z_i M, && \text{(23b)}\\
& z_i \in \{0, 1\}, && \text{(23c)}\\
& \|\mathbf{A}_k\boldsymbol{\theta} + \mathbf{b}_k\|_p \leq \epsilon(\mathbf{c}_k^T\boldsymbol{\theta} + d_k). && \text{(23d)}
\end{aligned}$$

By using the 1-norm or ∞-norm in (23b) and (23d), and following our derivations in Sec. 3.1 to convert them to linear inequalities, a lower bound for (23) can be obtained using BnB solvers for MILP. Algorithm 1 can thus be directly applied to conduct GORE for quasiconvex problems.

4. Results

We performed experiments to test the efficacy of GORE in speeding up maximum consensus. In particular, we compared the runtime of exactly solving (1) on the input data $\mathcal{X}$ (we call this EXACT), with the total runtime of running GORE to reduce $\mathcal{X}$ to $\mathcal{X}'$ and exactly solving (1) on $\mathcal{X}'$ (we call this GORE+EXACT). To solve (1) exactly, we apply the industry-grade Gurobi Optimizer to exactly solve the MILP formulation (3). Of course, as mentioned in Sec. 2, GORE itself is implemented using Gurobi; this combination thus enabled a cogent test of the benefit of GORE.

We emphasize that GORE is a *preprocessing* routine that is independent of any exact algorithm for maximum consensus [15, 16, 27, 9, 6] — therefore, these algorithms should not be seen as "competitors" of our work. While we expect preprocessing by GORE to also significantly speed-up these algorithms, the lack of mature implementations hamper accurate evaluations. More relevant competitors are [22, 5]. However, these methods are specialized for 2D rigid transform and 3D rotations, where the associated maximum consensus problems do not have MILP formulations, thus direct comparisons with GORE are infeasible.

To evaluate the practicality of GORE+EXACT, we compared against the following approximate methods:

- RANSAC [10];
- l_∞ outlier removal [20]; and
- l_1 outlier removal [17].

Our method was applied on several common problems in computer vision. The experiments were carried out on a standard 2.70 GHz machine with 128 GB of RAM. A value of $M = 1000$ was used in all the MILP instances for solving computer vision problems, and a value of $M = 10000$ was used to solve synthetic data. A demo program (requiring Matlab and Gurobi) is given in the supplementary material.

Note that GORE (Algorithm 1) requires an approximate solution to obtain the upper bound $\hat{u}$. In our experiments, we used RANSAC, but any approximate method is applicable. Note also that the runtime of the approximate method was included in the runtime of GORE in our results below.

4.1. Synthetic data

Synthetic data in the form $\mathcal{X} = \{\mathbf{A}_i, \mathbf{b}_i, \mathbf{c}_i, d_i\}_{i=1}^N$ (17) was produced to test GORE. For simplicity, in this experiment we used $\mathbf{c}_i = \mathbf{0}$ and $d_i = 1$ for all i. The ground truth $\boldsymbol{\theta} \in \mathbb{R}^L$ was generated uniformly in $[-1, 1]^L$. Each $\mathbf{A}_i$ was drawn uniformly randomly from $[-50, 50]^{2\times L}$, and $\mathbf{b}_i$ was obtained as $\mathbf{b}_i = -\mathbf{A}_i\boldsymbol{\theta}$. This justifies the residual

$$\|\mathbf{A}_i\boldsymbol{\theta} + \mathbf{b}_i\|_p, \tag{24}$$

which is a special case of (16). To simulate outliers, 55% of the "dependent measurements" $\{\mathbf{b}_i\}_{i=1}^N$ were perturbed

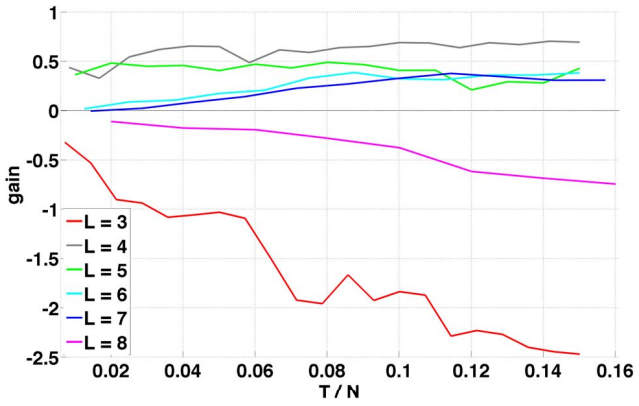


Figure 3. Computational gain of GORE on synthetic data for dimensions $L = 3, .., 8$ and increasing number of rejection tests T as a ratio of problem size N. Time per test c is fixed at 15s.

with i.i.d. uniform noise in the range $[-50, 50]$, while the rest were perturbed with i.i.d. Normal inlier noise with $\sigma = 1$. This produced an outlier rate of 50%–60%.

