

Efficient Globally Optimal Consensus Maximisation with Tree Search

Tat-Jun Chin*, Pulak Purkait*, Anders Eriksson[†] and David Suter*

School of Computer Science, The University of Adelaide*

School of Electrical Engineering and Computer Science, Queensland University of Technology[†]

Abstract

Maximum consensus is one of the most popular criteria for robust estimation in computer vision. Despite its widespread use, optimising the criterion is still customarily done by randomised sample-and-test techniques, which do not guarantee optimality of the result. Several globally optimal algorithms exist, but they are too slow to challenge the dominance of randomised methods. We aim to change this state of affairs by proposing a very efficient algorithm for global maximisation of consensus. Under the framework of LP-type methods, we show how consensus maximisation for a wide variety of vision tasks can be posed as a tree search problem. This insight leads to a novel algorithm based on A* search. We propose efficient heuristic and support set updating routines that enable A* search to rapidly find globally optimal results. On common estimation problems, our algorithm is several orders of magnitude faster than previous exact methods. Our work identifies a promising solution for globally optimal consensus maximisation¹.

1. Introduction

Maximum consensus is one of the most popular robust criteria for geometric estimation problems in computer vision. Given a set of measurements $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$, the criterion aims to find the estimate θ that agrees with as many of the data as possible (i.e., the inliers) up to a threshold ϵ

$$\begin{aligned} & \max_{\theta, \mathcal{I} \subseteq \mathcal{X}} |\mathcal{I}| \\ & \text{subject to } r_i(\theta) \leq \epsilon \quad \forall \mathbf{x}_i \in \mathcal{I}, \end{aligned} \quad (1)$$

where $r_i(\theta)$ is the residual of \mathbf{x}_i . For example, in triangulation we wish to estimate the 3D point θ seen in N views, where \mathcal{X} contains the 2D observations of the point. The residual $r_i(\theta)$ is the reprojection error in the i -th view.

Random sample consensus (RANSAC) [7] has been the dominant approach. The method randomly draws p -tuples from \mathcal{X} , where p is the minimum number of data to instantiate θ . The consensus score $|\mathcal{I}|$ of each sample θ is obtained,

and the θ with the highest score and its consensus set \mathcal{I} are returned. A probabilistic bound on the number of samples required can be derived as a stopping criterion.

A major shortcoming of randomised methods such as RANSAC is the lack of absolute certainty that the obtained solution is optimal, or indeed whether it represents a satisfactory approximation at all. A less recognised fact is that even if all $\binom{N}{p}$ subsets are examined, we may not find the globally optimal solution θ^* to (1), since θ^* does not generally correspond to an estimate from a p -tuple (see below).

Solving (1) exactly is computationally challenging. Several authors proposed methods based on branch-and-bound (BnB) [13, 21]. However, BnB is typically slow, especially if θ is high-dimensional. In fact, RANSAC is suggested to suboptimally preprocess the data (which may cause genuine inliers to be discarded), before BnB is invoked to refine the result. More fundamentally, the BnB methods are specialised for linear residuals. For many vision problems, this entails linearising $r_i(\theta)$ and adopting algebraic residuals which are not geometrically meaningful.

It has been proven [15, 5] that for various estimation tasks, θ^* can be found as the solution on a subset of \mathcal{X} of size d , where $d \geq p$ and $d \ll N$ (the actual value of d depends on the particular problem). Both works proposed to find θ^* by exhaustively searching over all $\binom{N}{d}$ subsets of \mathcal{X} . Although the number of subsets is polynomial w.r.t. N , in realistic problems the number is impractically large. Olsson et al. [15] also proposed using RANSAC with an optimality verification step. However, the fact remains that an enormous number of subsets may need to be sampled.

Due to the much greater computational expense, currently available global methods are not competitive against RANSAC and its variants. In this paper, we make significant progress towards solving (1) exactly and efficiently. Leveraging on the framework of LP-type methods [18, 14], we show how maximising consensus can be casted as a tree search problem. We then propose an algorithm based on A* search [8] to traverse the tree. Similar in spirit to [15, 5], we aim to find the optimal data subset. However, instead of sampling or enumerating the subsets, our algorithm deterministically locates the best subset. The A* technique ensures that only a tiny fraction of available subsets need

¹Source code is available on the authors' homepage.

to be explored. Despite its combinatorial nature, our algorithm is fast - on several common estimation problems, our algorithm is orders of magnitude faster than previous exact methods for maximum consensus. Further, our method does not require linearising the residual.

1.1. Related work

L_∞ minimisation (minmax problem) is well established in the context of geometric estimation [10, 11]. The task is to find the estimate θ that minimises the largest residual

$$\min_{\theta} \max_i r_i(\theta). \quad (2)$$

Whilst (2) is inherently non-robust since it effectively fits θ on the outliers, it contains a single (global) minimum, provided $r_i(\theta)$ is strictly quasiconvex and the data is nondegenerate. Algorithms based on bisection and other schemes [6] have been proposed to solve (2) exactly.

Since the largest residuals are contributed by the outliers, one can construct an outlier removal scheme where (2) is first solved and the data corresponding to the largest residuals are removed [19, 16, 20]. The fitting and removal steps can be conducted iteratively until the maximum error is below a threshold ϵ . There is no guarantee, however, that the remaining data is the largest possible consensus set, since genuine inliers may also be removed during the iterations.

If, instead of the largest, we minimise the j -th largest residual ($j \leq N$), the outliers can be ignored and a robust estimate can be obtained. However, the problem is now much harder since multiple local minima exist. Ke and Kanade [11] extended their algorithm to approximately minimise the j -th largest residual. Matoušek proposed a method [14] to examine *all* the minima of the problem to find the global minimiser. Li [12] pioneered the usage of Matoušek's method to conduct robust triangulation. It is provable that the time complexity of Matoušek's procedure is a p -th order polynomial on the number of outliers. However, on typical-sized problems the search is painfully slow.

In general, knowing in advance the correct value of j (i.e., the number of inliers) can be non-trivial. On the other hand setting the inlier threshold ϵ in (1) is arguably easier since we usually have an idea of how far the inliers can deviate (assuming geometrically meaningful residual functions are employed). For example, the reprojection error of inliers in triangulation is typically within a few pixels.

