J Math Imaging Vis
DOI 10.1007/s10851-013-0418-7

An Adversarial Optimization Approach to Efficient Outlier Removal

Jin Yu · Anders Eriksson · Tat-Jun Chin · David Suter

Abstract This paper proposes a novel adversarial optimization approach to efficient outlier removal in computer vision. We characterize the outlier removal problem as a game that involves two players of conflicting interests, namely, model optimizer and outliers. Such an adversarial view not only brings new insights into some existing methods, but also gives rise to a general optimization framework that provably unifies them. Under the proposed framework, we develop a new outlier removal approach that is able to offer a much needed control over the trade-off between reliability and speed, which is usually not available in previous methods. Underlying the proposed approach is a mixed-integer minmax (convex-concave) problem formulation. Although a minmax problem is generally not amenable to efficient optimization, we show that for some commonly used vision objective functions, an equivalent Linear Program reformulation exists. This significantly simplifies the optimization. We demonstrate our method on two representative multiview geometry problems. Experiments on real image data illustrate superior practical performance of our method over recent techniques.

Keywords Mixed-integer optimization · Convex relaxation · Model fitting · Outlier removal

J. Yu · A. Eriksson (✉) · T.-J. Chin · D. Suter
Australian Centre for Visual Technologies, School of Computer Science, University of Adelaide, North Terrace, SA 5005, Australia
e-mail: anders@cs.adelaide.edu.au

J. Yu
e-mail: jin.yu@cs.adelaide.edu.au

T.-J. Chin
e-mail: tjchin@cs.adelaide.edu.au

D. Suter
e-mail: dsuter@cs.adelaide.edu.au

Published online: 22 February 2013

1 Introduction

Model fitting as a fundamental problem in computer vision underlies numerous applications. It is typically posed as the minimization of an objective function on some input data. For instance, a particularly successful line of research in multiview geometry [8, 11] casts various multiview reconstruction problems as minimizing the maximum reprojection error (i.e., the L_∞ norm of the error vector) across all measurements. Such methods have demonstrated excellent performance on a variety of applications. However, they are also known [29] to be extremely vulnerable to outliers, which unfortunately are often unavoidable, due to imperfections in data acquisition and preprocessing, and, more generally, to failure of assumptions; e.g. only one dominant structure in the data. This paper provides a principled and unified framework for outlier removal in single-structure model fitting by casting outlier removal as a zero-sum two-player game. This not only brings new insights into some existing outlier removal schemes [19, 23, 29], but also gives rise to a general problem formulation that provably unifies them.

RANSAC [1] is probably the most commonly used method to handle outliers. The hope is that a "clean" data sample would be drawn so that a model that is representative of the genuine structure in the data can be estimated purely from the noise-free sample. RANSAC has met with great success in computer vision, though its efficiency largely relies on fast computation of candidate models. Where not possible, RANSAC has to resort to "cheap" but less accurate alternatives. For instance, when applied to multiview geometry problems, RANSAC and its accelerated variants, e.g. preemptive RANSAC [21], can only afford to compute candidate models from 2 or 3 views, instead of full tracks of the data. While with reduced sampling RANSAC is fast, it is unreliable, which can cause some outliers being undetected.

Even though the undetected outliers often only constitute a small portion of the data ($<15\ \%$), they can still cause disastrous results, especially when L_∞-norm-based methods are used. Such a limitation of RANSAC motivates the development of various alternative robust model fitting techniques [12, 14, 23, 28, 29].

Among them, the work of [12, 14] focuses on the optimization of robust objective functions tailored for multiview geometry problems. Similar to the least kth-order squares method [13], the method proposed in [12] approximately minimizes the kth smallest reprojection error, k being less than the overall number of the data. Unfortunately, since the resulting optimization problem has multiple local minima, the obtained solution is most likely to be sub-optimal. A search-based method was later proposed in [14]. This method seeks to identify a pre-specified number of data that produce the smallest cost. While this method comes with a global optimality guarantee, it is computationally intractable for large-scale problems. Similarly, in the context of regression the Least Trimmed Squares method [26] as a robust version of the least squares approach also suffers from computational efficiency issue, due to its combinatorial nature. Its approximate variants, e.g. FAST-LTS [25], are computationally less expensive, but only provide approximate solutions. Other robust model fitting methods, e.g. various M-estimators [10], can potentially be used to bound the influence of outliers. Although having only bounded influence, outliers still participate in the optimization, hence can still bias the solution. It is therefore desirable to exclude them completely from the data before invoking a model fitting method.

To eliminate the influence of outliers, RANSAC-type of sampling-based methods seek to identify a subset I of the input data such that a model $\mathbf{w}$ (e.g. a homography) exists with

$$f_i(\mathbf{w}) \le \epsilon, \quad \forall i \in I \subseteq \{1, \dots, N\}, \tag{1}$$

where the cost function $f_i(\mathbf{w})$ evaluates the discrepancy between datum i and $\mathbf{w}$, and $\epsilon \ge 0$ is a pre-specified error tolerance. Some recent global-optimization-based work aims to achieve the same goal but in a deterministic fashion. Ideally, one would aim for the largest subset I, and there exists work [4, 15] that targets this goal. However, these methods are either prohibitively expensive for high dimensional $\mathbf{w}$ (with a worst-case exponential complexity), or only tailored for a very specific class of applications. Recent research, e.g. [29], relaxed the goal of maximizing I to finding an I of sufficient size that is representative of the underlying structure in the data. The goal there is to identify and remove "noisy" data until a model is found to fulfil (1). Our work is guided by the same objective; also following [29], we say that *a dataset contains at least one outlier if there does not exist a model* $\mathbf{w}$ *that satisfies* (1) *on that dataset*.

Exploiting the special properties of strictly quasi-convex objective functions, as commonly used in multiview geometry, the outlier removal method proposed in [29] iteratively excludes data that generate the largest residual. This intuitive approach guarantees that at each iteration at least one outlier is among the removed data (assuming at least one exists), i.e., a $\mathbf{w}$ that satisfies (1) can not be found by fitting to the removed data. Although quite effective, the method is computationally expensive, due to the need of solving a quasi-convex problem at each iteration.

More recent research [23, 28] introduces slack variables into the objective function, and devises outlier removal schemes by analysing slack values. Take the following *1-slack* objective function as an example:

$$\min_{s,\mathbf{w}} s \quad \text{s.t.} \quad f_i(\mathbf{w}) \le \epsilon + s, \quad \forall i, \tag{2}$$

where $s \in \mathbb{R}$ is a slack variable. Under such a problem formulation, if the optimal s is positive, then data that generate the largest residual are guaranteed to contain at least one outlier. Based on this result, an iterative outlier removal scheme was developed, and shown to be equivalent to the strategy proposed in [29], yet much faster. Since at each iteration the "1-slack method" can only remove a limited number of outliers, it often requires many iterations to entirely clean the data, thus is too computationally expensive to be widely applicable for practical use.

In an attempt to accelerate the process, Olsson et al. [23] examines a variant of (2) that uses N "slacks":

$$\min_{s_i,\mathbf{w}} \sum_i s_i \quad \text{s.t.} \quad f_i(\mathbf{w}) \le \epsilon + s_i, \quad s_i \ge 0, \quad \forall i. \tag{3}$$

If we collect all the non-negative slack variables into a vector, then (3) essentially computes the L_1 norm of the vector of the slack variables (or sum of infeasibilities). Using this problem formulation, data associated with positive s_i can be removed as outliers *in one shot*. The L_1 method is therefore fast and shown to often work well in practice. However, this one-shot approach can sometimes lead to arbitrarily bad models. Figure 1 depicts a senario where fitting to the removed data (circles) produces a model $\mathbf{w}$ that satisfies (1), a model that, in fact, represents the desired structure.

Noting that the 1-slack and L_1 methods represent two ends of the spectrum of reliability versus speed, we aim for a general problem formulation that is able to strike a balance between these two performance factors. Our method is inspired by the classical minmax formulation of a zero-sum two-player game [27] in which one player makes moves to maximize his payoff, while the other player tries to minimize the payoff of his opponent. In the context of outlier removal, the optimizer can be seen as a player who tries to minimize the objective function, whereas the interest of outliers is to make the minimal objective as bad as possible. Such an adversarial view can be translated to a convex-concave optimization problem. Solving it allows us to effectively identify a set of "offending" data that are responsible for a "bad" objective value. To identify all outliers, one simply needs to remove the "offending" data, and repeat the process until the remaining data are clean. Figure 2 depicts one run of the three outlier removal schemes on sample 2D line-fitting data (dots) with 35 % of outliers. It shows that our approach (top right) identifies noticeably more outliers than the 1-slack method (top left), yet achieving so in a less "aggressive" manner than the L_1 method (bottom). Later, we show that the two existing methods are in fact special cases of the proposed approach.

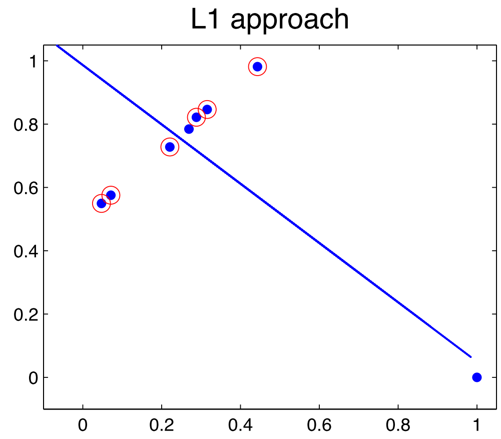


Fig. 1 An example that the L_1 formulation fails. The data removed by the L_1 method (*circles*) do not contain outliers of the desired structure, resulting in an incorrect model (*line*)

The paper proceeds as follows: We first provide a general description of our problem formulation. Section 3 shows how to optimally solve the resulting optimization problem. In Sect. 4 we discuss related work. Experiments on real data are reported in Sect. 6. Finally, we conclude the paper in Sect. 7.