Combinations of (L, N) tested were $(3, 140)$, $(4, 120)$, $(5, 100)$, $(6, 80)$, $(7, 70)$, and $(8, 50)$. We reduced N for higher L to avoid excessively long runtimes — this does not invalidate our assessment of GORE since we are primarily interested in the speed-up ratio. For each (L, N) pair, we created 20 data instances $\mathcal{X}$. On each $\mathcal{X}$, we executed EXACT and GORE+EXACT with $\epsilon = 2$. For GORE, the number of rejection tests T was varied from 1 to $\lceil 0.15N \rceil$, while the maximum duration per test c was fixed to 15s.

The computational gain achieved by GORE on input data $\mathcal{X}$ can be expressed as the ratio

$$1 - \frac{\text{runtime of GORE+EXACT}}{\text{runtime of EXACT}}. \tag{25}$$

The median gain across for varying T (expressed as a ratio of N) are shown in Fig. 3. The gain is negative for $L = 3$, since EXACT was very fast on such a low dimension and running GORE simply inflated the runtime (although in Fig. 1 a large positive gain was obtained for a problem with $L = 3$, the type of data and residual function used were different). For $L = 4$ to 7, the gain increases with T, implying that as more true outliers were removed, the runtime of EXACT was reduced quickly. After $T \approx 0.1N$, the gain increases very slowly. For $L = 8$, however, the gain is negative; this was because GORE was unable to reject sufficient true outliers within the limit of $c = 15s$ for the preprocessing to pay off. The result suggests that there exist a lower and upper limit on L for GORE to be useful.

4.2. Affine image matching

Given two images I and I', we aim to find the 6DOF affine transform $\mathbf{T} \in \mathbb{R}^{2\times 3}$ that aligns the images, where $\mathbf{x} \in I$ and $\mathbf{x}' \in I'$. Let $\mathcal{X} = \{\mathbf{x}_i, \mathbf{x}'_i\}_{i=1}^N$ represent a set of putative correspondences for estimating $\mathbf{T}$. Given a $\mathbf{T}$, the matching error for the i-th correspondence is

$$\left\| \mathbf{x}'_i - \mathbf{T}[\mathbf{x}_i^T \, 1]^T \right\|_p. \tag{26}$$

The matching error above can be rewritten as

$$\left\| \mathbf{x}'_i - \mathbf{X}_i \mathbf{t} \right\|_p, \tag{27}$$

where $\mathbf{t} \in \mathbb{R}^6$ is the vectorized form of $\mathbf{T}$, and

$$\mathbf{X}_i = \begin{bmatrix} [\mathbf{x}_i^T \; 1] & \mathbf{0}_{1\times 3} \\ \mathbf{0}_{1\times 3} & [\mathbf{x}_i^T \; 1] \end{bmatrix} \in \mathbb{R}^{2\times 6}. \tag{28}$$

Clearly (27) is a special case of (16), thus our MILP-based GORE can be applied to remove outliers.

We tested our approach on images that have been previously used for affine image matching: Wall, Graffiti and Boat from the affine covariant features dataset [2], and Tissue and Dental from a medical image registration dataset [24, 25]; see Fig. 4. The images were resized before SIFT features were detected and matched using the VLFeat toolbox [23] to yield ≈ 100 point matches per pair.

For GORE (Algorithm 1), the upper bound $\hat{u}$ was obtained by running RANSAC for 10,000 iterations, T was set to 10 and c was set to 15s. In this experiment, we used $p = \infty$ in the matching error (26) for outlier rejection.