2. Background

2.1. Solvable problems

We first define the type of problems solvable by our maximum consensus algorithm. We require that the residual function $r_i(\theta)$ be pseudoconvex. This is known to include many common applications [17]. Examples are as follows.

Linear regression The linear relation is defined by vector $\theta \in \mathbb{R}^M$ and the residual function is defined as

$$r_i(\theta) = |b_i - \theta^T \mathbf{a}_i|, \quad (3)$$

where $\mathbf{x}_i = [\mathbf{a}_i^T \ b_i]^T$. Here $r_i(\theta)$ is convex, which is a stricter condition than pseudoconvexity.

Triangulation We wish to estimate the 3D point θ seen in N views. The i -th image point and camera matrix are respectively \mathbf{x}_i and $\mathbf{P}_i \in \mathbb{R}^{3 \times 4}$. The reprojection error is

$$r_i(\theta) = \frac{\|(\mathbf{P}_{i,1:2} - \mathbf{x}_i \mathbf{P}_{i,3}) \tilde{\theta}\|}{\mathbf{P}_{i,3} \tilde{\theta}}, \quad (4)$$

where $\tilde{\theta} = [\theta^T \ 1]^T$, $\mathbf{P}_{i,1:2}$ is the first-two rows of \mathbf{P}_i , and $\mathbf{P}_{i,3}$ is the third row of \mathbf{P}_i . It is known that (4) is pseudoconvex. The additional constraint $\mathbf{P}_{i,3} \tilde{\theta} > 0$ must be imposed such that the 3D point lies in front of the cameras.

Homography fitting Given a set of point matches $\mathcal{X} = \{(\mathbf{u}_i, \mathbf{u}'_i)\}_{i=1}^N$ across two views, we wish to estimate the homography $\theta \in \mathbb{R}^{3 \times 3}$ that aligns the points. The residual is

$$r_i(\theta) = \frac{\|(\theta_{1:2} - \mathbf{u}'_i \theta_3) \tilde{\mathbf{u}}_i\|}{\theta_3 \tilde{\mathbf{u}}_i}, \quad (5)$$

where $\tilde{\mathbf{u}}_i = [\mathbf{u}_i^T \ 1]^T$, $\theta_{1:2}$ is the first-two rows of θ , and θ_3 is the third row of θ . It has been established that (5) is pseudoconvex. Note the resemblance of (5) to (4).

Many other estimation problems in computer vision naturally involve pseudoconvex residuals, e.g., camera resectioning, SfM with known rotations. In fact, Olsson et al. showed that all the problems in [10] are pseudoconvex.

2.2. LP-type problems

LP-type problems can be regarded as generalisations of linear programming (LP) problems [18]. They are intimately connected to minmax problems such as (2). In particular, it has been established that the minmax of a set of pseudoconvex functions is an LP-type problem [2, 6]. This implies the satisfaction of several key properties [1].

First, we define $f(\mathcal{X})$ as the solution (the minimal objective value) of the minmax problem (2) on data \mathcal{X} , and $\theta(\mathcal{X})$ as the corresponding globally optimal estimate.

Property 1 (Monotonicity) For every two sets $\mathcal{P} \subseteq \mathcal{Q} \subseteq \mathcal{X}$, we can establish the inequality $f(\mathcal{P}) \leq f(\mathcal{Q}) \leq f(\mathcal{X})$.

Property 2 (Locality) For every two sets $\mathcal{P} \subseteq \mathcal{Q} \subseteq \mathcal{X}$ and every $\mathbf{x} \in \mathcal{X}$, if $f(\mathcal{P}) = f(\mathcal{Q}) = f(\mathcal{P} \cup \{\mathbf{x}\})$, then we can establish the equality $f(\mathcal{P}) = f(\mathcal{Q} \cup \{\mathbf{x}\})$.

The concept of *basis* is integral to LP-type problems.

Definition 1 (Basis) A basis \mathcal{B} is a subset of \mathcal{X} whereby every proper subset of \mathcal{B} has a strictly smaller value of f than \mathcal{B} itself, i.e., if $\mathcal{A} \subset \mathcal{B}$ then $f(\mathcal{A}) < f(\mathcal{B})$.

Definition 2 (Combinatorial dimension) The combinatorial dimension is the largest possible size for a basis. For pseudoconvex problems where θ lies in a p -dimensional domain, the combinatorial dimension is $p + 1$ [6].

Definition 3 (Violation set, coverage and level) The violation set $\mathcal{V}(\mathcal{B})$ of a basis $\mathcal{B} \subseteq \mathcal{X}$ contains the data from \mathcal{X} whose residuals are greater than $f(\mathcal{B})$ when evaluated at $\theta(\mathcal{B})$. The coverage $\mathcal{C}(\mathcal{B})$ of \mathcal{B} is the complement of $\mathcal{V}(\mathcal{B})$. The level $l(\mathcal{B})$ of \mathcal{B} is the size of $\mathcal{V}(\mathcal{B})$.

Definition 4 (Support set) The level-0 basis for a dataset \mathcal{X} is called the support set of \mathcal{X} .

Solving (2) amounts to finding the support set of \mathcal{X} ; Fig. 1 illustrates. Assuming \mathcal{X} is non-degenerate², it has only one support set (cf. the minmax problem has a single minimum).

Consider modifying (2) such that the j -th largest residual is minimised

$$\min_{\theta} r_{(j)}(\theta), \quad (6)$$

where $r_{(1)}(\theta), \dots, r_{(N)}(\theta)$ are the sorted residuals given θ . Defining $k := N - j$, we can rewrite (6) as

$$\min_{\mathcal{B}} f(\mathcal{B}), \quad \text{s.t. } l(\mathcal{B}) = k \quad (7)$$

i.e., we seek the level- k basis \mathcal{B} whose estimate $\theta(\mathcal{B})$ gives the smallest $f(\mathcal{B})$ value. By definition all level- k bases will have j residuals less than or equal to $f(\mathcal{B})$ at $\theta(\mathcal{B})$. To solve (7) for $k > 0$, we describe Matoušek’s method [14].

Definition 5 (Basis adjacency) Two bases \mathcal{B} and \mathcal{B}' of respectively levels k and $k + 1$ are adjacent if $\mathcal{V}(\mathcal{B}') = \mathcal{V}(\mathcal{B}) \cup \{\mathbf{x}\}$ for some $\mathbf{x} \in \mathcal{B}$.