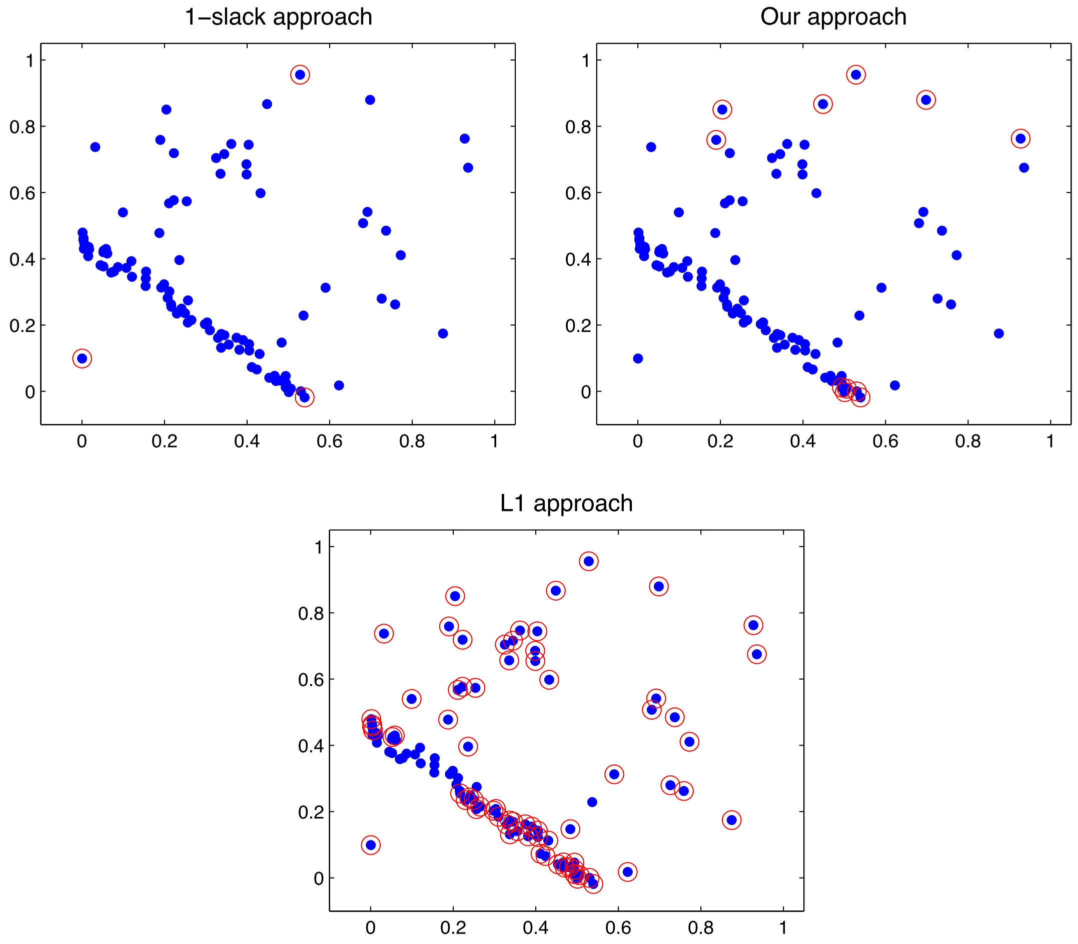


Fig. 2 Behaviour of three outlier removal methods on sample 2D line-fitting data (*dots*). *Top left*: the 1-slack method removes only 3 data points (*circles*), hence many iterations are needed to clean the data, whereas the overly "aggressive" L_1 method (*bottom*) mistakenly removes many inliers. Our approach (*top right*) offers a more balanced result: it is clearly more productive than the 1-slack method, yet not sacrificing too many inliers

2 Problem Formulation

This section introduces our adversarial problem formulation for outlier removal.

2.1 Outlier Removal as a Minmax Problem

We view the outlier removal problem as a zero-sum game that involves two competing players: optimizer and outliers. The optimizer aims to find a model that achieves the minimal objective value, whereas the outliers' strategy is to make the minimal objective value as bad as possible. To mathematically formulate such a game, we first introduce a binary variable $\boldsymbol{\pi} \in \{0,1\}^N$ (All vectors are by default column vectors.); each of its entries, denoted by π_i, corresponds to a datum; $\pi_i = 1$ indicates that datum i is an outlier; otherwise $\pi_i = 0$. Taking into account the fact that outliers only constitute part of the data, we enforce $\boldsymbol{\pi}^\top \mathbf{1} = K$, where $\mathbf{1}$ is a vector of all ones and $K \in [1, N]$. We use the sum of non-negative slack variables to measure model fitting error. The resulting adversarial problem takes the following form:

$$\min_{\mathbf{s},\mathbf{w}} \max_{\boldsymbol{\pi}} \; \boldsymbol{\pi}^\top \mathbf{s} \tag{4a}$$

$$\text{s.t.} \quad f_i(\mathbf{w}) \le \epsilon + s_i, \quad s_i \ge 0, \quad \forall i \tag{4b}$$

$$\boldsymbol{\pi} \in \{0,1\}^N, \quad \boldsymbol{\pi}^\top \mathbf{1} = K, \tag{4c}$$

where $\mathbf{s} \in \mathbb{R}_+^N$ collects the N slack variables s_i, each corresponding to a datum. The outer $\min_{\mathbf{s},\mathbf{w}}$ and inner $\max_{\boldsymbol{\pi}}$ operations characterize the strategies of the optimizer and outliers, respectively. Denote the objective of (4a)–(4c) by $J(\mathbf{s}, \mathbf{w}, \boldsymbol{\pi})$. Then, for a joint action $(\mathbf{s}, \mathbf{w}, \boldsymbol{\pi})$, the payoffs of the optimizer and outliers are $-J(\mathbf{s}, \mathbf{w}, \boldsymbol{\pi})$ and $J(\boldsymbol{\pi}, \mathbf{s}, \mathbf{w})$, respectively. In the equilibrium, the optimizer minimizes the sum of the K-largest slacks identified by the K positive entries of the vector $\boldsymbol{\pi}$.

2.2 A Relaxed Minmax Formulation

The problem (4a)–(4c) may appear difficult to optimize due to the presence of the discrete constraint on $\boldsymbol{\pi}$. Fortunately, the linearity of (4a)–(4c) in $\boldsymbol{\pi}$ allows us to relax the constraint $\boldsymbol{\pi} \in \{0,1\}^N$ to $\boldsymbol{\pi} \in [0,1]^N$ without changing the optimal objective value. To see this, for the moment, let us relax the discrete constraint to obtain the following relaxation:

$$\min_{\mathbf{s},\mathbf{w}} \max_{\boldsymbol{\pi}} \; \boldsymbol{\pi}^\top \mathbf{s} \tag{5a}$$

$$\text{s.t.} \quad f_i(\mathbf{w}) \le \epsilon + s_i, \quad s_i \ge 0, \quad \forall i \tag{5b}$$

$$\boldsymbol{\pi} \in [0,1]^N, \quad \boldsymbol{\pi}^\top \mathbf{1} = K. \tag{5c}$$

The inner maximization problem of (5a)–(5c) is simply a Linear Program (LP). This ensures that the optimal $\boldsymbol{\pi}$ can be attained at vertices of the linear constraints (5c). Since the vertices of these constraints are integral, the optimal $\boldsymbol{\pi}$ attained at a vertex is integral. Therefore, the relaxation (5a)–(5c) is *equivalent* to its mixed-integer counterpart (4a)–(4c). However, note that the solution to the inner $\max_{\boldsymbol{\pi}}$ problem is not necessarily unique, and in such cases fractional solutions exist.

Nevertheless, one can always obtain an integral solution for $\boldsymbol{\pi}$ by analysing the slack values. Let $(\mathbf{s}^*, \mathbf{w}^*, \boldsymbol{\pi}^*)$ denote the optimal solution to (5a)–(5c), and let s^K be the Kth largest value of $\mathbf{s}^*$. If $\mathbf{s}^*$ has more than one entry that shares the value of s^K, then $\boldsymbol{\pi}^*$ is not unique, and can be fractional since it can spread the mass of 1 over the data with the same slack value of s^K. However, from the construction of the objective function (5a), it is easy to see that an integral $\boldsymbol{\pi}^*$ can be obtained by selecting exactly K elements from the K-largest entries of $\mathbf{s}^*$ and setting their corresponding π_i to 1.

2.3 Identification of Potential Outliers

Having solved (5a)–(5c), we can identify potential outliers by forming the following *potential outlier set*:

$$\mathcal{O} := \left\{ i \in \{1, \dots, N\} \mid s_i^* \ge s^{\bar{K}} > 0 \right\}, \tag{6}$$

where $s^{\bar{K}} \ge s^K$ is the smallest *positive* entry among the K-largest entries of $\mathbf{s}^*$. If more than one slack variable takes on the value of $s^{\bar{K}}$, then $|\mathcal{O}| > K$, where $|\cdot|$ computes the cardinality of a set. Conversely, if $\mathbf{s}^*$ has fewer than K positive entries, then $|\mathcal{O}| < K$.

Before discussing the optimization strategy for solving (5a)–(5c), in Fig. 3 we first provide a pictorial illustration of its solution and the corresponding $\mathcal{O}$ on sample line-fitting data. We define the cost function as:

$$f_i(\mathbf{w}) := \left| \mathbf{x}_i^\top \mathbf{w} - y_i \right|, \tag{7}$$

where $\mathbf{x}_i$ consists of the x coordinate of datum i and 1, and y_i is the corresponding y coordinate. We used $\epsilon = 0.05$ and $K = 5$ throughout. Figure 3 shows that compared to the rest of the data, the potential outliers identified in $\mathcal{O}$ (circles) produce larger residuals with respect to the optimal model $\mathbf{w}^*$ (line), and clearly contain genuine outliers that are inconsistent with the prominent structure in the data.

If $|\mathcal{O}| < K$ (Fig. 3(left)), we immediately know $f_i(\mathbf{w}^*) \le \epsilon, \forall i \notin \mathcal{O}$, because $s_i^* = 0$ for those i. Therefore, after removing the data in $\mathcal{O}$, we can stop the outlier removal procedure. In the case of $|\mathcal{O}| \ge K$ Theorem 2 (Appendix A) proves that $\mathcal{O}$ contains at least one outlier (provided that the cost function $f_i(\mathbf{w})$ satisfies some mild technical conditions). Note that this is a very pessimistic performance bound on the number of identified outliers. The practical efficiency of our approach is often significantly

