Robust Fitting in Computer Vision: Easy or Hard?

Tat-Jun Chin · Zhipeng Cai · Frank Neumann

Received: 29 January 2019 / Accepted: 31 July 2019 / Published online: 9 August 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

Robust model fitting plays a vital role in computer vision, and research into algorithms for robust fitting continues to be active. Arguably the most popular paradigm for robust fitting in computer vision is consensus maximisation, which strives to find the model parameters that maximise the number of inliers. Despite the significant developments in algorithms for consensus maximisation, there has been a lack of fundamental analysis of the problem in the computer vision literature. In particular, whether consensus maximisation is “tractable” remains a question that has not been rigorously dealt with, thus making it difficult to assess and compare the performance of proposed algorithms, relative to what is theoretically achievable. To shed light on these issues, we present several computational hardness results for consensus maximisation. Our results underline the fundamental intractability of the problem, and resolve several ambiguities existing in the literature.

Keywords

Robust fitting · Consensus maximisation · Inlier set maximisation · Computational hardness

1 Introduction

Robustly fitting a geometric model onto noisy and outlier-contaminated data is a necessary capability in computer vision (Meer 2004), due to the imperfectness of data acquisition systems and preprocessing algorithms (e.g., edge detection, keypoint detection and matching). Without robustness against outliers, the estimated model will be biased, leading to failure in the overall pipeline.

In computer vision, robust fitting is typically performed under the framework of inlier set maximisation, a.k.a. consensus maximisation (Fischler and Bolles 1981), where one seeks the model with the most number of inliers. For concreteness, say we wish to estimate the parameter vector $x \in \mathbb{R}^d$ that defines the linear relationship $a^T x = b$ from a set of outlier-contaminated measurements $D = \{ (a_i, b_i) \}_{i=1}^N$. The consensus maximisation formulation for this problem is as follows.

Problem 1 [MAXCON] Given input data $D = \{ (a_i, b_i) \}_{i=1}^N$, where $a_i \in \mathbb{R}^d$ and $b_i \in \mathbb{R}$, and an inlier threshold $\epsilon \in \mathbb{R}_+$, find the $x \in \mathbb{R}^d$ that maximises

$$
\Psi_\epsilon (x \mid D) = \sum_{i=1}^N \mathbb{I} (|a_i^T x - b_i| \leq \epsilon),
$$

where $\mathbb{I}$ returns 1 if its input predicate is true, and 0 otherwise.

The quantity $|a_i^T x - b_i|$ is the residual of the $i$-th measurement with respect to $x$, and the value given by $\Psi_\epsilon (x \mid D)$ is the consensus of $x$ with respect to $D$. Intuitively, the consensus of $x$ is the number of inliers of $x$. For the robust estimate to fit the inlier structure well, the inlier threshold $\epsilon$ must be set to an appropriate value; the large number of applications that employ the consensus maximisation framework indicate that this is usually not an obstacle.

Developing algorithms for robust fitting, specifically for consensus maximisation, is an active research area in computer vision. Currently, the most popular algorithms belong to the class of randomised sampling techniques, i.e., RANSAC (Fischler and Bolles 1981) and its variants (Choi et al. 2009; Raguram et al. 2013). Unfortunately, such techniques do not provide certainty of finding satisfactory solutions, let alone optimal ones (Tran et al. 2014).

Increasingly, attention is given to constructing globally optimal algorithms for robust fitting, e.g., Li (2009), Zheng et al. (2011), Enqvist et al. (2012), Bazin et al. (2013), Yang et al. (2014), Parra Bustos et al. (2014), Enqvist et al. (2015), Chin et al. (2015) and Campbell et al. (2017). Such algo-

Translated by Yair Weiss.
rithms are able to deterministically calculate the best possible solution, i.e., the model with the highest achievable consensus. This mathematical guarantee is regarded as desirable, especially in comparison to the “rough” solutions provided by random sampling heuristics.

Recent progress in globally optimal algorithms for consensus maximisation seems to suggest that global solutions can be obtained efficiently or tractably (Li 2009; Zheng et al. 2011; Enqvist et al. 2012; Bazin et al. 2013; Yang et al. 2014; Parra Bustos et al. 2014; Enqvist et al. 2015; Chin et al. 2015; Campbell et al. 2017). Moreover, decent empirical performances have been reported. This raises hopes that good alternatives to the random sampling methods are now available. However, to what extent is the problem solved? Can we expect the global algorithms to perform well in general? Are there fundamental obstacles toward efficient robust fitting algorithms? What do we even mean by “efficient”?