Property 3 (Existence of basis path) There exists a path from the level-0 basis to every level- k basis ($k > 0$) by following adjacent bases. See [14] for the proof.

The above property implies that the set of bases can be arranged in a tree structure, where two bases are linked by an edge if they are adjacent. Matoušek proposed to generate the tree up to the k -th level (by iteratively removing data from bases and refitting to obtain adjacent bases in the next level), such that all level- k bases can be examined; Fig. 2(a) illustrates. Further, he also proved that the number of level- k bases is bounded above by $\mathcal{O}((k + 1)^p)$. In practice, however, the number of bases can be very high, and the method is suitable only for a small number of outliers (one or two)³.

²Else, data perturbation methods can be applied [14]. Henceforth, for brevity we assume that all input data \mathcal{X} is non-degenerate

³As the title of [14] suggests, only a few outliers can be handled.

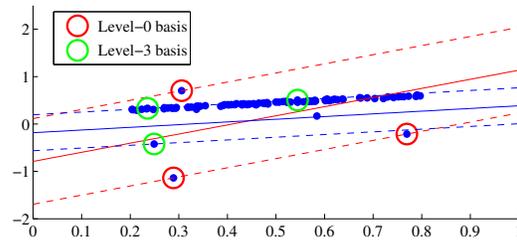


Figure 1. Illustrating the concept of basis on linear regression with $\theta \in \mathbb{R}^2$ ($p = 2$). Finding the level-0 basis and its corresponding estimate (red solid line) solves (2) on \mathcal{X} . A level-3 basis is also shown; observe that 3 points violate the corresponding estimate. For the linear regression problem (3), all bases are of size $p + 1$.

3. Consensus maximisation as tree search

We are now ready to describe our novel algorithm for solving the maximum consensus problem (1).

Definition 6 (Feasibility) A basis \mathcal{B} is feasible if $f(\mathcal{B}) \leq \epsilon$. Similarly, a set $\mathcal{Q} \subseteq \mathcal{X}$ is feasible if $f(\mathcal{Q}) \leq \epsilon$.

We can rewrite (1) as the following related problem

$$\min_{\mathcal{B}} l(\mathcal{B}), \quad \text{s.t. } f(\mathcal{B}) \leq \epsilon. \quad (8)$$

In words, we seek the lowest level basis (i.e., exclude as few data as possible) that is feasible. Given an optimal \mathcal{B}^* for (8), the maximum consensus set \mathcal{I}^* is simply $\mathcal{C}(\mathcal{B}^*)$.

To find \mathcal{B}^* , we can traverse Matoušek’s tree structure using breadth-first search (BFS), i.e., test for feasibility and expand (generate adjacent bases of) all the bases at a particular level, before going to the next level. Such a search regime guarantees that the first feasible basis found is the shallowest (has the lowest level). Fig. 2(b) depicts the idea.

Assume that \mathcal{B}^* occurs at level k . In contrast to solving (7) with the same k , the BFS algorithm does not attempt to generate all level- k bases. However, it must still generate all the bases at the previous levels, thus BFS is not efficient in practice. Nonetheless, considerable improvements to this basic scheme can be made, as we describe in the following. For quick reference, Alg. 1 summarises our method.

3.1. Avoiding repeated bases

Starting from the root node (level-0 basis), there are multiple paths for arriving at the same level- k basis. This corresponds to the fact that as long as the same set of data are removed, the order of the removal is not important. Significant redundancy thus exists in the tree. This characteristic was not exploited in Matoušek’s original method. Note that for the tree in Fig. 2, the redundancy has been removed.

In our algorithm, a basis \mathcal{B} (and the branch leading from it) are ignored if \mathcal{B} has been seen previously (Line 10 in

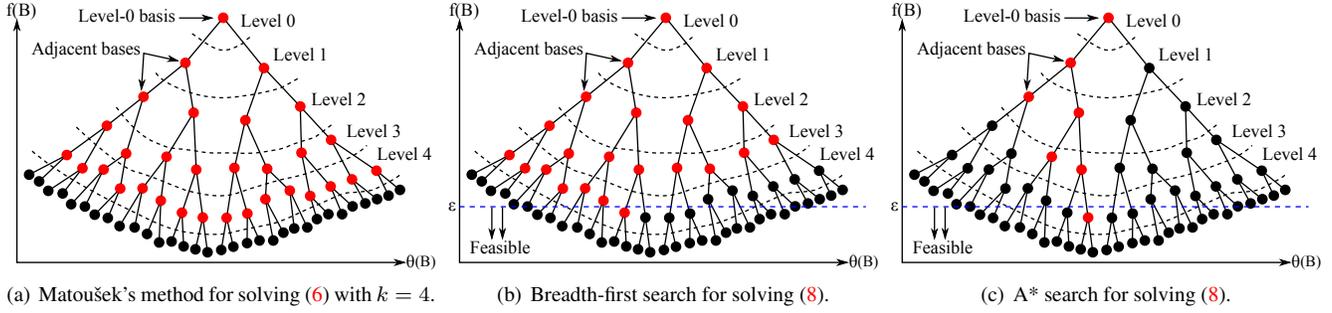


Figure 2. Tree traversal for solving (6) (panel a) and (8) (panels b and c) on a problem with $p = 1$. The ordinate is $f(\mathcal{B})$ while the abscissa is $\theta(\mathcal{B})$. Nodes in red are bases expanded in the respective algorithms. In Matoušek's method, all level- k bases (here, $k = 4$) must be tested, since there are multiple local minima. In panels b and c, the two different globally optimal solutions found are equally good w.r.t. (8).