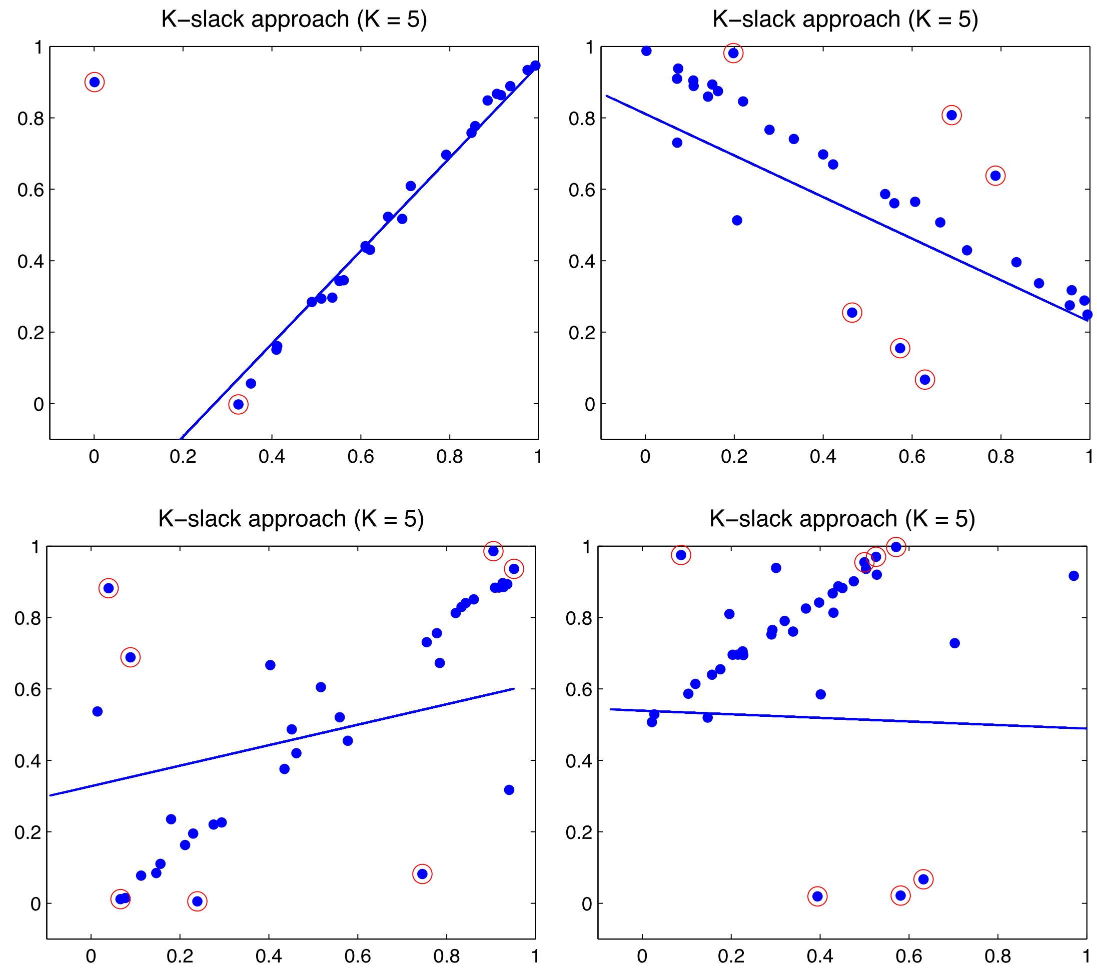


Fig. 3 The model (*line*) and the potential outliers (circles) identified in $\mathcal{O}$ (6) for sample 2D line-fitting data (*dots*). Data in $\mathcal{O}$ produce the K-largest residuals (w.r.t. the found model), and clearly contain genuine outliers

better. Figure 4 examines the empirical performance of our method on 500 random instances of line-fitting problems, each with 65 inliers and 35 outliers (i.e., outlier ratio of 35 %). We fixed $\epsilon = 0.05$ and $K = \lceil 5\ \%N \rceil$ throughout, N being adjusted to the number of remaining data after each iteration. It can be seen (top) that it takes 7–20 (most commonly 10) iterations to clean the data. Among the removed data, the average percentage of genuine outliers (with respect to ground truth models) starts out as high as 70 % (Fig. 4, bottom), which is well above the theoretical bound of $1/|\mathcal{O}| \leq 1/\lceil 5\ \%N \rceil = 20\ \%$ with $N = 100$. As more and more outliers are removed, the average percentage of detected outliers gradually reduces to the theoretical bound of $1/|\mathcal{O}| = 33\ \%$, where $|\mathcal{O}|$ is observed to equal 3 at the final two iterations of all runs.

3 An Equivalent Reformulation

One potential technical difficulty with the proposed min-max problem formulation (5a)–(5c) is that it is generally not amenable to efficient convex optimization. However, if the constraints in (5b) are linear in $\mathbf{w}$, an equivalent LP reformulation exists. This allows for the use of off-the-shelf large-scale LP solvers.

Noting that the objective function (5a) is bilinear in $\mathbf{s}$ and $\boldsymbol{\pi}$, we first decouple these two terms so as to eliminate non-convexity. This is done by analysing the dual of the inner $\max_{\pi}$ problem:

$$\max_{\boldsymbol{\pi}}\ \boldsymbol{\pi}^{\top}\mathbf{s} \tag{8a}$$

$$\text{s.t.} \quad \boldsymbol{\pi} \in [0,1]^N, \quad \boldsymbol{\pi}^{\top}\mathbf{1} = K. \tag{8b}$$

Introducing Lagrange multipliers $\alpha \in \mathbb{R}$ and $\boldsymbol{\beta} \in \mathbb{R}_+^N$, we can write the dual of (8a)–(8b) as:

$$\min_{\alpha,\boldsymbol{\beta}}\ \alpha K + \boldsymbol{\beta}^{\top}\mathbf{1} \quad \text{s.t.} \quad \alpha\mathbf{1} + \boldsymbol{\beta} \geq \mathbf{s}, \quad \boldsymbol{\beta} \geq \mathbf{0}, \tag{9}$$

where $\mathbf{0}$ denotes a column vector of all zeros. Replacing the inner $\max_{\pi}$ problem in (5a)–(5c) with (9) gives the follow-

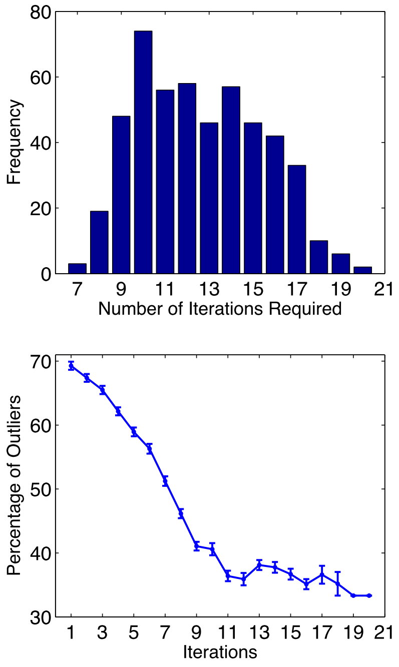


Fig. 4 *Top*: histogram of the number of iterations needed to clean 500 random line-fitting data with 35 % outliers. *Bottom*: average (with standard errors) outlier percentage among the data removed at each iteration

ing equivalent minimization problem:

$$\min_{\alpha,\boldsymbol{\beta},\mathbf{s},\mathbf{w}} \quad \alpha K + \boldsymbol{\beta}^\top \mathbf{1} \tag{10a}$$

$$\text{s.t.} \quad f_i(\mathbf{w}) \le \epsilon + s_i, \qquad \forall i \tag{10b}$$

$$\alpha \mathbf{1} + \boldsymbol{\beta} \ge \mathbf{s}, \quad \boldsymbol{\beta} \ge \mathbf{0}, \quad \mathbf{s} \ge \mathbf{0}. \tag{10c}$$

It is clearly a convex problem if $f_i(\mathbf{w})$ is convex; all problems considered in this paper are convex. However, it is worth noting that the reformulation technique adopted here does not require $f_i(\mathbf{w})$ to be convex. This means that (10a)–(10c) is equivalent to (5a)–(5c) regardless of the form of the cost function.

Now, suppose that $f_i(\mathbf{w})$ is linear, then (10a)–(10c) is simply an LP, and (10b) can be represented in compact matrix form as

$$\mathbf{A}_j \mathbf{w} \le \mathbf{b}_j + \mathbf{s}, \quad \forall j, \tag{11}$$

where the matrix $\mathbf{A}_j \in \mathbb{R}^{N\times d}$ and the vector $\mathbf{b}_j \in \mathbb{R}^N$ contain the coefficients of linear constraints enforced by (10b). The values of the coefficients and the range of j are determined by the exact form of the cost function $f_i(\mathbf{w})$. Take (7) for instance, it can be realized by two linear constraints:

$$\mathbf{x}_i^\top \mathbf{w} \le y_i + \epsilon \quad \text{and} \quad -\mathbf{x}_i^\top \mathbf{w} \le -y_i + \epsilon.$$

Expressing these constraints for all i in the matrix form of (11), we obtain two coefficient matrices: $\mathbf{A}_1 := [\mathbf{x}_1^\top; \ldots; \mathbf{x}_N^\top]$ and $\mathbf{A}_2 := -\mathbf{A}_1$, with their corresponding coefficient vectors given by $\mathbf{b}_1 := [y_1; \ldots; y_N] + \epsilon\mathbf{1}$ and $\mathbf{b}_2 := -[y_1; \ldots; y_N] + \epsilon\mathbf{1}$, respectively. Having solved the LP, one can sort the entries of the optimal $\mathbf{s}$ in non-ascending order to construct the potential outlier set $\mathcal{O}$ (6), remove all the data in $\mathcal{O}$, and repeat the process on the remaining data until the optimal objective value of (10a)–(10c) reduces to 0. Algorithm 1 details our approach (named K-slack).

Algorithm 1 K-Slack Outlier Removal Method

```
input data and the choice of $K$
output a subset $I$ of the data such that
        $\exists \mathbf{w} \in \mathbb{R}^d : f_i(\mathbf{w}) \le \epsilon, \ \forall i \in I$
repeat
  solve (10a)–(10c) on the current data to obtain the optimal solution $(\alpha^*, \boldsymbol{\beta}^*, \mathbf{w}^*, \mathbf{s}^*)$
  if $\alpha^* K + \boldsymbol{\beta}^{*\top}\mathbf{1} > 0$ then
    construct the potential outlier set $\mathcal{O}$ via (6)
    remove the data collected in $\mathcal{O}$
  end if
until $\alpha^* K + \boldsymbol{\beta}^{*\top}\mathbf{1} = 0$ or $|\mathcal{O}| < K$
optional: restore inliers from the removed data
```

4 Related Work

This section relates our work to other recent efforts in outlier removal, and shows that these methods can be unified under the proposed optimization framework.

4.1 Connections with the 1-Slack and L_1 Methods

We first establish the connection of our method with the 1-slack (2) and L_1 (3) approaches through its parameter K. It is clear that if $K = N$, the minmax problem (4a)–(4c) (and hence its reformulation (5a)–(5c)) reduces to the L_1 problem, since in this case all π_i are forced to take on the value of 1. At the other extreme of $K = 1$, (4a)–(4c) is equivalent to the 1-slack problem. To see this, first note that solving (2) is equivalent to minimizing the maximum residual measured by the single slack variable s. This can be formulated as the following minmax problem with N slack variables:

$$\min_{s_i,\mathbf{w}} \max_i \ s_i \quad \text{s.t.} \quad f_i(\mathbf{w}) \le \epsilon + s_i, \quad \forall i. \tag{12}$$

If there exist outliers in the data, the maximum s_i must be positive, hence we can add the constraint $s_i \ge 0$ for all i to (12) without changing its optimal objective value. Now, it is easy to recognize that (12) with the additional non-negativity constraint on s_i can be alternatively formulated as (4a)–(4c) with $K = 1$.