Table 1 shows the results, where for each method or pipeline, we recorded the obtained consensus size and total runtime; the size of the reduced input $\mathcal{X}'$ by GORE is also shown. As expected, although the approximate methods were fast, they did not return the globally optimal result. GORE was able to reduce $\mathcal{X}$ by 5 to 10 true outliers with the given runtime. However, this was sufficient to significantly reduce the runtime of solving (1) such that GORE+EXACT was much faster than EXACT on all of the data instances (respectively 78%, 79%, 57%, 35% and 63% gain).

4.3. Affine epipolar geometry estimation

Following [15, Sec. 6.3], we aim to estimate the affine fundamental matrix $\mathbf{F}$ that underpins two overlapping images I and I' captured using affine cameras, where

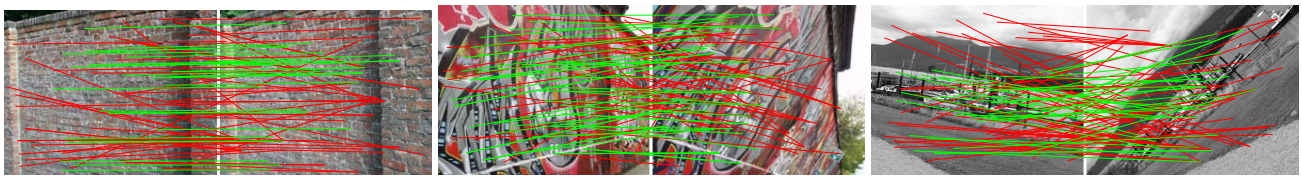
$$\mathbf{F} = \begin{bmatrix} 0 & 0 & f_{1,3} \\ 0 & 0 & f_{2,3} \\ f_{1,3} & f_{2,3} & f_{3,3} \end{bmatrix} \tag{29}$$

is a homogeneous matrix (4DOF). Let $\mathcal{X} = \{\mathbf{x}_i, \mathbf{x}'_i\}_{i=1}^N$ be a set of putative correspondences across I and I'. Given an $\mathbf{F}$, the algebraic error for the i-th correspondence is

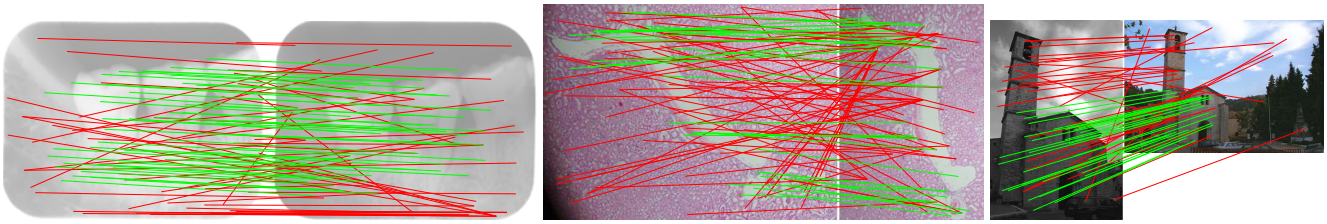
$$\tilde{\mathbf{x}}_i'^T \mathbf{F} \tilde{\mathbf{x}}_i, \tag{30}$$

where $\tilde{\mathbf{x}}$ is $\mathbf{x}$ in homogeneous coordinates. Linearizing and dehomogenizing using standard manipulations [12, Sec. 4.1.2] allows (30) to be written in regression form as

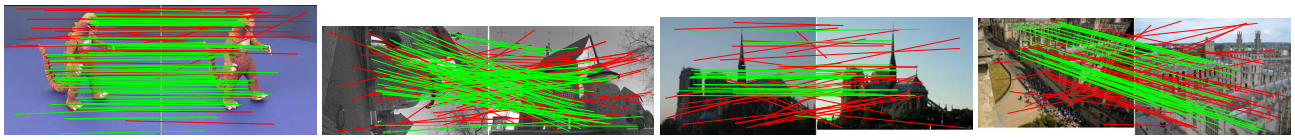
$$|\boldsymbol{\alpha}_i \mathbf{f} - \boldsymbol{\beta}_i| \tag{31}$$



(a) Wall (b) Graffiti (c) Boat



(d) Dental (e) Tissue (f) Valbonne



(g) Dinosaur (frames 7 and 8) (h) Kapel (i) Notre Dame (j) All Souls

Figure 4. Image pairs with SIFT correspondences used in the experiments. Green lines indicate true inliers while red lines indicate true outliers. (a)–(e) were used for affine image matching, and (f)–(g) were used for affine epipolar geometry estimation.