Algorithm 1 A* search for consensus maximisation

- 1: $\mathcal{B} \leftarrow$ Support set of \mathcal{X} .
 - 2: Insert \mathcal{B} with priority $e(\mathcal{B})$ into queue q .
 - 3: Initialise hash table T to null.
 - 4: **while** q is not empty **do**
 - 5: Retrieve from q the \mathcal{B} with the lowest $e(\mathcal{B})$.
 - 6: **if** $f(\mathcal{B}) \leq \epsilon$ **then**
 - 7: Exit with $\mathcal{C}(\mathcal{B})$ as the maximum consensus set.
 - 8: **end if**
 - 9: **for each** $\mathbf{x} \in \mathcal{B}$ **do**
 - 10: **if** indices of $\mathcal{V}(\mathcal{B}) \cup \{\mathbf{x}\}$ do not exist in T **then**
 - 11: Hash indices of $\mathcal{V}(\mathcal{B}) \cup \{\mathbf{x}\}$ into T .
 - 12: $\mathcal{B}' \leftarrow$ Support set of $[\mathcal{C}(\mathcal{B}) \setminus \{\mathbf{x}\}]$.
 - 13: Insert \mathcal{B}' with priority $e(\mathcal{B}')$ into q .
 - 14: **end if**
 - 15: **end for**
 - 16: **end while**
 - 17: Return error (no feasible solution).
-

Alg. 1). Since it is possible for two equal bases to have different coverages, two bases \mathcal{B}_1 and \mathcal{B}_2 are declared the same only if $\mathcal{C}(\mathcal{B}_1) = \mathcal{C}(\mathcal{B}_2)$, or equivalently, if $\mathcal{V}(\mathcal{B}_1) = \mathcal{V}(\mathcal{B}_2)$. We propose to hash the violation set of seen bases into a table. Let $\mathcal{Q} \subseteq \mathcal{X}$ be an arbitrary subset. The *indices* of the data in \mathcal{Q} are sorted to yield an integer sequence. We convert the integers to ASCII characters and concatenate them into a string. The string is then hashed into a table via standard techniques to facilitate constant time repetition detections. For example, the set $\mathcal{Q} = \{\mathbf{x}_{71}, \mathbf{x}_{10}, \mathbf{x}_5, \mathbf{x}_4, \mathbf{x}_{28}\}$ is converted to the string 0405102871 and hashed. This simple idea brings significant computational savings.

3.2. Informed search with A* algorithm

Our approach is based on the A* algorithm, which is an informed search method used extensively in pathfinding [8]. Whilst BFS chooses the shallowest basis in the queue to test and expand, A* chooses the basis with the lowest *evaluation*

value e (Step 5 in Alg. 1), which is defined as

$$e(\mathcal{B}) = l(\mathcal{B}) + h(\mathcal{B}), \quad (9)$$

where h is a *heuristic*. Intuitively, $h(\mathcal{B})$ estimates the number of data to be removed from $\mathcal{C}(\mathcal{B})$ to make it feasible. Bases with low e are shallow and close to feasibility. Expanding according to e thus accelerates the search; see Fig. 2(c). Note that if $h(\mathcal{B}) = 0$ for all \mathcal{B} , A* reduces to BFS.

An immediate question is whether A* is optimal, i.e., *does it always find the shallowest feasible basis?*

Definition 7 (Admissibility) A heuristic h is *admissible* if

$$h(\mathcal{B}) \geq 0 \text{ and } h(\mathcal{B}) \leq h^*(\mathcal{B}), \quad (10)$$

where $h^*(\mathcal{B})$ is the *minimum number of data that must be removed from $\mathcal{C}(\mathcal{B})$ to make the data feasible*. Implied by the above is that $h(\mathcal{B}) = 0$ if \mathcal{B} is feasible.

Note that calculating $h^*(\mathcal{B})$ amounts to solving exactly the original maximum consensus problem on $\mathcal{C}(\mathcal{B})$.

Theorem 1 A* is optimal if h is admissible.

Proof Following [8], we need to show that when a suboptimal feasible basis \mathcal{B}^\dagger exists in the queue, it will not be chosen and tested *before* an optimal basis \mathcal{B}^* exists in the queue. Let \mathcal{B} be another basis in the queue on the path leading to \mathcal{B}^* (such a basis \mathcal{B} always exists in a tree structure, e.g., the root basis is one). The following can be established

$$\begin{aligned} e(\mathcal{B}^\dagger) &= l(\mathcal{B}^\dagger) && \text{(since } \mathcal{B}^\dagger \text{ is feasible)} \\ &> l(\mathcal{B}^*) && \text{(since } \mathcal{B}^\dagger \text{ is suboptimal)} \\ &= l(\mathcal{B}) + h^*(\mathcal{B}) && \text{(since } \mathcal{B} \text{ leads to } \mathcal{B}^*) \\ &\geq l(\mathcal{B}) + h(\mathcal{B}). && \text{(since } h \text{ is admissible)} \end{aligned} \quad (11)$$

The above result shows that \mathcal{B} which lies on the path to \mathcal{B}^* will always have a higher priority than \mathcal{B}^\dagger to be tested and expanded. Therefore A* will always find \mathcal{B}^* before \mathcal{B}^\dagger . ■

If the current lowest e value in the queue is given by more than one basis, we choose the \mathcal{B} with the lowest $f(\mathcal{B})$ first. This tie breaking does not affect the optimality of A*.

3.3. Heuristic for consensus maximisation

Our heuristic is inspired by previous outlier removal and reinsertion techniques [19, 16, 20] but with significant modifications. We call our heuristic h_{ins} and prove that it is applicable under A^* . To calculate $h_{ins}(\mathcal{B})$, let

$$\mathcal{O} = \{\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_M\}, \quad f(\mathcal{B}_m) > \epsilon \quad \forall m \quad (12)$$

contain the sequence of support sets that must be recursively removed from $\mathcal{C}(\mathcal{B})$ to make $\mathcal{C}(\mathcal{B})$ feasible, where

$$\mathcal{B}_m = \text{Support set of } [\mathcal{C}(\mathcal{B}) \setminus \{\mathcal{B}_1, \dots, \mathcal{B}_{m-1}\}] \text{ for } m > 1$$

and $\mathcal{B}_1 = \mathcal{B}$. As an example, Fig. 1 shows the first-2 support sets of such a sequence. Let \mathcal{F} be the remaining data, i.e.,

$$\mathcal{F} = \mathcal{C}(\mathcal{B}) \setminus \mathcal{O}, \quad \text{where } f(\mathcal{F}) \leq \epsilon. \quad (13)$$

We then attempt to insert the individual data from \mathcal{O} one-by-one into \mathcal{F} . If an insertion makes \mathcal{F} infeasible, the support set of the enlarged \mathcal{F} is removed and the heuristic $h_{ins}(\mathcal{B})$ is incremented by one; else if the enlarged \mathcal{F} remains feasible, the insertion is made permanent with no change to the heuristic. Alg. 2 gives a formal description.

Theorem 2 h_{ins} is admissible.