By Theorem 2, the identification of at least one outlier is guaranteed, provided $|\mathcal{O}| \ge K$, which is clearly true if

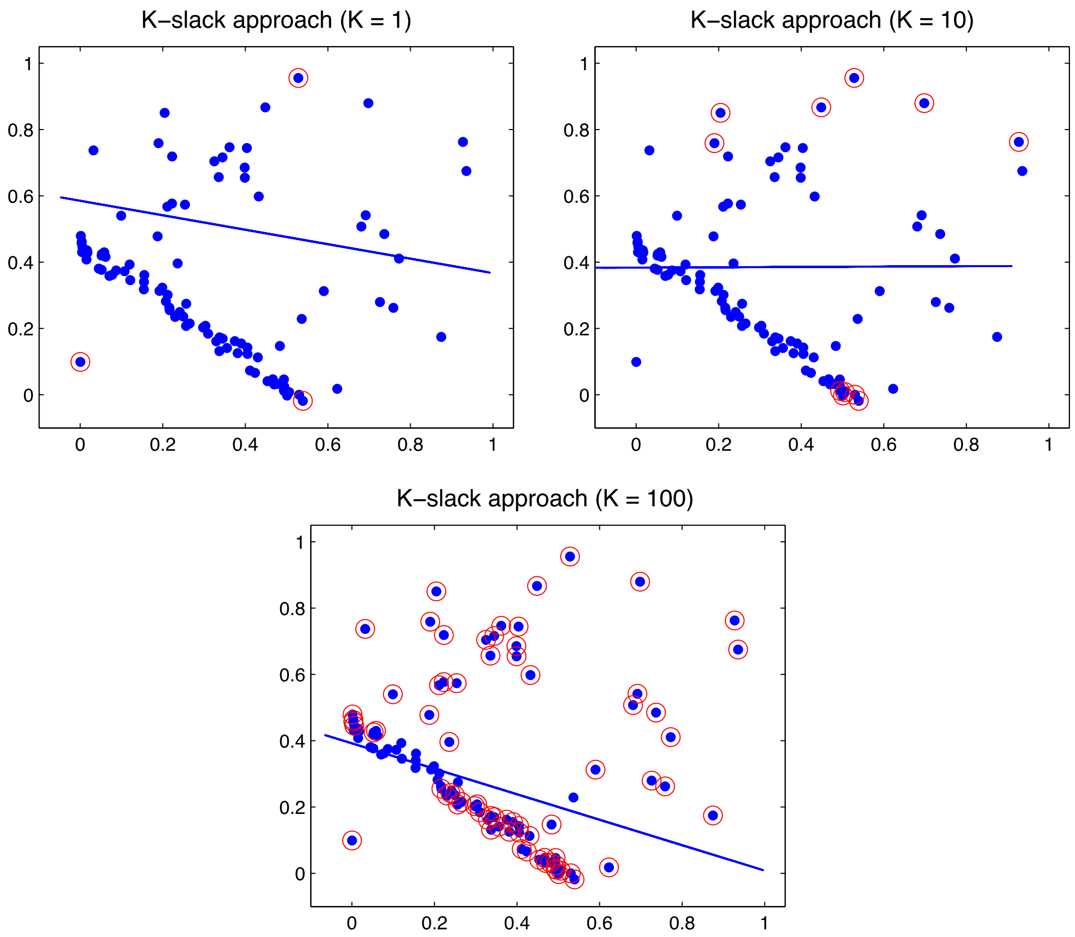


Fig. 5 The found model (*line*) and the potential outliers (*circles*) found by solving (5a)–(5c) with various K values on the same data as in Fig. 2. *Left*: our approach reduces to the 1-slack method (Fig. 2(left)) with $K = 1$, respectively to the L_1 method (Fig. 2(right)) with $K = N = 100$ (*right*)

$K = 1$; hence the 1-slack method is effective. On the contrary, if $K = N$, we always have $|\mathcal{O}| < K$. No performance guarantee can be provided for this case. This renders the L_1 method unreliable. When an intermediate K is used, at each of the iterations leading up to the final step, $|\mathcal{O}|$ can not be less than K (otherwise, Algorithm 1 would have stopped), therefore by Theorem 2 at least one outlier is guaranteed to be removed. Although at the final iteration, it is possible that $|\mathcal{O}| < K$ for a $K > 1$ and we lose the performance guarantee, the maximum possible number of inliers that can be mistakenly removed at the final step is bounded from above by K. Empirically, we found that to achieve a good compromise between reliability and speed, it is sufficient to set $K = \lceil 5\ \%N \rceil$ to $\lceil 15\ \%N \rceil$.

Figure 5 illustrates the influence of K on the solution to (5a)–(5c). The data in Fig. 2 were again used here. By comparing to Fig. 2, we can see that our method reduces to the 1-slack and L_1 methods with $K = 1$ and N, respectively. Figure 5 shows that with different K values, models (lines) returned by solving (5a)–(5c) are different. The 1-slack model (top left) is determined by the worst residual (i.e., the largest slack), while the L_1 model (bottom) is influenced by the residuals of all data. These two settings overlook the fact that the size of outlier population is almost always *between* 1 and N and it is the joint effort of this subpopulation of the data that degrades the model quality. To capture this phenomenon, the introduction of the K parameter as in our problem formulation is therefore necessary. Figure 5(top right) also shows that with an intermediate K, more “offending” data are identified than setting $K = 1$, while unlike the $K = N$ case, the majority of them are indeed outliers.

Note that the three methods considered here (and the maxmin approach (13a)–(13b) discussed below) share a common limitation, that is, they can mistakenly remove inliers. Therefore, they are not recommended for use on severely contaminated data, that may require many iterations to clean, hence increasing the chance of removing a lot of inliers. For this reason, these methods are best used at a refinement stage after a crude RANSAC-type of outlier filtering procedure. Nevertheless, our experiments suggest that 1-slack and our method with an intermediate K can reliably handle 15 %–35 % of outliers, while the performance of the L_1 method is unstable and data-dependent.

4.2 Extension to Least Squares Regression

Another recent work that is close in spirit to our paper is by Nguyen and Welsch [19], who tailored a maxmin problem formulation of the following form for outlier removal in least squares regression:

$$\max_{\boldsymbol{\pi}} \min_{\mathbf{w}\in\mathbb{R}^d} \sum_i \pi_i \; f_i(\mathbf{w})^2 \tag{13a}$$

$$\text{s.t.} \quad \boldsymbol{\pi} \in [0,1]^N, \quad \boldsymbol{\pi}^\top \mathbf{1} = K, \tag{13b}$$

where $f_i(\mathbf{w})$ is defined as (7). To optimize (13a)–(13b), they first substitute the inner $\min_{\mathbf{w}}$ problem (a weighted least squares problem) with its closed-form solution. This allows them to convert (13a)–(13b) to a maximization problem in $\boldsymbol{\pi}$ only, which is then further reformulated as a semi-definite program (SDP) for convex optimization. The resulting SDP has $(N+1)$ unknowns and $(N+1)$ real symmetric matrices of size $(d+1)\times(d+1)$ in its linear matrix inequality constraint.

Compared to our approach, this SDP-based method is clearly less general since it restricts the cost function to be in the least-squares form, which may not be desirable for some computer vision problems [7]. Moreover, due to computational and memory restrictions inherent in SDPs, most of off-the-shelf SDP solvers can only scale to problems with dimensionality and the number of data under a few thousand. Solving the SDP reformulation of (13a)–(13b) typically requires a computational effort of $O((N+1)^2(d+1)^2)$ per iteration [16]. This makes the SDP approach computationally infeasible for large-scale problems (large N or d) such as the Structure from Motion problems considered in Sect. 5.2, where we need to estimate 3D coordinates of all the observed 2D image points as well as translation parameters of a set of cameras.

One possible remedy for the computational issue of the SDP-based method is to use our minmax formulation (5a)–(5c) to rewrite (13a)–(13b) as an *equivalent* Second-Order Cone Program (SOCP), which is known [2] to be less expensive to solve than an SDP of similar size and structure. In order to rewrite (13a)–(13b) as a minmax problem of the form (5a)–(5c), we use the fact that the $\min_{\mathbf{s},\mathbf{w}}$ and $\max_{\boldsymbol{\pi}}$ operations in (5a)–(5c) are interchangeable if $f_i(\mathbf{w})$ is continuous and convex. With such a cost function, the feasible region of $\mathbf{w}$, as specified by (5b), is convex and closed. It is also clear that the feasible region of $\mathbf{s}$ is convex and closed; and that of $\boldsymbol{\pi}$ is convex and bounded. By the minimax theorem [24, Corollary 37.3.2] (presented as Theorem 1 below), swapping $\min_{\mathbf{s},\mathbf{w}}$ and $\max_{\boldsymbol{\pi}}$ in (5a)–(5c) does not change the optimal object value.

Theorem 1 *Let C and D be non-empty closed convex sets in $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively, and let F be a continuous finite concave-convex function on $C \times D$. If either C or D is bounded, one has*

$$\inf_{\mathbf{v}\in D}\sup_{\mathbf{u}\in C} F(\mathbf{u},\mathbf{v}) = \sup_{\mathbf{u}\in C}\inf_{\mathbf{v}\in D} F(\mathbf{u},\mathbf{v}).$$

Note that the minimax theorem still applies if $f_i(\mathbf{w})$ in (5a)–(5c) is replaced with the squared cost as used in (13a)–(13b) because the squared cost is continuous and convex. Having replaced $f_i(\mathbf{w})$ with $f_i(\mathbf{w})^2$, one simply needs to set ϵ in (5a)–(5c) to 0 to get an equivalent reformulation of (13a)–(13b). In this sense (5a)–(5c) subsumes (13a)–(13b). Moreover, rewriting (13a)–(13b) as a minmax problem also allows for the application of the reformulation technique introduced in Sect. 3 to further convert it to a minimization problem of the form (10a)–(10c) with $f_i(\mathbf{w})$ replaced with $f_i(\mathbf{w})^2$. Converting this minimization problem to an SOCP is straightforward since it only involves rewriting the quadratic constraints (introduced by using a squared loss) to second order cone constraints.

If further speedup is preferred, then a linear cost function such as (7) is probably more appropriate since in this case one only needs to solve an LP of the form (10a)–(10c) with $(2N+d+1)$ unknowns. The computational complexity of solving an LP of this size is typically $O((2N+d+1)^{2.5})$ per iteration [18]. As will be shown later in the experiments of Sect. 6, the proposed LP formulation scales well to problems with up to half a million of data and unknown variables.

We have shown that it is straightforward to extend our minmax problem formulation (5a)–(5c) to outlier removal in least squares regression, and the resulting optimization problem is equivalent to but computationally much cheaper than the SDP considered in [19]. Moreover, unlike the reformulation technique adopted in [19], our strategy (Sect. 3) does not rely on a specific form of the cost function, hence has the potential to be applied to a wider range of applications.

5 K-Slack for Multiview Geometry Problems

This section demonstrates the use of our method on two representative multiview geometry problems: homograhy estimation and Structure from Motion (SfM) with known camera rotations.