1.1 Our Contributions and Their Implications

Our contributions are theoretical. We resolve the above ambiguities in the literature, by proving the following computational hardness results. The implications of each result are also listed below.

MAXCON is NP-hard (Sect. 2).

\[ \implies \text{There are no algorithms that can solve MAXCON in time polynomial to the input size, which is proportional to } N \text{ and } d. \]

MAXCON is W[1]-hard in the dimension \( d \) (Sect. 3.2).

\[ \implies \text{There are no algorithms that can solve MAXCON in time } f(d)\text{poly}(N), \text{ where } f(d) \text{ is an arbitrary function of } d, \text{ and poly}(N) \text{ is a polynomial of } N. \]

MAXCON is APX-hard (Sect. 4).

\[ \implies \text{There are no polynomial time algorithms that can approximate MAXCON up to } (1 - \delta)\psi^* \text{ for any known factor } \delta, \text{ where } \psi^* \text{ is the maximum consensus.}^1 \]

As usual, the implications of the hardness results are subject to the standard complexity assumptions \( P \neq NP \) (Garey 1990) and \( \text{FPT} \neq \text{W}[1] \) (Downey and Fellows 1999).

Our analysis indicates the “extreme” difficulty of consensus maximisation. MAXCON is not only intractable (by standard notions of intractability Garey 1990; Downey and Fellows 1999), the W[1]-hardness result also suggests that any global algorithm will scale exponentially in a function of \( d \), i.e., \( N^{f(d)} \). In fact, if a conjecture of Erickson et al. (2006) holds, MAXCON cannot be solved faster than \( N^d \).

Thus, the decent performances in Li (2009), Zheng et al. (2011), Enqvist et al. (2012), Bazin et al. (2013), Yang et al. (2014), Parra Bustos et al. (2014), Enqvist et al. (2015), Chin et al. (2015) and Campbell et al. (2017) are unlikely to extend to the general cases in practical settings, where \( N \geq 1000 \) and \( d \geq 6 \) are common. More pessimistically, APX-hardness shows that MAXCON is impossible to approximate, in that there are no polynomial time approximation schemes (PTAS) (Vazirani 2001) for MAXCON. 2

A slightly positive result is as follows.

MAXCON is FPT (fixed parameter tractable) in the number of outliers \( o \) and dimension \( d \) (Sect. 3.3).

This is achieved by applying a special case of the algorithm of Chin et al. (2015) on MAXCON to yield a runtime of \( O((o + 1)d^{d+1}\text{poly}(N, d)) \). However, for most computer vision problems, the values of \( o \) and \( d \) are moderate to large (e.g., \( o \) in the range of hundreds, \( d \geq 5 \)), hence, in practice, the FPT algorithm is fast usually only for instances where \( o \) is small (e.g., \( o \leq 10 \)).

While this paper can also find relevance with a theoretical computer science audience, our work is important to the computer vision community since it helps to clarify the ambiguities on the efficiency and solvability of consensus maximisation (see also Sect. 1.2). Second, our analysis shows how the computational effort scales with the different input size parameters, thus suggesting more cogent ways for algorithm designers in computer vision to test and compare algorithms. Third, since developing algorithms for consensus maximisation is an active topic in computer vision, it is important for researchers to be aware of the fundamental limitations of solving the problem. Our theoretical findings also encourage researchers to consider alternative paradigms for robust fitting, e.g., deterministically convergent heuristic algorithms (Le et al. 2017; Purkait et al. 2017; Cai et al. 2018) or preprocessing techniques (Svärm et al. 2014; Parra Bustos and Chin 2015; Chin et al. 2016).

While our results are based specifically on MAXCON, which is concerned with fitting linear models, in practice, computer vision applications require the fitting of non-linear geometric models (e.g., fundamental matrix, planar perspective transforms, rotation matrices). However, while a case-by-case treatment is ideal, it is unlikely that non-linear consensus maximisation will be easier than linear consensus maximisation (Johnson et al. 1978; Ben-David et al. 2002; Aronov and Har-Peled 2008).

Note also that our purpose here is not to promote consensus maximisation as the “best” robust criterion. However, as a robust formulation that is “native” to computer vision, consensus maximisation enjoys prevalent use in the community.