Proof That h_{ins} is nonnegative is obvious. Let $\mathcal{C}(\mathcal{B}) = \mathcal{F}^* \cup \mathcal{O}^*$, where \mathcal{F}^* is the largest feasible subset of $\mathcal{C}(\mathcal{B})$. \mathcal{O}^* is thus the smallest subset that must be removed from $\mathcal{C}(\mathcal{B})$ to achieve feasibility, i.e., $h^*(\mathcal{B}) = |\mathcal{O}^*|$. For intuition, we refer to \mathcal{F}^* and \mathcal{O}^* as “true” inliers and outliers.

Let \mathbf{x} be a datum from \mathcal{O} , and \mathcal{B}' the support set of $[\mathcal{F} \cup \{\mathbf{x}\}]$. If \mathcal{B}' is infeasible, then by Property 1 (monotonicity)

$$f(\mathcal{F}^* \cup \mathcal{B}') \geq f(\mathcal{B}') > \epsilon, \quad (14)$$

i.e., \mathcal{B}' must contain at least one true outlier. The removal of \mathcal{B}' from $\mathcal{F} \cup \{\mathbf{x}\}$ thus removes at least one true outlier from $\mathcal{F} \cup \{\mathbf{x}\}$ (note that the remaining set $\mathcal{F} \leftarrow \mathcal{F} \cup \{\mathbf{x}\} \setminus \mathcal{B}'$ is always feasible). There are thus at most $|\mathcal{O}^*|$ infeasible support set removals. Since each removal increments $h_{ins}(\mathcal{B})$ by one, its final value cannot be greater than $h^*(\mathcal{B})$.

Note also that in general \mathcal{O}^* is not a subset of \mathcal{O} and the proof above does not assume that it is. ■

3.4. Early termination

Define $g(\mathcal{B})$ as an upper bound on the number of data that must be removed from $\mathcal{C}(\mathcal{B})$ to make $\mathcal{C}(\mathcal{B})$ feasible. Coupled with the heuristic function, we can thus establish

$$h(\mathcal{B}) \leq h^*(\mathcal{B}) \leq g(\mathcal{B}). \quad (15)$$

Given $g(\mathcal{B})$, it is possible to terminate Alg. 1 early. Specifically, if $h(\mathcal{B}) = g(\mathcal{B})$ for the \mathcal{B} currently under test in Line 6, we can exit regardless of whether \mathcal{B} is feasible. The maximum consensus set \mathcal{I}^* is simply \mathcal{F} as defined in (13).

Algorithm 2 Calculation of $h_{ins}(\mathcal{B})$.

```

1: If  $f(\mathcal{B}) \leq \epsilon$ , return 0.
2:  $\mathcal{O} \leftarrow \emptyset$ .
3: while  $f(\mathcal{B}) > \epsilon$  do
4:    $\mathcal{O} \leftarrow \mathcal{O} \cup \mathcal{B}$ .
5:    $\mathcal{B} \leftarrow$  Support set of  $[\mathcal{C}(\mathcal{B}) \setminus \mathcal{B}]$ .
6: end while
7:  $h_{ins} \leftarrow 0$ ,  $\mathcal{F} \leftarrow \mathcal{C}(\mathcal{B})$ .
8: for each  $\mathcal{B} \in \mathcal{O}$  do
9:   for each  $\mathbf{x} \in \mathcal{B}$  do
10:     $\mathcal{B}' \leftarrow$  Support set of  $[\mathcal{F} \cup \{\mathbf{x}\}]$ .
11:    if  $f(\mathcal{B}') \leq \epsilon$  then
12:       $\mathcal{F} \leftarrow \mathcal{F} \cup \{\mathbf{x}\}$ .
13:    else
14:       $h_{ins} \leftarrow h_{ins} + 1$ .
15:       $\mathcal{F} \leftarrow \mathcal{F} \cup \{\mathbf{x}\} \setminus \mathcal{B}'$ .
16:    end if
17:   end for
18: end for
19: Return  $h_{ins}$ .
```

Theorem 3 A^* is optimal with early termination.

Proof The value given by

$$e^*(\mathcal{B}) = l(\mathcal{B}) + h^*(\mathcal{B}) \quad (16)$$

is the minimum number of data that must be removed from \mathcal{X} to achieve feasibility, given that we have removed $\mathcal{V}(\mathcal{B})$. Comparing with (9), it is clear that $e(\mathcal{B}) \leq e^*(\mathcal{B})$.

If the current \mathcal{B} has $h(\mathcal{B}) = g(\mathcal{B})$, then from (15), $e(\mathcal{B}) = e^*(\mathcal{B})$. Since we search the tree according to the A^* method, $e(\mathcal{B}) \leq e(\mathcal{B}')$ where \mathcal{B}' is any other basis in the queue. Thus $e^*(\mathcal{B}) \leq e(\mathcal{B}') \leq e^*(\mathcal{B}')$, i.e., the path to the shallowest feasible basis must pass through \mathcal{B} . ■

We propose the following upper bound

$$g_{rem}(\mathcal{B}) = |\mathcal{C}(\mathcal{B}) \setminus \mathcal{F}| = |\mathcal{O}|, \quad (17)$$

which can be obtained as a by-product of Alg. 2.

Theorem 4 $g_{rem}(\mathcal{B}) \geq h^*(\mathcal{B})$.

Proof Since \mathcal{F} is feasible, then $|\mathcal{F}| \leq |\mathcal{F}^*|$ where \mathcal{F}^* is the largest feasible subset of $\mathcal{C}(\mathcal{B})$. This implies that $|g_{rem}(\mathcal{B})| \geq |\mathcal{O}^*| = h^*(\mathcal{B})$, where $\mathcal{O}^* = \mathcal{C}(\mathcal{B}) \setminus \mathcal{F}^*$. ■

4. Efficient support set updating

It is clear that the overall efficiency of our algorithm hinges on the efficient calculation of support sets. This amounts to solving the minmax problem (2) for the relevant

data subsets. Fortunately, except the initialisation in Step 1 of Alg. 1 where the problem needs to be solved from scratch on all \mathcal{X} , all other support set computations merely requires updating a known support set after a small number of data are removed (Step 12 in Alg. 1 and Step 5 in Alg. 2) or inserted (Step 10 in Alg. 2). Thus, our method can be made very fast if the update routine is implemented carefully.