5.1 Homography Estimation

In homography estimation, given keypoint matches (e.g. established by SIFT [17]): $\mathbf{u} := (x_i, y_i, 1)$ and $\mathbf{u}' := (x'_i, y'_i, 1)$

(in homogeneous plane coordinates), a 3 × 3 homography matrix $\mathbf{H}$ is estimated so that $\mathbf{u} \simeq \mathbf{H}\mathbf{u}'$. We consider a parameterization of $\mathbf{H}$ that sets the right bottom entry to 1. (See e.g. [11] for more details on such a parameterization.) Let $\mathbf{u}_i$ and $\mathbf{u}'_i$ be a pair of matched keypoints. The reprojection error with respect to an $\mathbf{H}$ can be quantified by

$$V_i(\mathbf{H}) := \max\left(\left|\mathbf{h}_1^\top\mathbf{u}'_i/\mathbf{h}_3^\top\mathbf{u}'_i - x_i\right|, \left|\mathbf{h}_2^\top\mathbf{u}'_i/\mathbf{h}_3^\top\mathbf{u}'_i - y_i\right|\right)$$

$$= \max\left(\left|\frac{(\mathbf{h}_1^\top - x_i\mathbf{h}_3^\top)\mathbf{u}'_i}{\mathbf{h}_3^\top\mathbf{u}'_i}\right|, \left|\frac{(\mathbf{h}_2^\top - y_i\mathbf{h}_3^\top)\mathbf{u}'_i}{\mathbf{h}_3^\top\mathbf{u}'_i}\right|\right), \quad (14)$$

where $\mathbf{h}_j^\top$ denotes the jth row of $\mathbf{H}$. To account for the fact that all the observed image points are in front of the cameras, the *cheirality* condition:

$$\mathbf{h}_3^\top\mathbf{u}'_i > 0, \quad (15)$$

is enforced for all i. For a given $\mathbf{H}$, datum i (a pair of keypoint matches) is an inlier if its reprojection error is bounded by a given error tolerance:

$$V_i(\mathbf{H}) \le \epsilon. \quad (16)$$

Provided the cheirality condition (15), the inlier condition (16) can be realized by constraining the errors along the x and y coordinates via

$$\left|(\mathbf{h}_1^\top - x_i\mathbf{h}_3^\top)\mathbf{u}'_i\right| \le \epsilon\mathbf{h}_3^\top\mathbf{u}'_i, \quad (17)$$

and similarly for the error along the y coordinate.

Introducing non-negative slack variables s_i and applying the reformulation techniques introduced in Sect. 3, we arrive at the following LP for outlier removal in homography estimation:

$$\min_{\alpha,\beta_i,s_i,\mathbf{H}} \ \alpha K + \sum_i \beta_i \quad (18a)$$

$$\text{s.t.} \quad \left|(\mathbf{h}_1^\top - x_i\mathbf{h}_3^\top)\mathbf{u}'_i\right| - \epsilon\mathbf{h}_3^\top\mathbf{u}'_i \le s_i, \quad \forall i \quad (18b)$$

$$\left|(\mathbf{h}_2^\top - y_i\mathbf{h}_3^\top)\mathbf{u}'_i\right| - \epsilon\mathbf{h}_3^\top\mathbf{u}'_i \le s_i, \quad \forall i \quad (18c)$$

$$-\mathbf{h}_3^\top\mathbf{u}'_i < s_i, \quad \forall i \quad (18d)$$

$$\alpha + \beta_i \ge s_i, \quad \beta_i \ge 0, \quad s_i \ge 0, \quad \forall i. \quad (18e)$$

Note that (18b) can be rewritten as linear constraints:

$$-s_i - \epsilon\mathbf{h}_3^\top\mathbf{u}'_i \le \left(\mathbf{h}_1^\top - x_i\mathbf{h}_3^\top\right)\mathbf{u}'_i \le \epsilon\mathbf{h}_3^\top\mathbf{u}'_i + s_i, \quad (19)$$

similarly, for (18c). Compared to the general form of the LP given in (10a)–(10c), the LP tailored for homography estimation is in a slightly different form. Consequently, at Step 4 of Algorithm 1, instead of solving (10a)–(10c), one needs to solve (18a)–(18e). The performance guarantee provided by Theorem 2 is nevertheless still valid here, owing to the convexity of (18a)–(18e).

5.2 Structure from Motion Estimation

The proposed method can also be applied to Structure from Motion (SfM) estimation, a fundamental problem in multiview geometry. Given tracked 2D image points, SfM infers the 3D structure (3D point positions) of the scene as well as camera motions. Here we consider SfM problems in a calibrated framework where the intrinsic camera parameters are provided. In addition, the rotation matrix (w.r.t. a reference frame) of each camera is assumed given. Our goal is to remove outliers and simultaneously estimate scene structure and camera translations in a global coordinate system.

Let $\mathbf{P}_j = [\mathbf{R}_j|\mathbf{t}_j]$ be the 3×4 camera matrix of camera j. The 3×3 rotation matrix $\mathbf{R}_j$ is known, while the translation vector $\mathbf{t}_j$ is to be estimated. Parameterizing an unknown 3D point by $\mathbf{u}_i := (x_i, y_i, z_i, 1)$ and denoting the coordinates of its corresponding observation in image j by (x_{ij}, y_{ij}), we can measure the reprojection error of a 3D point i in image j by the following cost function:

$$V(\mathbf{u}_i, \mathbf{t}_j) := \max\left(V_x(\mathbf{u}_i, \mathbf{t}_j), V_y(\mathbf{u}_i, \mathbf{t}_j)\right) \quad (20)$$

where

$$V_x(\mathbf{u}_i, \mathbf{t}_j) := \left|\frac{\mathbf{r}_{j1}^\top\mathbf{u}_i + t_{j1}}{\mathbf{r}_{j3}^\top\mathbf{u}_i + t_{j3}} - x_{ij}\right| \quad \text{and}$$

$$V_y(\mathbf{u}_i, \mathbf{t}_j) := \left|\frac{\mathbf{r}_{j2}^\top\mathbf{u}_i + t_{j2}}{\mathbf{r}_{j3}^\top\mathbf{u}_i + t_{j3}} - y_{ij}\right|$$

quantify the reprojection errors along the x and y coordinates, respectively, with $\mathbf{r}_{jk}^\top$ denoting the kth row of $\mathbf{R}_j$ and t_{jk} denoting the kth entry of $\mathbf{t}_j$. Given $\mathbf{u}_i$ and $\mathbf{t}_j$, a 2D observation in image j is an inlier if

$$V(\mathbf{u}_i, \mathbf{t}_j) \le \epsilon. \quad (21)$$

To ensure that the reconstructed 3D scene is within a certain region in front of all cameras, we can enforce the reconstruction depth to be within a positive range:

$$d_1 \le \mathbf{r}_{j3}^\top\mathbf{u}_i + t_{j3} \le d_2, \quad \forall i, j. \quad (22)$$

Given (22), one can verify (21) by checking whether the following two conditions hold simultaneously:

$$f_x(\mathbf{u}_i, \mathbf{t}_j) \le 0 \quad \text{and} \quad f_y(\mathbf{u}_i, \mathbf{t}_j) \le 0, \quad (23)$$

where the functions $f_x(\mathbf{u}_i, \mathbf{t}_j)$ and $f_y(\mathbf{u}_i, \mathbf{t}_j)$ are defined as $|(\mathbf{r}_{j1} - x_{ij}\mathbf{r}_{j3})^\top\mathbf{u}_i + t_{j1} - x_{ij}t_{j3}| - \epsilon(\mathbf{r}_{j3}^\top\mathbf{u}_i + t_{j3})$ and $|(\mathbf{r}_{j2} - y_{ij}\mathbf{r}_{j3})^\top\mathbf{u}_i + t_{j2} - y_{ij}t_{j3}| - \epsilon(\mathbf{r}_{j3}^\top\mathbf{u}_i + t_{j3})$, respectively.

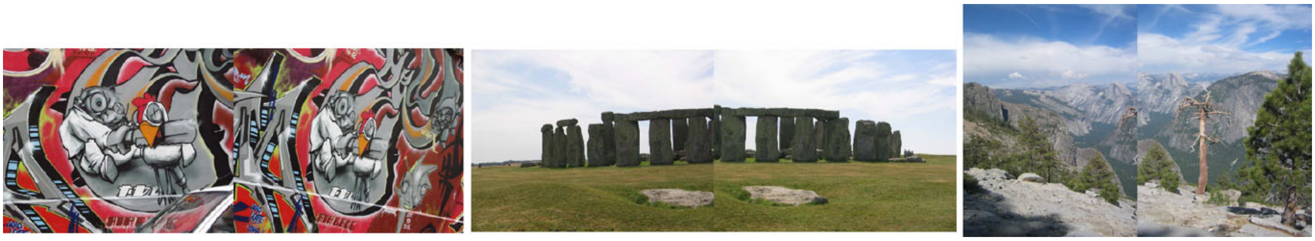


Fig. 6 Three image pairs: Graf (*left*), StoneHenge (*center*), and Mountain (*right*), used in the Homography estimation experiments of Sect. 6.1

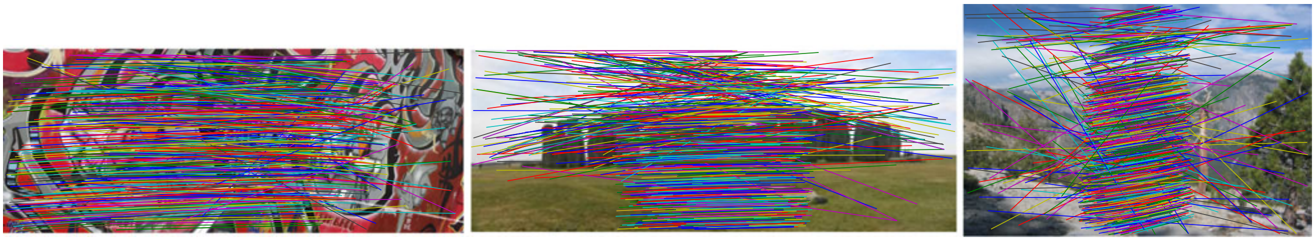


Fig. 7 SIFT keypoint matches on the three image pairs in Fig. 6. False matches are clearly present

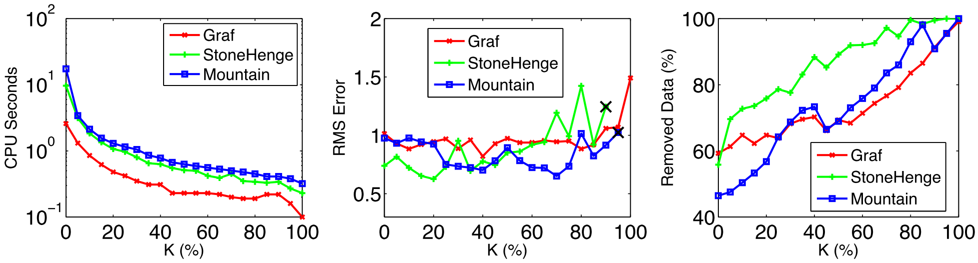


Fig. 8 The influence of K value (in percentage of N, $K = 1$ plotted as 0 $\%N$) on CPU time (*left*, on a log scale), RMS error in pixels (*center*), and the number of removed matches (in percentage of N)

Introducing non-negative slack variables s_{ij} and applying the reformulation techniques introduced in Sect. 3, we have the following LP for outlier removal in SfM estimation:

$$\min_{\alpha,\beta_{ij},s_{ij},\mathbf{u}_i,\mathbf{t}_j} \quad \alpha K + \sum_{i,j} \beta_{ij} \tag{24a}$$

$$\text{s.t.} \quad f_x(\mathbf{u}_i, \mathbf{t}_j) \leq s_{ij}, \quad \forall i, j \tag{24b}$$

$$f_y(\mathbf{u}_i, \mathbf{t}_j) \leq s_{ij}, \quad \forall i, j \tag{24c}$$

$$d_1 - \mathbf{r}_{j3}^\top \mathbf{u}_i + t_{j3} \leq s_{ij}, \quad \forall i, j \tag{24d}$$

$$\mathbf{r}_{j3}^\top \mathbf{u}_i + t_{j3} - d_2 \leq s_{ij}, \quad \forall i, j \tag{24e}$$

$$\alpha + \beta_{ij} \geq s_{ij}, \quad \forall i, j \tag{24f}$$

$$\beta_{ij} \geq 0, \quad s_{ij} \geq 0, \quad \forall i, j. \tag{24g}$$

Similar to the conversion (19), the constraints (24b) and (24c) can be realized by linear constraints.