---

1 It may be the case that MAXCON is in class APX, i.e., that it could be approximated in polynomial time to some factor. However, we are not aware of any such algorithms.

2 Since RANSAC does not provide any approximation guarantees, it is not an “approximation scheme” by standard definition (Vazirani 2001).
It is thus vital to know what is and isn’t possible according to current algorithmic thinking. Second, it is unlikely that other robust criteria are easier to solve (Bernholt 2005). Although some that use differentiable robust loss functions (e.g., M-estimators) can be solved up to local optimality, it is unknown how far the local optima deviate from the global solution.

1.2 Previous Complexity Analyses of Robust Fitting

In the broader literature, complexity results have been obtained for a number of robust criteria (Bernholt 2005; Erickson et al. 2006), such as Least Median Squares (LMS), Least Quantile of Squares (LQS), and Least Trimmed Squares (LTS). However, the previous works have not studied consensus maximisation, which, as alluded to in Sect. 1.1, is of significant importance to computer vision. Moreover, the analyses in Bernholt (2005) and Erickson et al. (2006) have focussed mainly on NP-hardness, while our work here establishes a broader set of intractability results (parametrised intractability, inapproximability).

A closely related combinatorial problem is maximum feasible subsystem (MaxFS), which aims to find the largest feasible subset of a set of infeasible linear constraints. A number of complexity results have been developed by Amaldi and Kann (1995) for MaxFS. Unfortunately, we were not able to transfer the results in Amaldi and Kann (1995) to MAXCON, hence, necessitating the present work.

In computer vision, a significant step towards complexity analysis of robust fitting (including consensus maximisation) was the work by Enqvist et al. (2012, 2015). Specifically, an $O(N^{d+1})$ algorithm was presented, which was regarded as tractable since $d$ is restricted to a few small values in the applications considered. Strictly speaking, however, Enqvist et al. (2012, 2015) have only established that robust fitting is in class XP (slice-wise polynomial) when parametrised by the dimension $d$, and this does not (yet) established tractability by standard definition (Downey and Fellows 1999). Unfortunately, our W[1]-hardness results in Sect. 3.2 rules out tractability when parametrised by $d$ alone.

1.3 Differences to the Conference Version

This paper is an extension of the conference version (Chin et al. 2018). The main differences to the conference version are:

- A correction is made to Algorithm 1 in Chin et al. (2018) to ensure consistency with the FPT result. Briefly, the previous Algorithm 1 conducts a depth-first tree search, whereas FPT requires breadth-first tree search.
- The runtime complexity of Algorithm 1 is corrected to $O((d + 1)^d \cdot \text{poly}(N, d))$. In Chin et al. (2018), it was shown as $O(d^d \cdot \text{poly}(N, d))$. Note that this modification does not change the FPT outcome.
- A faster FPT algorithm (Algorithm 2) with $O((d + 1)^{d+1} \cdot \text{poly}(N, d))$ runtime is developed using the repeated basis checking technique of Chin et al. (2015).
- The concept of kernelisation (Downey and Fellows 1999) is explored for MAXCON (Sect. 3.5).
- Empirical validation of the FPT runtime bound is now provided. The performance of the FPT algorithm on real data is also investigated (Sect. 3.6).

The rest of the paper is devoted to developing the above theoretical and empirical results.

2 NP-Hrdness

The decision version of MAXCON is as follows.

**Problem 2 [MAXCON-D]** Given data $D = \{(a_i, b_i)\}^N_{i=1}$, an inlier threshold $\epsilon \in \mathbb{R}^+$ and a number $\psi \in \mathbb{N}^+$, does there exist $x \in \mathbb{R}^d$ such that $\Psi_e(x | D) \geq \psi$?

Another well-known robust fitting paradigm is least median squares (LMS), where we seek the vector $x$ that minimises the median of the residuals

$$\min_{x \in \mathbb{R}^d} \text{med} \left| a_1^T x - b_1, \ldots, a_N^T x - b_N \right|.$$ (2)

LMS can be generalised by minimising the $k$-th largest residual instead

$$\min_{x \in \mathbb{R}^d} \text{kos} \left( a_1^T x - b_1, \ldots, a_N^T x - b_N \right).$$ (3)

where function kos returns its $k$-th largest input value.