Various schemes can be used to calculate support sets for pseudoconvex residuals, e.g., generalised LP [3], bisection [10, 11] or pseudoconvex programming [6]. In fact, nonlinear (iterative) optimisation can be used to directly seek the globally optimal solution [17]. To this end, rewrite and solve (2) as the constrained nonlinear problem

$$\min_{\theta, \gamma} \gamma, \quad \text{s.t. } r_i(\theta) \leq \gamma, \quad (18)$$

where the inner max operator is removed. Note that for certain applications, additional constraints must be imposed on θ , e.g., chirality for triangulation (4). Given the solution θ^* and γ^* , the support set can be obtained as the data whose residual $r_i(\theta^*)$ equals γ^* . In our work, we use Matlab’s SQP algorithm in `fmincon` to solve (18).

We build an efficient support set update routine based on SQP. Let \mathcal{P} and \mathcal{Q} be two subsets of \mathcal{X} , where $\theta(\mathcal{P})$ is known and we wish to update it to obtain $\theta(\mathcal{Q})$. Also, let \mathcal{Q} be obtained from \mathcal{P} by removing from or adding to \mathcal{P} a small number of data; we thus expect $\theta(\mathcal{Q})$ to be close to $\theta(\mathcal{P})$. Our update procedure is to simply “hot start” SQP with $\theta(\mathcal{P})$ when solving (18) on \mathcal{Q} . Given the solution $\theta(\mathcal{Q})$, the support set of \mathcal{Q} can be “read off” from the residuals. In practice, this simple update scheme is very efficient.

Much greater efficiency can be obtained by specialising the update routine for certain types of residuals. For linear regression (3), we employ the well-known *vertex-to-vertex* descend algorithm (see [4, Chapter 2]) for Chebyshev regression (l_∞ minimisation), which is extremely fast.

5. Results

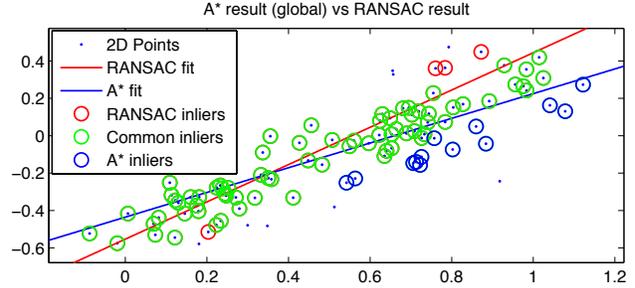
We performed a number of experiments on several common estimation problems. We focussed on comparing our algorithm (A*) against globally optimal methods for robust estimation. For the quantile regressor (7), we tested Matoušek’s method [14]. For the maximum consensus estimator (1), we tested BFS (Sec. 3), BnB with MaxFS formulation⁴ [21], and BnB with bilinear formulation (BILIN)⁵ [13]. RANSAC was also run as a baseline.

We divided our results into two categories: linear regression (3) and problems with pseudoconvex residuals. Note that MaxFS and BILIN only works for linearised residuals.

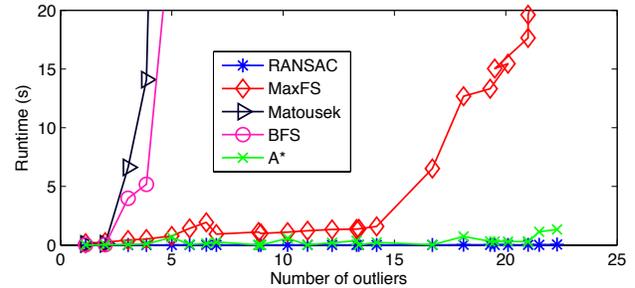
A* was implemented fully in Matlab: support set updating was conducted using `fmincon` for the pseudo-

⁴Using our own implementation. The original code was unavailable.

⁵Using the original author’s implementation.



(a) Typical data instance and lines fitted by RANSAC and A*. In this data instance, RANSAC found 76 inliers while A* found 87 inliers.



(b) Average runtime of different methods against number of outliers.

Figure 3. Line fitting results.

convex case, whilst the vertex-to-vertex algorithm was programmed by ourselves. These routines were also used for Matoušek and BFS. MaxFS requires solving Mixed Integer Linear Programming (MILP) problems, for which we used the state-of-the-art Gurobi solver. For the LP subroutines in BILIN, we made use of the highly efficient MOSEK solver.

For RANSAC, the standard stopping criterion [7] was used with confidence $\rho = 0.99$. However, if the criterion had been satisfied before 100 iterations, we forced RANSAC to continue until 100 iterations. On the other hand, if the number of unique p -subsets for the problem was less than 100, all the p -subsets were enumerated and tested.

All the experiments were run on a standard system with 2.70 GHz processor and 8 GB of RAM.

5.1. Linear regression

Line fitting We generated $N = 100$ points around a straight line under Gaussian noise with $\sigma_{in} = 0.1$. A number of the points were then corrupted by a larger Gaussian noise ($\sigma_{out} = 1$) to simulate outliers. The inlier threshold in (1) was chosen as $\epsilon = 0.3$, and k for (7) was set as the true number of outliers (known after solving (1) exactly).

Fig. 3(a) shows a typical data instance and result. Inliers found by RANSAC are circled in red, while those found by A* are circled in blue. Points circled in green are inliers found by both RANSAC and A*. As expected, RANSAC found a suboptimal solution with 76 inliers, while A* found the globally optimal solution with 87 inliers.