	Wall $N = 85, \epsilon = 1$			Graffiti $N = 92, \epsilon = 1$			Boat $N = 98, \epsilon = 1$			Tissue $N = 110, \epsilon = 2$			Dental $N = 101, \epsilon = 2$		
Methods	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{X}'\rvert$	time(s)	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{X}'\rvert$	time(s)	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{X}'\rvert$	time(s)	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{X}'\rvert$	time(s)	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{X}'\rvert$	time(s)
l_∞ method [20]	9		0.04	25		0.04	8		0.05	2		0.05	21		0.04
l_1 method [17]	14		0.02	20		0.02	3		0.02	0		0.02	16		0.02
RANSAC [10]	27		0.88	42		0.89	34		0.91	35		0.89	48		0.89
EXACT	31		1028.07	47		472.90	34		603.55	37		284.07	49		258.34
GORE		75	45.25		82	70.08		88	129.39		100	90.82		91	64.41
GORE+EXACT	31		226.45	47		100.41	34		262.58	37		185.59	49		96.33

Table 1. Results of affine image matching. N = size of input data $\mathcal{X}$, ϵ = inlier threshold (in pixels) for maximum consensus, $|\mathcal{I}|$ = size of optimized consensus set, $|\mathcal{X}'|$ = size of reduced data by GORE.

where $\mathbf{f} \in \mathbb{R}^4$ contains four elements from $\mathbf{F}$, and $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$ contain monomials calculated from $\{\mathbf{x}_i, \mathbf{x}'_i\}$.

We tested our approach on images that have been previously used for affine epipolar geometry estimation [3]: Dinosaur (frames 7 and 8), Kapel and Valbonne Church from the Oxford Visual Geometry Group's (VGG) multi-view reconstruction dataset, Notre Dame from VGG's Paris Dataset, and All Souls from VGG's Oxford Buildings Dataset; see Fig. 4. The images were resized before SIFT features were detected and matched using the VLFeat toolbox [23] to yield approx. 50–100 matches per pair.

For GORE (Algorithm 1), the upper bound $\hat{u}$ was obtained by running RANSAC for 10,000 iterations, T was set to 10 and c was set to 15s.

Table 1 shows the results. In general, despite having a lower DOF than an affine transform, solving maximum consensus exactly on the affine fundamental matrix takes longer time. Also, unlike in affine image matching, GORE was less prolific in removing true outliers; on average it was able to remove just 2 true outliers out of $T = 10$ attempts. Interestingly, however, we can see rejecting just 2 true outliers was able to speed-up significantly the overall runtime of GORE+EXACT (at least for 3 of the image pairs).

4.4. Triangulation

Given N 2D image measurements $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ of the same 3D point and the camera matrix $\mathbf{P}_i \in \mathbb{R}^{3\times4}$ for each of the view, we wish to estimate the coordinates $\boldsymbol{\theta} \in \mathbb{R}^3$ of the point. For an estimate $\boldsymbol{\theta}$, the residual in the i-th view can be written as the reprojection error [14]

$$\frac{\|(\mathbf{P}_{i,1:2} - \mathbf{x}_i\mathbf{P}_{i,3})\tilde{\boldsymbol{\theta}}\|_p}{\mathbf{P}_{i,3}\tilde{\boldsymbol{\theta}}} \quad \text{with} \quad \mathbf{P}_{i,3}\tilde{\boldsymbol{\theta}} > 0, \tag{32}$$

	Dinosaur $N = 115, \epsilon = 1$			Kapel $N = 107, \epsilon = 2$			Notre Dame $N = 69, \epsilon = 2$			All Souls $N = 76, \epsilon = 2$			Valbonne $N = 63, \epsilon = 2$		
Methods	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{X}'\rvert$	time(s)	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{X}'\rvert$	time(s)	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{X}'\rvert$	time(s)	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{X}'\rvert$	time(s)	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{X}'\rvert$	time(s)
l_∞ method [20]	30		0.03	42		0.03	14		0.03	31		0.02	13		0.03
l_1 method [17]	38		0.01	43		0.01	7		0.01	15		0.01	20		0.01
RANSAC [10]	68		1.04	62		0.99	28		1.00	39		0.99	29		0.98
EXACT	71		3924.57	65		5538.66	31		515.08	42		2795.35	32		593.34
GORE		113	135.95		104	148.27		68	150.92		72	75.25		62	59.58
GORE+EXACT	71		1237.90	65		1263.25	31		634.98	42		858.06	32		470.91