Geometrically, LMS seeks the slab of the smallest width that contains half of the data points $D$ in $\mathbb{R}^{d+1}$. A slab in $\mathbb{R}^{d+1}$ is defined by a normal vector $x$ and width $w$ as

$$h_w(x) = \left\{ (a, b) \in \mathbb{R}^{d+1} : |a^T x - b| \leq \frac{1}{2} w \right\}.$$ (4)
Problem (3) thus seeks the thinnest slab that contains $k$ of the points. The decision version of (3) is as follows.

Problem 3 [k-SLAB] Given data $D = \{(a_i, b_i)\}_{i=1}^N$, an integer $k$ where $1 \leq k \leq N$, and a number $w' \in \mathbb{R}_+$, does there exist $x \in \mathbb{R}^d$ such that $k$ of the members of $D$ are contained in a slab $h_w(x)$ of width at most $w'$?

k-SLAB has been proven to be NP-complete in Erickson et al. (2006).

Theorem 1 MAXCON-D is NP-complete.

Proof Let $D$, $k$ and $w'$ define an instance of k-SLAB. This can be reduced to an instance of MAXCON-D by simply reusing the same $D$, and setting $\epsilon = \frac{1}{2}w'$ and $\psi = k$. If the answer to k-SLAB is positive, then there is an $x$ such that $k$ points from $D$ lie within vertical distance of $\frac{1}{2}w'$ from the hyperplane defined by $x$, hence $\Psi_\epsilon(x \mid D)$ must be at least $\psi$ and the answer to MAXCON-D is also positive. Conversely, if the answer to MAXCON-D is positive, then there is an $x$ such that $\psi$ points have vertical distance of less than $\epsilon$ to $x$, hence a slab that is centred at $x$ of width at most $w'$ can enclose $k$ of the points, and the answer to k-SLAB is also positive. \hfill \Box

The NP-completeness of MAXCON-D implies the NP-hardness of the optimisation version MAXCON. See Sect. 1.1 for the implications of NP-hardness.

3 Parametrised Complexity

Parametrised complexity is a branch of algorithmics that investigates the inherent difficulty of problems with respect to structural parameters in the input Downey and Fellows (1999). In this section, we report several parametrised complexity results of MAXCON.

First, the consensus set $C_\epsilon(x \mid D)$ of $x$ is defined as

$$C_\epsilon(x \mid D) := \{i \in \{1, \ldots, N\} \mid |a_i^T x - b_i| \leq \epsilon\}.$$ (5)

An equivalent definition of consensus (1) is thus

$$\Psi_\epsilon(x \mid D) = |C_\epsilon(x \mid D)|.$$ (6)

Henceforth, we do not distinguish between the integer subset $C \subseteq \{1, \ldots, N\}$ that indexes a subset of $D$, and the actual data that are indexed by $C$.

3.1 XP in the Dimension

The following is the Chebyshev approximation problem (Cheney 1966, Chapter 2) defined on the input data indexed by $C$:

$$\min_{x \in \mathbb{R}^d} \max_{i \in C} |a_i^T x - b_i|$$ (7)

Problem (7) has the linear programming (LP) formulation

$$\min_{x \in \mathbb{R}^d, \gamma \in \mathbb{R}} \gamma$$ (LP[C])

s.t. $|a_i^T x - b_i| \leq \gamma$, $i \in C$.

which can be solved in polynomial time. Chebyshev approximation also has the following property.

Lemma 1 There is a subset $B$ of $C$, where $|B| \leq d + 1$, such that

$$\min_{x \in \mathbb{R}^d} \max_{i \in \hat{B}} |a_i^T x - b_i| = \min_{x \in \mathbb{R}^d} \max_{i \in C} |a_i^T x - b_i|$$ (8)

Proof See Cheney (1966, Section 2.3). \hfill \Box

We call $B$ a basis of $C$. Mathematically, $B$ is the set of active constraints to LP[C], hence bases can be computed easily. In fact, LP[B] and LP[C] have the same minimisers. Further, for any subset $B$ of size $d + 1$, a method by de la Vallée-Poussin can solve LP[B] analytically in time polynomial to $d$; see Cheney (1966, Chapter 2) for details.

Let $x$ be an arbitrary candidate solution to MAXCON, and $(\hat{x}, \hat{\gamma})$ be the minimisers to LP[C]. We call $(\hat{x}, \hat{\gamma})$ the Chebyshev approximation problem on the consensus set of $x$. The following property can be established.