Methods	Linearised homography estimation						Linearised fundamental matrix estimation			
	Road Sign		Cambridge		Keble College		Valbonne Church		Merton College III	
	NSub	Runtime (s)	NSub	Runtime (s)	NSub	Runtime (s)	NSub	Runtime (s)	NSub	Runtime (s)
RANSAC	100	0.0903	100	0.1130	100	0.3689	100	0.0417	100	0.0448
MaxFS	67548	1.0433	65118	1.8661	1976336	> 3600	16267	9.2132	15515	5.7162
BILIN	2526	> 3600	1252	> 3600	8	> 3600	2220	> 3600	140	> 3600
Matoušek	220743	> 3600	189901	> 3600	179353	> 3600	238006	> 3600	237124	> 3600
BFS	166852	> 3600	143750	> 3600	137062	> 3600	71178	525.2548	194183	3234.6948
A*	911	1.2936	1473	2.2619	560	2.9121	1838	1.3922	589	0.6458
	$N = 36, p = 8, \epsilon = 0.1$ $ \mathcal{I}_R = 29.67, \mathcal{I}^* = 30$		$N = 55, p = 8, \epsilon = 0.1$ $ \mathcal{I}_R = 44.80, \mathcal{I}^* = 47$		$N = 299, p = 8, \epsilon = 0.1$ $ \mathcal{I}_R = 282.74, \mathcal{I}^* = 284$		$N = 58, p = 8, \epsilon = 0.1$ $ \mathcal{I}_R = 43.66, \mathcal{I}^* = 52$		$N = 187, p = 8, \epsilon = 0.1$ $ \mathcal{I}_R = 176.69, \mathcal{I}^* = 181$	

N: data size, p: number of parameters, ϵ : inlier threshold
 $|\mathcal{I}_R|$: average consensus size of RANSAC, $|\mathcal{I}^*|$: optimal consensus size, NSub: number of subproblems solved

Table 1. Results for linearised homography estimation and linearised fundamental matrix estimation.

We varied the number of points that were corrupted as outliers, and invoke all the compared methods. Each method was run 100 times, and the average runtime was recorded. Note that, apart from RANSAC, the methods deterministically gave the same regression result. Fig. 3(b) shows the average consumed time of the different methods against number of outliers (again, this number is known after (1) is solved). As predicted, Matoušek and BFS scaled extremely badly with the number of outliers, since progressively larger tree depth need to be explored. MaxFS also scaled badly with the number of outliers. We do not plot BILIN since it required more than 120 seconds even for 1 outlier. A* was only slower than RANSAC but vastly outperformed the other methods. The scaling property of A* was also much better. We observed that RANSAC produced suboptimal solutions in 97% of the runs.

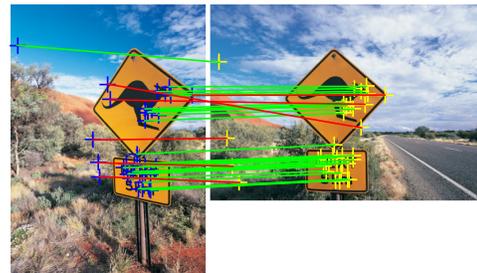
Linearised homography We tested on homography estimation using linear regression, where the dimensionality of θ is 8 (see [9, Chapter 4] on linearising the homography constraint). In Sec. 5.2, we will present results on homography estimation using geometric (pseudoconvex) residuals. Note that each keypoint match produces two residual functions; thus the number of data N and outliers k are doubled in the actual optimisation (see [9, Chapter 4] for details).

We tested on three image pairs, the first two pairs used extensively in previous works on robust geometric estimation: Road Sign (image index 1 and 2) [16], Keble College (image index 2 and 3)⁶, and Cambridge (downloaded from Flickr); see Fig. 4. SIFT keypoints were detected and matched using the VLFeat toolbox⁷, and Lowe’s second nearest neighbour test was invoked to prune matches. The keypoints were then normalised before generating the linearised homography constraints. The inlier threshold ϵ was chosen as $\epsilon = 0.1$, and k for quantile regression was set as the true outlier count. Table 1 presents the results of all methods (results of RANSAC are averages over 100 runs).

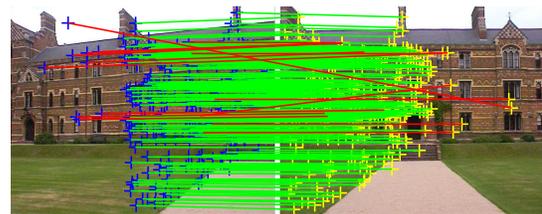
In addition to reporting actual runtime, we also show the

⁶<http://www.robots.ox.ac.uk/~vgg/data/>

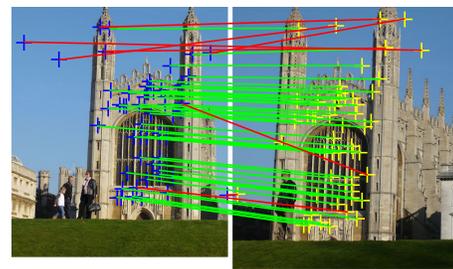
⁷<http://www.vlfeat.org>



(a) Road Sign (image index 1 and 2)



(b) Keble College (image index 2 and 3)



(c) Cambridge

Figure 4. Homography estimation result by A* on the three image pairs. Lines in green and red respectively indicate detected inlier and outlier SIFT keypoint matches.

number of subproblems (NSub) invoked/solved. This is defined as the following for the various methods:

- RANSAC: Number of trials before termination.
- MaxFS: Number of simplex iterations as reported in the Gurobi output.

Methods	Homography estimation (pseudoconvex residual)						Triangulation (pseudoconvex residual)			
	Road Sign		Cambridge		Keble College		Cathedral		House	
	NSub	Runtime (s)	NSub	Runtime (s)	NSub	Runtime (s)	NSub	Runtime (s)	NSub	Runtime (s)
RANSAC	100	0.0815	100	0.0824	100	0.1371	35	2.5845	56	4.1530
Matoušek	54921	1909.1582	34187	> 3600	12912	> 3600	280	12.8718	427	18.0139
BFS	8530	747.3163	37368	> 3600	14104	> 3600	119	5.0947	151	5.9715
A*	688	120.9280	1599	262.2883	56	19.7666	136	5.5920	281	8.7528
	$N = 36, p = 8, \epsilon = 0.1$		$N = 55, p = 8, \epsilon = 0.1$		$N = 299, p = 8, \epsilon = 0.1$		$N = 7, p = 3, \epsilon = 0.01$		$N = 8, p = 3, \epsilon = 0.01$	
	$ \mathcal{I}_R = 29.85, \mathcal{I}^* = 30$		$ \mathcal{I}_R = 44.70, \mathcal{I}^* = 47$		$ \mathcal{I}_R = 282.47, \mathcal{I}^* = 284$		$ \mathcal{I}_R = 1, \mathcal{I}^* = 2$		$ \mathcal{I}_R = 3, \mathcal{I}^* = 4$	

N: data size, p: number of parameters, ϵ : inlier threshold
 $|\mathcal{I}_R|$: average consensus size of RANSAC, $|\mathcal{I}^*|$: optimal consensus size, NSub: number of subproblems solved

Table 2. Results for homography estimation and triangulation with pseudoconvex residuals.