6 Experiments

We evaluated the proposed method (K-slack, Algorithm 1) on homography estimation and SfM estimation with known camera rotations. The LPs tailored for these two tasks are given in Sect. 5. Our method was implemented in MATLAB with MOSEK LP solver.[1] All experiments were run on a machine with 2.67 GHz Intel quad core processors and 4 GB of RAM.

6.1 Outlier Removal in Homography Estimation

Our first set of experiments were conducted on keypoint correspondences for homography estimation. Figure 6 shows the image pairs[2] used in our experiments. Keypoint correspondences were established by SIFT matching (Fig. 7). The Graf data contain 437 matches, StoneHenge 928, and Mountain 1272. These data were directly fed to K-slack without outlier pre-filtering. The error tolerance ϵ was set to 2 pixels.

We investigated the performance of K-slack under the influence of various K values. Three performance measures were considered: overall CPU time required to clean the data, quality of the model returned by the last LP, and the overall number of removed matches. We measured model quality by Root Mean Square (RMS) reprojection error on the remaining clean data. Figure 8(left) shows that on all

[1] Available from http://www.mosek.com.

[2] The Graf data were obtained from http://www.robots.ox.ac.uk/~vgg and the rest from http://echoone.com/doubletake/samples.

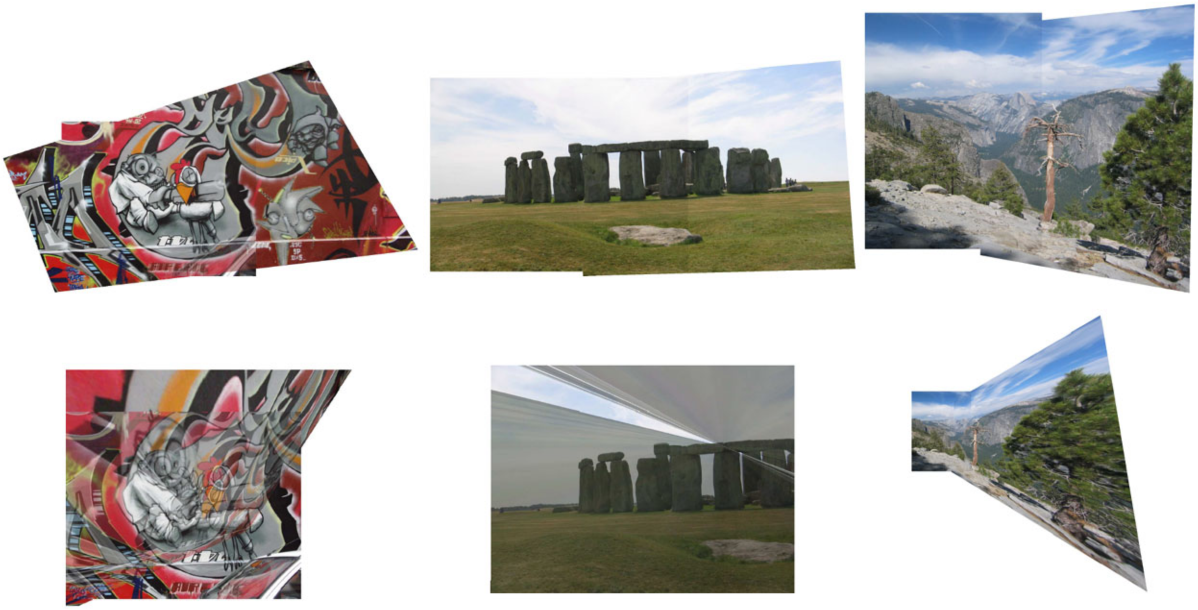


Fig. 9 The mosaics obtained by setting $K = \lceil 15\ \%N \rceil$ (*top*) versus the skewed results (*bottom*) for $K \approx N$

datasets the required CPU time reduces with increasing values of K. This is not surprising because more data (hence potentially more outliers) are removed for a larger K, therefore fewer runs of LPs are needed. This also explains why the L_1 (*resp.* 1-slack) method is fast (*resp.* slow) in general. In terms of RMS error, Fig. 8(center) shows that it maintains at a relatively low level for a wide range of K values; for K from 1 to $\lceil 65\ \%N \rceil$, the RMS errors on all image pairs are below 1 pixel. As K further increases, larger errors start to appear. Moreover, on the StoneHenge and Mountain data, as K surpasses $\lceil 90\ \%N \rceil$ and respectively, $\lceil 95\ \%N \rceil$ (marked as $\times$), K-slack fails to find a valid homography before removing all the data. Indeed, as can be seen in Fig. 8 (right), there is a general increasing trend in the number of removed data as the value of K increases. As already illustrated in Fig. 5(right), being too "aggressive" in removing data may cause disastrous model fitting results. As also evidenced in Fig. 9(bottom), large K values lead to skewed mosaicing results. Empirically, we found that to obtain a good compromise between various performance measures, it is sufficient to set $K = \lceil 5\ \%N \rceil$ to $\lceil 15\ \%N \rceil$. Figure 9(top) shows the mosaics for a sample choice of K within this range. These results are clearly more satisfactory than the skewed mosaics (bottom) given by setting $K \approx N$ ($K = N$, $\lceil 90\ \%N \rceil$, and $\lceil 95\ \%N \rceil$ for the Graf, StoneHenge, and Mountain data, respectively). Meanwhile, the speedup gained from using a $K > 1$ is also apparent (Fig. 8(left)).

6.2 Outlier Removal in SFM Estimation

We also applied our method to the more challenging SfM problem. Table 1 lists the datasets used in our experiments.

Table 1 Image sequences used in the SfM experiments, the number of cameras, 3D points (visible in at least 2 cameras), and their corresponding 2D observations

Datasets	Cameras	3D points	Observations
Cathedral	17	15300	40559
Vercingetorix	69	20549	56400
Sthlm	43	38848	223988
Alcatraz	133	41668	273468
Triumph	173	55253	317432
Eglise	85	75061	516128

The tracked image points with camera calibration information were obtained from the web.[3] We followed the procedure proposed in [22] to estimate global camera rotations, which involves applying the robust motion averaging method introduced in [6] on *relative* camera rotations between pairs of images. The relative camera rotations were estimated via the five-point method [20], followed by a bundle adjustment [30] refinement step. Both the estimation of relative rotations and the subsequent motion averaging were operated in a RANSAC framework. The 2D observations listed in Table 1 were deemed as inliers by RANSAC (with inlier threshold of 1 pixel). The aforementioned data preprocessing procedure was done by using the publicly available MATLAB implementation from the web.[3]

For all the tested methods, we set the error tolerance ϵ to 1 pixel. We also required the structure depth (c.f. Sect. 5.2) to be within [0.1, 100].

[3] www.maths.lth.se/matematiklth/personal/calle.

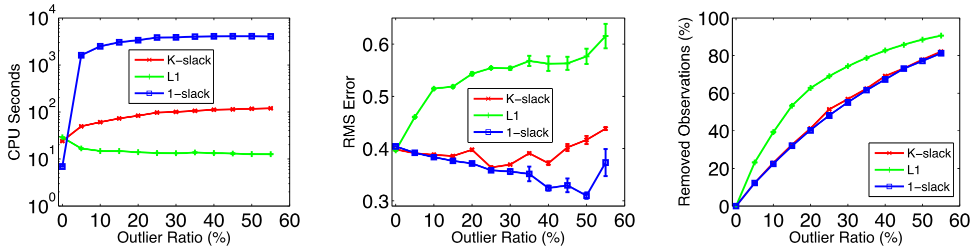


Fig. 10 Average (with standard errors) CPU time (*left*), RMS error in pixels (*center*), and the number (in percentage of N) of removed 2D observations (*right*) under the influence of different outlier ratios

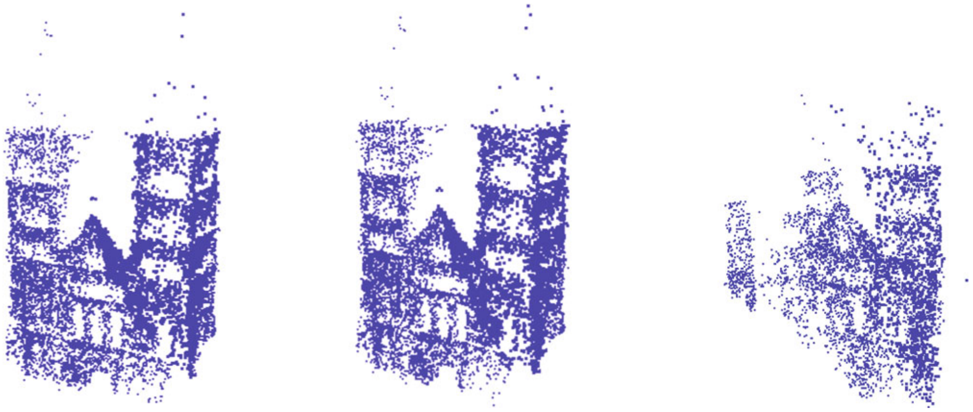


Fig. 11 3D reconstructions of the 1-slack (*left*), K-slack with $K = \lceil 10\ \%N \rceil$ (*center*), and L_1 (*right*) methods on the Cathedral data contaminated with 15 % of outliers

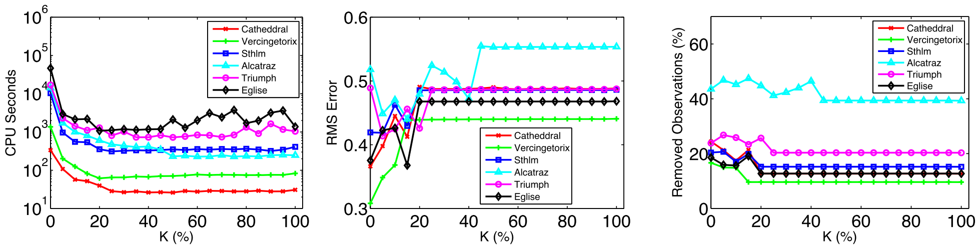


Fig. 12 The influence of K (in percentage of N, $K = 1$ plotted as 0 %N) on CPU time (*left*, on a log scale), RMS error in pixels (*center*), and the number (in percentage of N) of removed data (*right*)

6.2.1 *Performance Under Different Outlier Ratios*

To investigate the influence of outlier ratio on the performance of various methods, we applied the 1-slack, K-slack with $K = \lceil 10\ \%N \rceil$, and L_1 methods on the Cathedral data with different percentages (0 %–55 %) of outliers. We first cleaned the original data (by applying the L_1 method) to obtain a noise-free test instance with 34778 2D observations. Various fractions of randomly selected clean data were then replaced with random points within the image range to create different levels of contaminations.