Lemma 2 $\Psi_\epsilon(\hat{x} \mid D) \geq \Psi_\epsilon(x \mid D)$.

Proof By construction, $\hat{\gamma} \leq \epsilon$. Hence, if $(a_i, b_i)$ is an inlier to $x$, i.e., $|a_i^T x - b_i| \leq \epsilon$, then $|a_i^T \hat{x} - b_i| \leq \hat{\gamma} \leq \epsilon$, i.e., $(a_i, b_i)$ is also an inlier to $\hat{x}$. Thus, the consensus of $\hat{x}$ is no smaller than the consensus of $x$. \hfill \Box

Lemmas 1 and 2 suggest a rudimentary algorithm for consensus maximisation that attempts to find the basis of the maximum consensus set, as encapsulated in the proof of the following theorem.

Theorem 2 MAXCON is XP (slice-wise polynomial) in the dimension $d$.

Proof Let $x^*$ be a witness to an instance of MAXCON-D with positive answer, i.e., $\Psi_\epsilon(x^* \mid D) \geq \psi$. Let $(\hat{x}^*, \hat{\gamma}^*)$ be the minimisers to LP[C]. By Lemma 2, $\hat{x}^*$ is also a positive witness to the instance. By Lemma 1, $\hat{x}^*$ can be found by enumerating all $(d + 1)$-subsets of $D$, and solving Chebyshev approximation (7) on each $(d + 1)$-subset. There are a total of $\binom{N}{d+1}$ subsets to check; including the time to evaluate $\Psi_\epsilon(x \mid D)$ for each candidate, the runtime of this simple algorithm is $O(N^{d+2}\text{poly}(d))$, which is polynomial in $N$ for a fixed $d$. \hfill \Box
Theorem 2 shows that for a fixed dimension $d$, MAXCON can be solved in time polynomial in the number of measurements $N$ [this is consistent with the results in Enqvist et al. (2012, 2015)]. However, this does not imply that MAXCON is tractable (following the standard meaning of tractability in complexity theory Garey 1990; Downey and Fellows 1999). Moreover, in practical applications, $d$ could be large (e.g., $d \geq 5$), thus the rudimentary algorithm above will not be efficient for large $N$.

### 3.2 W[1]-Hard in the Dimension

Can we remove $d$ from the exponent of the runtime of a globally optimal algorithm? By establishing W[1]-hardness in the dimension, this section shows that it is not possible. Our proofs are inspired by, but extends quite significantly from, that of Giannopoulos et al. (2009, Section 5). First, the source problem is as follows.

**Problem 4** [k-CLIQUE] Given undirected graph $G = (V, E)$ with vertex set $V$ and edge set $E$ and a parameter $k \in \mathbb{N}_+$, does there exist a clique in $G$ with $k$ vertices?

k-CLIQUE is W[1]-hard w.r.t. parameter $k$. Here, we demonstrate an FPT reduction from k-CLIQUE to MAXCON-D with fixed dimension $d$.

#### 3.2.1 Generating the Input Data

Given input graph $G = (V, E)$, where $V = \{1, \ldots, M\}$, and size $k$, we construct a $(k + 1)$-dimensional point set $D_G = \{(a_v, b_1)\}^N_i=1 = D_V \cup D_E$ as follows:

- The set $D_V$ is defined as
  $\mathcal{D}_V = \{(a^v_v, b^v_v) \in \mathbb{R}^k : (a^v_v, b^v_v) \in \mathcal{D}_V \}$

  where
  
  $a^v_v = [0, \ldots, 0, 1, 0, \ldots, 0]^T$  

  is a $k$-dimensional vector of 0’s except at the $\alpha$-th element where the value is 1, and
  
  $b^v_v = v$.  

- The set $D_E$ is defined as
  $\mathcal{D}_E = \{(a^u_v, b^u_v) \mid u, v = 1, \ldots, M, \langle u, v \rangle \in E, \langle v, u \rangle \in E, \alpha, \beta = 1, \ldots, k, \alpha < \beta \}$

  where
  
  $a^u_v = [0, \ldots, 0, 1, 0, \ldots, 0, M, 0, \ldots, 0]^T$  

  is a $k$-dimensional vector of 0’s, except at the $\alpha$-th element where the value is 1 and the $\beta$-th element where the value is $M$, and
  
  $b^u_v = u + M v$.  