Table 2. Results of affine epipolar geometry estimation. N = size of input data $\mathcal{X}$, ϵ = inlier threshold (in pixels) for maximum consensus, $|\mathcal{I}|$ = size of optimized consensus set, $|\mathcal{X}'|$ = size of reduced data by GORE.

	Point 1 $N = 167, \epsilon = 1$			Point 2 $N = 105, \epsilon = 1$			Point 36 $N = 175, \epsilon = 1$			Point 682 $N = 153, \epsilon = 1$			Point 961 $N = 129, \epsilon = 1$		
Methods	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{X}'\rvert$	time(s)	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{X}'\rvert$	time(s)	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{X}'\rvert$	time(s)	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{X}'\rvert$	time(s)	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{X}'\rvert$	time(s)
l_∞ method [20]	96		1.93	23		1.83	63		2.17	79		1.78	58		1.81
l_1 method [17]	111		0.03	33		0.03	73		0.04	87		0.03	61		0.03
RANSAC [10]	114		1.16	36		1.12	84		1.15	96		1.13	69		1.12
EXACT	115		17.80	38		112.83	90		956.35	97		56.44	70		110.90
GORE		157	14.65		95	11.02		165	20.09		143	16.12		119	8.55
GORE+EXACT	115		27.80	38		75.67	90		771.32	97		35.40	70		25.86

Table 3. Results of triangulation. N = size of input data $\mathcal{X}$, ϵ = inlier threshold (in pixels) for maximum consensus, $|\mathcal{I}|$ = size of optimized consensus set, $|\mathcal{X}'|$ = size of reduced data by GORE.

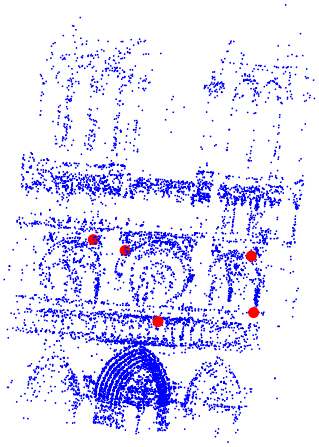


Figure 5. Illustration of the first 10,000 points in the Notre Dame dataset. Points computed in Table 3 are highlighted using red.

where $\tilde{\boldsymbol{\theta}} = [\boldsymbol{\theta}^T \ 1]^T$, $\mathbf{P}_{i,1:2}$ is the first-two rows of $\mathbf{P}_i$, and $\mathbf{P}_{i,3}$ is the third row of $\mathbf{P}_i$. The strictly positive constraint on $\mathbf{P}_{i,3}\tilde{\boldsymbol{\theta}}$ ensures that $\boldsymbol{\theta}$ lies in front of the cameras. Observe that (32) is a special case of (16).

We used the Photo Tourism dataset from [21], which contained a set of image observations and camera matrices. We specifically chose 5 of the points with more than 100 views ($N > 100$) to triangulate; the index of these points are listed in Table 3. With an ϵ of 1 pixel, these problem instances contain 40% to 60% outliers. Note that our model (32) ignores radial distortion error, thus the actual outlier rates could be lower. Nonetheless, for the purpose of testing the speed-up by GORE, our experiment is sufficient.

We used $p = \infty$ in (32). For GORE, T was set to 10 and c was set to 15 s. Table 3 shows the results. For 4 of the 5 points, preprocessing with GORE provided considerable reduction in total runtime, especially in cases with high outlier rate. Fig. 5 shows the estimated 3D points.

5. Conclusions

We propose GORE as a way to speed up the exact solution of the maximum consensus problem. GORE preprocesses the data to identify and remove instances guaranteed to be outliers to the globally optimal solution. The proposed method works by inspecting data instances that are most likely outliers to the global solution, and prove using contradiction that it is a true outlier. The rejection test is based on comparing upper and lower bound values derived using MILP. We also showed how GORE can be extended to deal with geometric residuals that are quasiconvex. The fact that GORE is formulated as a MILP allows it to be easily implemented using any off-the-shelf optimization software.