Figure 10 shows the average performance of the three methods over 5 random runs. We can see (left) that on the noise-free data the three methods operate at a comparable speed. This is because all methods require only one iteration to realize that the data is clean. While on contaminated data, they exhibit clearly different runtime performance: The L_1 method is not sensitive to the outlier ratio since it runs in a one-shot fashion. The 1-slack method suffers the most from the increase in outlier ratio. It is over 2 orders of magnitude slower than the L_1 method. The impact of increasing outlier ratio on K-slack is less severe, it requires up to 4 times more CPU time than the L_1 method. Across all non-zero outlier ratios, the qualities of the L_1 models are the worst (Fig. 10, center). Figure 11 provides sample 3D structures reconstructed by the three methods from data with 15 % of outliers. As can be seen, compared to the convincing reconstructions given by the 1-slack (left) and K-slack (center) methods, the L_1 result (right) is incomprehensible. This indicates that the L_1 method is unreliable when applied on noisy data. Finally, it is not surprising to see that as more and more observations are contaminated, the number of data removed by the three methods increases (Fig. 10(right)). We can also see that the L_1 method tends to remove more data than the other two methods.

This set of experiments show that under various noise levels, K-slack with an intermediate K value performs comparably to the "conservative" 1-slack method in terms of model quality, yet achieving so in considerably (over an order of magnitude) less time.

6.2.2 *Performance with Various K*

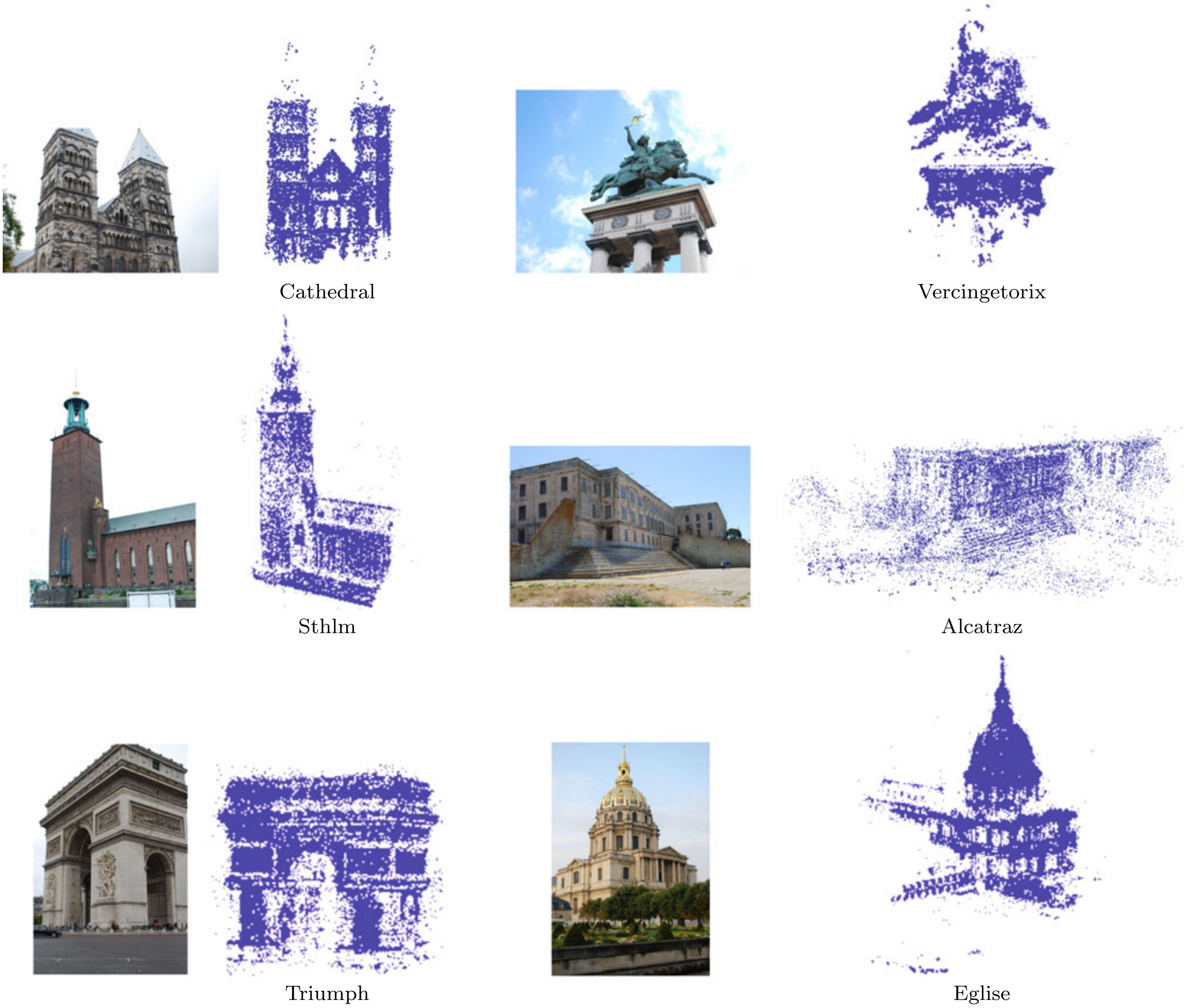
In the final set of experiments K-slack was applied on the data (Table 1) pre-cleaned by RANSAC. Figure 12 shows the performance of K-slack with various K values. As can

Table 2 Performance of various methods on the SfM data in Table 1 in terms of the number of LPs solved by each method in order to clean the data and the total number of removed 2D observations (with the corresponding number of removed 3D points in parentheses). We also report the required CPU seconds, the RMS reprojection error (in pixels) of the model returned by each outlier removal method, and the RMS reprojection error after bundle adjustment on the cleaned data

Datasets	Method	LP	Removed	CPU	RMS	Bundle RMS
Cathedral	L_1	1	5130 (689)	30.9	0.488	0.302
	1-slack	45	9908 (2632)	335.6	0.366	0.280
	K-slack	2	7082 (1427)	57.2	0.445	0.298
	RANSAC	–	–	–	1.352	0.707
Vercingetorix	L_1	1	5411 (743)	83.1	0.441	0.175
	1-slack	84	9385 (1791)	1361.6	0.308	0.152
	K-slack	2	8352 (1673)	125.1	0.368	0.157
	RANSAC	–	–	–	3.243	1.277
Sthlm	L_1	1	33983 (728)	409.8	0.486	0.377
	1-slack	111	45634 (1802)	10358.1	0.419	0.203
	K-slack	2	38161 (974)	552.2	0.463	0.364
	RANSAC	–	–	–	2.901	1.564
Alcatraz	L_1	1	107500 (3874)	246.6	0.554	0.315
	1-slack	149	119070 (6961)	14255.0	0.518	0.327
	K-slack	5	123643 (6714)	1005.6	0.470	0.305
	RANSAC	–	–	–	4.174	2.763
Triumph	L_1	1	64420 (1557)	1040.4	0.488	0.268
	1-slack	127	75563 (3545)	16982.6	0.489	0.279
	K-slack	3	82025 (3934)	1430.1	0.425	0.257
	RANSAC	–	–	–	3.914	2.117
Eglise	L_1	1	65626 (3176)	1375.0	0.468	0.197
	1-slack	197	95744 (4594)	46319.8	0.375	0.191
	K-slack	2	80784 (4308)	2139.4	0.426	0.193
	RANSAC	–	–	–	7.459	4.310

be seen in Fig. 12(left), for $K \leq \lceil 20\ \%N \rceil$, the required CPU time decreases with increasing values of K. The runtime performance of K-slack starts to stabilize for larger K values. This is because for any $K > \lceil 20\ \%N \rceil$, only 1–2 iterations are needed to remove all outliers from the pre-cleaned data, hence causing no apparent difference in runtime. However, on the Triumph and Eglise data we observed some oscillations in CPU time for K greater than $\lceil 40\ \%N \rceil$ and respectively, $\lceil 75\ \%N \rceil$. On both datasets a single iteration is sufficient to remove all outliers for these K values. We therefore believe that the oscillations are caused by varying difficulties in solving the optimization problems with different K choices. In terms of model quality, we can see (center) that the RMS errors for $K < \lceil 15\ \%N \rceil$ is noticeably lower than those of $K = N$, except on the Triumph data where the RMS error for $K = 1$ is marginally worse than that for $K = N$ (a case where the L_1 method works well). On the relatively clean datasets as used here, the influence of K on the number of removed data is small (right). The 1-slack method can actually remove slightly more data than using a larger K simply because it executes more outlier removal steps. Overall, these experiments again suggest that a $K \in [5\ \%N, 15\ \%N]$ offers a good compromise between various performance measures.

Table 2 provides a more detailed comparison between K-slack with K fixed to $\lceil 10\ \%N \rceil$ against its two extremes. Before feeding the pre-cleaned data to K-slack, we generated a baseline model for each dataset by minimizing the sum of reprojection errors (with the reprojection error computed by (20)) over the entire data. Recall that those data were classified by RANSAC as inliers (with inlier threshold of 1 pixel, same value as the error tolerance used in K-slack). Using the models returned by the outlier removal methods for initialization, we also applied bundle adjustment on the cleaned data to obtain further refined models. In Table 2 we can see that on all datasets the models obtained from model fitting on RANSAC inliers are associated with considerably larger RMS errors than those of other methods (e.g. over one order of magnitude larger on the Eglise data). This demonstrates the advantage of taking full tracks of the


Cathedral

Vercingetorix

Sthlm

Alcatraz

Triumph

Eglise

Fig. 13 3D reconstructions produced by bundle adjustment on the data cleaned by K-slack with $K = \lceil 10\,\%N \rceil$

data into account when performing outlier removal. Again, the L_1 method has the best runtime performance. It is up to 57 times faster than the 1-slack method; when compared to K-slack, its speedup is however not as significant (only up to 3 times faster). Although efficient, the L_1 models produce noticeably larger RMS errors than those of 1-slack and K-slack in all cases, except on the Triumph data where the 1-slack model is negligibly worse than the L_1 method. It is also found that better models returned by solving the LP (24a)–(24g) generally lead to better bundle adjustment results (see Bundle RMS in Table 2); as shown in Fig. 13, bundle adjustment on the data cleaned by K-slack convincingly recovers the main structures in various scenes.[4] Between the 1-slack and K-slack methods, K-slack requires much fewer runs of LPs, exhibiting substantial speedups of 83 %–95 %. More importantly, such a superior runtime performance is achieved without compromising too much the quality of final 3D reconstructions; and in fact, on 2 out of 6 datasets, K-slack produces better models. This is because besides removing outliers, K-slack also removes some "noisy" inliers, i.e., data with non-zero (but $\leq \epsilon$) residuals, hence giving lower RMS errors than the 1-slack method.