- BILIN: Number of LP instances solved as reported in the MOSEK output.
- Matoušek, BFS and A*: Number of support set update instances (including in heuristic computation for A*).

A runtime limit of 1 hour was imposed on all methods. For methods that did not terminate within the limit, the NSub values reported were as obtained at the 1 hour mark.

It is clear that A* was much faster than the other global optimisers. Although A* still took two orders of magnitude more time than RANSAC, the fact that A* terminated in seconds makes it a very practical global optimiser. BFS and Matoušek did not finish within 1 hour, thus pointing to the vast improvement in tree search efficiency given by our ideas in Sec. 3. MaxFS was efficient for Road Sign and Cambridge; however, in Keble College its run time increased drastically, owing to the much larger data size N . BILIN did not finish within 1 hour on all datasets; the method also suffers from scalability issues as N increases.

Linearised fundamental matrix We repeated the previous experiment for linearised fundamental matrix estimation, where the parameter of interest θ is 8 dimensional and inlier threshold $\epsilon = 0.1$ (see [9, Chapter 11] for the constraint linearisation step). To cogently test the optimisation performance of all methods, we did not enforce the rank-2 constraint on the resulting fundamental matrices. For this experiment, we used Valbonne Church (image index 3 and 7) and Merton College III (image index 2 and 3) images from Oxford VGG. Results are shown in Table 1. Similar performances were observed for all methods.

5.2. Pseudoconvex residuals

Homography estimation We used the same datasets and settings as the linearised homography estimation experiment. The inlier threshold ϵ was set to 0.1 pixels, but due to using the pseudoconvex residual function (5), the maximum consensus sets obtained were different from the linearised case. The results are shown in Table 2. It is apparent that BFS and Matoušek were faster here than in the linearised case, even though support set updating is slower for pseudoconvex residuals. The reason behind this is that there is

no doubling of the effective data size and outlier numbers due to linearising the homography constraint (observe that NSub values were much higher in Table 1). Although A* was slower in this experiment, it was still much faster than BFS and Matoušek, especially on the larger Cambridge and Keble College datasets. Note that A* managed to return the globally optimal estimate within tens of seconds.

Triangulation We used the Cathedral and House image sequences [16], which contain a large number of feature tracks. Tracks which are shorter than 7 frames were removed, thus 191 and 547 tracks remained for Cathedral and House respectively. For each feature track, the different methods were executed with an inlier threshold of 0.01 pixels. Table 2 summarises the results, which were averaged over the number of tracks in each dataset. Again, A* was much more efficient than Matoušek and BFS.

From a practical standpoint, owing to the small p and N for this problem (the longest track was less than $N = 20$ for both datasets), the globally optimal result can be more efficiently found by exhaustively testing all $\binom{N}{p+1}$ subsets of \mathcal{X} [15, 5]. Nonetheless, our conclusion that A* is more efficient than Matoušek and BFS still holds. Note also that although RANSAC enumerated all $\binom{N}{p}$ minimal subsets in these problems, it was unable to find the global optima.

6. Conclusions

We presented an efficient globally optimal algorithm for maximum consensus, based on the A* tree search algorithm. A heuristic function was proposed and proven to be admissible for use with A*. Experiments showed that the runtime of our method is several orders of magnitude smaller than previous exact methods. Our work identifies a promising direction to solve maximum consensus exactly. The good performance of A* relies on the tightness of the heuristic h as a lower bound of h^* . Our method thus has great potential to be speeded up further by constructing specialised heuristic functions for the intended applications.

Acknowledgements This work was supported by ARC grants DP130102524 and DE130101775.

References

- [1] http://en.wikipedia.org/wiki/LP-type_problem. 2
- [2] N. Amenta, M. Bern, and D. Eppstein. Optimal point placement for mesh smoothing. In *SODA*, 1997. 2
- [3] B. Chazelle and J. Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimensions. In *Symp. Discrete Algorithms*, 1993. 6
- [4] E. W. Cheney. *Introduction to approximation theory*. McGraw-Hill, 1966. 6
- [5] O. Enqvist, E. Ask, F. Kahl, and K. Åström. Robust fitting for multiple view geometry. In *ECCV*, 2012. 1, 8
- [6] D. Eppstein. Quasiconvex programming. *Combinatorial and Computational Geometry*, 25, 2005. 2, 3, 6
- [7] 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, 6
- [8] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. on Systems Science and Cybernetics*, 4(2):100–107, 1968. 1, 4
- [9] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2nd edition, 2004. 7, 8
- [10] F. Kahl. Multiple view geometry and the l_∞ -norm. In *ICCV*, 2005. 2, 6
- [11] Q. Ke and T. Kanade. Quasiconvex optimization for robust geometric reconstruction. In *ICCV*, 2005. 2, 6
- [12] H. Li. A practical algorithm for l_∞ triangulation with outliers. In *CVPR*, 2007. 2
- [13] H. Li. Consensus set maximization with guaranteed global optimality for robust geometry estimation. In *ICCV*, 2009. 1, 6
- [14] J. Matoušek. On geometric optimization with few violated constraints. *Discrete and computational geometry*, 14(1):365–384, 1995. 1, 2, 3, 6
- [15] C. Olsson, O. Enqvist, and F. Kahl. A polynomial-time bound for matching and registration with outliers. In *CVPR*, 2008. 1, 8
- [16] C. Olsson, A. Eriksson, and R. Hartley. Outlier removal using duality. In *CVPR*, 2010. 2, 5, 7, 8
- [17] C. Olsson, A. Eriksson, and F. Kahl. Efficient optimization for l_∞ -problems using pseudoconvexity. In *ICCV*, 2007. 2, 6
- [18] M. Sharir and E. Welzl. A combinatorial bound for linear programming and related problems. In *STACS*, 1992. 1, 2
- [19] K. Sim and R. Hartley. Removing outliers using the l_∞ norm. In *CVPR*, 2006. 2, 5
- [20] J. Yu, A. Eriksson, T.-J. Chin, and D. Suter. An adversarial optimization approach to efficient outlier removal. In *ICCV*, 2011. 2, 5
- [21] Y. Zheng, S. Sugimoto, and M. Okutomi. Deterministically maximizing feasible subsystems for robust model fitting with unit norm constraints. In *CVPR*, 2011. 1, 6