GORE is experimentally evaluated on synthetically generated and real data, based on common computer vision applications. While there are limitations to GORE in terms of the range of model dimensions that it can handle, we demonstrate that it is still very useful and effective on a wide range of practical computer vision applications.

Acknowledgements The authors were supported by ARC grants DP160103490, DE130101775 and DP140103400.

References

[1] https://en.wikipedia.org/wiki/Lp_space. 4

[2] http://www.robots.ox.ac.uk/~vgg/research/affine/index.html. 6

[3] R. Arandjelović and A. Zisserman. Efficient image retrieval for 3D structures. In *Proceedings of the British Machine Vision Conference*, 2010. 7

[4] A. B. Bordetski and L. S. Kazarinov. Determining the committee of a system of weighted inequalities. *Kibernetika*, 6:44–48, 1981. 1

[5] A. P. Bustos and T.-J. Chin. Guaranteed outlier removal for rotation search. In *ICCV*, 2015. 2, 5

[6] T.-J. Chin, P. Purkait, A. Eriksson, and D. Suter. Efficient globally optimal consensus maximisation with tree search. In *CVPR*, 2015. 1, 5

[7] J. W. Chinneck. *Feasibility and infeasibility in optimization: algorithms and computational methods*. Springer, 2008. 1, 2, 3

[8] M. Conforti, G. Cornuéjols, and G. Zambelli. *Integer programming*. Springer, 2014. 2, 3, 4

[9] O. Enqvist, E. Ask, F. Kahl, and K. Åström. Robust fitting for multiple view geometry. In *ECCV*, 2012. 1, 5

[10] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24(6):381–395, 1981. 1, 5, 7, 8

[11] R. Greer. *Trees and hills: methodology for maximizing functions of systems of linear relations*. Annals of Discrete Mathematics 22. North-Holland, 1984. 1

[12] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2nd edition, 2004. 6

[13] R. I. Hartley and F. Schaffalitzky. L-infinity minimization in geometric reconstruction problems. In *CVPR*, 2004. 4

[14] F. Kahl and R. Hartley. Multiple-view geometry under the l_∞ norm. *IEEE TPAMI*, 30(9):1603–1617, 2008. 1, 2, 4, 7

[15] H. Li. Consensus set maximization with guaranteed global optimality for robust geometry estimation. In *ICCV*, 2009. 1, 2, 5, 6

[16] C. Olsson, O. Enqvist, and F. Kahl. A polynomial-time bound for matching and registration with outliers. In *CVPR*, 2008. 1, 5

[17] C. Olsson, A. Eriksson, and R. Hartley. Outlier removal using duality. In *CVPR*, 2010. 2, 5, 7, 8

[18] M. Padberg. *Linear optimization and extensions*. Springer, 1999. 3

[19] M. Pfetsch. Branch-and-cut for the maximum feasible subsystem problem. *SIAM Journal on Optimization*, 19(1):21–38, 2008. 1

[20] K. Sim and R. Hartley. Removing outliers using the l_∞ norm. In *CVPR*, 2006. 2, 5, 7, 8

[21] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. http://phototour.cs.washington.edu/, 2006. 8

[22] L. Svärm, O. Enqvist, M. Oskarsson, and F. Kahl. Accurate localization and pose estimation for large 3d models. In *CVPR*, 2014. 2, 5

[23] A. Vedaldi and B. Fulkerson. http://www.vlfeat.org/. 6, 7

[24] C.-W. Wang. Improved image registration for biological and medical data. http://www-o.ntust.edu.tw/~cweiwang/ImprovedImageRegistration/. 6

[25] C.-W. Wang and H.-C. Chen. Improved image alignment method in application to x-ray images and biological images. *Bioinformatics*, 29(15):1879–1887, 2013. 6

[26] J. Yu, A. Eriksson, T.-J. Chin, and D. Suter. An adversarial optimization approach to efficient outlier removal. In *ICCV*, 2011. 2

[27] Y. Zheng, S. Sugimoto, and M. Okutomi. Deterministically maximizing feasible subsystems for robust model fitting with unit norm constraints. In *CVPR*, 2011. 1, 2, 5