Since among the removed data there often exist inliers, it may be desirable to have an inlier restoring scheme for some applications. Noting that after the outlier removal procedure, all the camera parameters are known to us (since they are part of the model), we can recover inliers by solving a triangulation problem [9] on the 2D observations of each removed 3D point, and restore observations that have a reprojection error less than a given tolerance.

[4] The images of the scenes were obtained from [5, 22].

7 Conclusion

We proposed a novel adversarial view of the outlier removal problem that not only gives rise to a general optimization framework that unifies various previous methods, but also brings new insights into the outlier removal problem from a game-theoretic perspective. To solve the resulting mixed-integer program, we also developed an equivalent LP reformulation that significantly simplifies the process. Owing to its general formulation, our method is able to control the trade-off between reliability and speed, which is otherwise not possible using existing methods. Experiments on real image data demonstrate the superior practical performance of our method over recent approaches.

Acknowledgements This work is supported by the Australian Research Council grants DP0878801 and DP0988439. An initial version of this paper was presented at the 2011 ICCV conference [31].

Appendix A: A Theoretical Performance Bound

Theorem 2 below establishes a performance guarantee for our method. Its proof uses a known property of *active constraints*. In convex optimization, an inequality constraint $c(\mathbf{z}) \geq 0$ at a solution $\mathbf{z}$ is *active* if $c(\mathbf{z}) = 0$. The objective value achieved at a solution is determined only by its associated active constraints [3]. This property can be extended to problems that involve strictly quasi-convex functions [29].

Theorem 2 *If $f_i(\mathbf{w})$ in (4a)–(4c) is strictly quasi-convex or convex and $|\mathcal{O}| \geq K$, then the potential outlier set $\mathcal{O}$ as defined in (6) contains at least one outlier.*

Proof First, we show that the optimal objective value of (4a)–(4c) (*resp.* (5a)–(5c)) is only influenced by the potential outliers identified in $\mathcal{O}$. Toward this end, we rewrite (4a)–(4c) as

$$\min_{\mathbf{s},\mathbf{w}} \max_{\boldsymbol{\psi}\in\Psi} \; g(\boldsymbol{\psi}, \mathbf{s}) \tag{25a}$$

$$\text{s.t.} \quad f_i(\mathbf{w}) \leq \epsilon + s_i, \quad s_i \geq 0, \quad \forall i, \tag{25b}$$

where $\Psi := \{\boldsymbol{\pi} \in \{0,1\}^N \mid \boldsymbol{\pi}^\top \mathbf{1} = K\}$, and $g(\boldsymbol{\psi}, \mathbf{s}) := \boldsymbol{\psi}^\top \mathbf{s}$ is an auxiliary function. Further reformulation of (25a)–(25b) results in the following equivalent problem:

$$\min_{\mathbf{s},\mathbf{w},\delta} \; \delta \tag{26a}$$

$$\text{s.t.} \quad g(\boldsymbol{\psi}, \mathbf{s}) \leq \delta, \quad \forall \boldsymbol{\psi} \in \Psi \tag{26b}$$

$$f_i(\mathbf{w}) \leq \epsilon + s_i, \quad s_i \geq 0, \quad \forall i, \tag{26c}$$

where $\delta \in \mathbb{R}$. Let $(\mathbf{s}^*, \mathbf{w}^*, \delta^*)$ be the solution to (26a)–(26c). It is clear that δ^* must equal the sum of the K-largest entries of $\mathbf{s}^*$. By the definition of $\mathcal{O}$, $g(\boldsymbol{\psi}, \mathbf{s}^*) = \delta^*$ only if $\boldsymbol{\psi} \in \Psi^* := \{\boldsymbol{\pi} \in \{0,1\}^N \mid \boldsymbol{\pi}^\top \mathbf{1} = K, \pi_i = 0, \forall i \notin \mathcal{O}\}$. Therefore, the constraint $g(\boldsymbol{\psi}, \mathbf{s}) \leq \delta$ is inactive for all $\boldsymbol{\psi} \notin \Psi^*$. This means we can reduce Ψ in (26b) to Ψ^* without changing the optimal objective value. In this case $\pi_i = 0, \forall i \notin \mathcal{O}$, hence the data not in $\mathcal{O}$ are redundant to the optimization.

Suppose that a $\mathbf{w}$ exists with $f_i(\mathbf{w}) \leq \epsilon, \forall i \in \mathcal{O}$. This ensures a zero optimal objective value for (26a)–(26c) (hence also for (4a)–(4c) and (5a)–(5c)). This contradicts the fact that $|\mathcal{O}| \geq K$. □

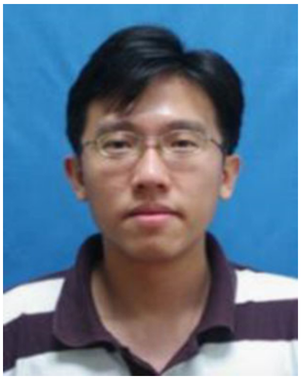
References

1. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
2. Alizadeh, F., Goldfarb, D.: Second-order cone programming. Math. Program. **95**(1), 3–51 (2003)
3. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
4. Enqvist, O., Josephson, K., Kahl, F.: Optimal correspondences from pairwise constraints. In: ICCV (2009)
5. Enqvist, O., Olsson, C., Kahl, F.: Stable structure from motion using rotational consistency. Tech. rep., Lund University (2011)
6. Govindu, V.: Robustness in motion averaging. In: ACCV (2006)
7. Hartley, R., Kahl, F.: Optimal algorithms in multiview geometry. In: ACCV (2007)
8. Hartley, R., Schaffalitzky, F.: L_∞ minimization in geometric reconstruction problems. In: CVPR (2004)
9. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2003)
10. Huber, P.J.: Robust estimation of a location parameter. Ann. Math. Stat. **35**(1), 73–101 (1964)
11. Kahl, F., Hartley, R.: Multiple-view geometry under the L_∞-norm. IEEE Trans. Pattern Anal. Mach. Intell. **30**(9), 1603–1617 (2008)
12. Ke, Q., Kanade, T.: Quasiconvex optimization for robust geometric reconstruction. In: ICCV (2005)
13. Lee, K., Meer, P., Park, R.: Robust adaptive segmentation of range images. IEEE Trans. Pattern Anal. Mach. Intell. **20**(2), 200–205 (1998)
14. Li, H.: A practical algorithm for L_∞ triangulation with outliers. In: CVPR (2007)
15. Li, H.: Consensus set maximization with guaranteed global optimality for robust geometry estimation. In: ICCV (2009)
16. Lobo, M., Vandenberghe, L., Boyd, S., Lebret, H.: Applications of second-order cone programming. Linear Algebra Appl. **284**(1–3), 193–228 (1998)
17. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
18. Monteiro, R., Adler, I.: Interior path following primal-dual algorithms. Part I: Linear programming. Math. Program. **44**(1), 27–41 (1989)
19. Nguyen, T., Welsch, R.: Outlier detection and least trimmed squares approximation using semi-definite programming. Comput. Stat. Data Anal. **54**(12), 3212–3226 (2010)
20. Nistér, D.: An efficient solution to the five-point relative pose problem. IEEE Trans. Pattern Anal. Mach. Intell. **26**(6), 756–770 (2004)
21. Nistér, D.: Preemptive RANSAC for live structure and motion estimation. Mach. Vis. Appl. **16**(5), 321–329 (2005)
22. Olsson, C., Enqvist, O.: Stable structure from motion for unordered image collections. In: Scandinavian Conference on Image Analysis (2011)

23. Olsson, C., Eriksson, A., Hartley, R.: Outlier removal using duality. In: CVPR (2010)
24. Rockafellar, R.: Convex Analysis. Princeton University Press, Princeton (1997)
25. Rousseeuw, P., Van Driessen, K.: Computing LTS regression for large data sets. Data Min. Knowl. Discov. **12**(1), 29–45 (2006)
26. Rousseeuw, P.J., Leroy, A.M.: Robust Regression and Outlier Detection. Wiley, New York (1987)
27. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn. Prentice Hall, New York (2003)
28. Seo, Y., Lee, H., Lee, S.: Outlier removal by convex optimization for L-infinity approaches. In: Advances in Image and Video Technology, pp. 203–214 (2009)
29. Sim, K., Hartley, R.: Removing outliers using the L_∞ norm. In: CVPR (2006)
30. Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment—a modern synthesis. In: Vision Algorithms: Theory and Practice, pp. 153–177 (2000)
31. Yu, J., Eriksson, A., Chin, T.J., Suter, D.: An adversarial optimization approach to efficient outlier removal. In: ICCV (2011)

Jin Yu received a Master's degree in Artificial Intelligence from the Katholieke Universiteit Leuven, Belgium in 2004 and a Ph.D. in Machine Learning from The Australian National University in 2010. She is currently a senior Data Scientist with EMC Greenplum and an Adjunct Lecturer with the School of Computer Science at The University of Adelaide, Australia. Her main research interests are in stochastic learning, nonsmooth optimisation, and robust model fitting for Machine Learning and Computer Vision problems.

Anders Eriksson received his M.Sc. degree in Electrical Engineering in 2000 and his Ph.D. in Mathematics in 2008, both from Lund University, Sweden. His research are a include optimization theory and numerical methods applied to the fields of Computer Vision and Machine Learning. He is currently a senior Research Associate at The University of Adelaide, Australia.



Tat-Jun Chin received a B.Eng. in Mechatronics Engineering from Universiti Teknologi Malaysia (UTM) in 2003, and subsequently in 2007 a Ph.D. in Computer Systems Engineering from Monash University, Victoria, Australia. He was a Research Fellow at the Institute for Infocomm Research (I2R) in Singapore from 2007–2008. Since 2008 he is a Lecturer at The University of Adelaide, South Australia. His research interests include robust estimation and statistical learning methods in Computer Vision.

David Suter received a B.Sc. degree in applied mathematics and physics (The Flinders University of South Australia 1977), a Grad. Dip. Comp. (Royal Melbourne Institute of Technology 1984), and a Ph.D. in computer science (La Trobe University, 1991). He was a Lecturer at La Trobe from 1988 to 1991; and a Senior Lecturer (1992), Associate Professor (2001), and Professor (2006–2008) at Monash University, Melbourne, Australia. Since 2008 he has been a professor in the School of Computer Science, The University of Adelaide. He is head of the School of Computer Science. He served on the Australian Research Council (ARC) College of Experts from 2008–2010. He is on the editorial boards of International Journal of Computer Vision and the Journal of Mathematical Imaging and Vision. He has previously served on the editorial boards of Machine Vision and Applications and the International Journal of Image and Graphics. He was General co-Chair of the Asian Conference on Computer Vision (Melbourne 2002) and is currently co-Chair of the IEEE International Conference on Image Processing (ICIP2013).