The size $N$ of $D_G$ is thus $|D_V| + |D_E| = kM + 2|E|\binom{k}{2}$.

#### 3.2.2 Setting the Inlier Threshold

Under our reduction, $x \in \mathbb{R}^d$ is responsible for “selecting” a subset of the vertices $V$ and edges $E$ of $G$. First, we say that $x$ selects vertex $v$ if a point $(a^v_v, b^v_v) \in \mathcal{D}_V$, for some $\alpha$, is an inlier to $x$, i.e., if

$$|(a^v_v)^T x - b^v_v| \leq \epsilon \equiv x_\alpha \in \left[\epsilon - v, \epsilon + v\right],$$

where $x_\alpha$ is the $\alpha$-th element of $x$. The key question is how to set the value of the inlier threshold $\epsilon$, such that $x$ selects no more than $k$ vertices, or equivalently, such that $\Psi_\epsilon(x \mid D_V) \leq k$ for all $x$.

**Lemma 3** If $\epsilon < \frac{1}{2}$, then $\Psi_\epsilon(x \mid D_V) \leq k$, with equality achieved if and only if $x$ selects $k$ vertices of $G$.

**Proof** For any $u$ and $v$, the ranges $[u - \epsilon, u + \epsilon]$ and $[v - \epsilon, v + \epsilon]$ cannot overlap if $\epsilon < \frac{1}{2}$. Hence, $x_\alpha$ lies in at most one of the ranges, i.e., each element of $x$ selects at most one of the vertices; see Fig. 1. This implies that $\Psi_\epsilon(x \mid D_V) \leq k$. □

Second, a point $(a^u_v, b^u_v) \in \mathcal{D}_E$ is an inlier to $x$ if

$$|(a^u_v)^T x - b^u_v| \leq \epsilon \equiv |(x_\alpha - u) + M(x_\beta - v)| \leq \epsilon.$$  

As suggested by (16), the pairs of elements of $x$ are responsible for selecting the edges of $G$. To prevent each element pair $x_\alpha, x_\beta$ from selecting more than one edge, or equivalently, to maintain $\Psi_\epsilon(x \mid D_E) \leq \binom{k}{2}$, the setting of $\epsilon$ is crucial.
Lemma 4 If $\epsilon < \frac{1}{2}$, then $\Psi_\epsilon(x | D_G) \leq \binom{k}{2}$, with equality achieved if and only if $x$ selects $\binom{k}{2}$ edges of $G$.

Proof For each $\alpha, \beta$ pair, the constraint (16) is equivalent to the two linear inequalities

\begin{align*}
    x_\alpha + Mx_\beta - u - Mv &\leq \epsilon, \\
    x_\alpha + Mx_\beta - u - Mv &\geq -\epsilon,
\end{align*}

which specify two opposing half-planes (i.e., a slab) in the space $(x_\alpha, x_\beta)$. Note that the slopes of the half-plane boundaries do not depend on $u$ and $v$. For any two unique pairs $(u_1, v_1)$ and $(u_2, v_2)$, we have the four linear inequalities

\begin{align*}
    x_\alpha + Mx_\beta - u_1 - Mv_1 &\leq \epsilon, \\
    x_\alpha + Mx_\beta - u_1 - Mv_1 &\geq -\epsilon, \\
    x_\alpha + Mx_\beta - u_2 - Mv_2 &\leq \epsilon, \\
    x_\alpha + Mx_\beta - u_2 - Mv_2 &\geq -\epsilon.
\end{align*}

The system (18) can be simplified to

\begin{align*}
    \frac{1}{2} [u_2 - u_1 + M(v_2 - v_1)] &\leq \epsilon, \\
    \frac{1}{2} [u_1 - u_2 + M(v_1 - v_2)] &\leq \epsilon.
\end{align*}

(19)

Setting $\epsilon < \frac{1}{2}$ ensures that the two inequalities (19) cannot be consistent for all unique pairs $(u_1, v_1)$ and $(u_2, v_2)$. Geometrically, with $\epsilon < \frac{1}{2}$, the two slabs defined by (17) for different $(u_1, v_1)$ and $(u_2, v_2)$ pairs do not intersect; see Fig. 2 for an illustration.

Hence, if $\epsilon < \frac{1}{2}$, each element pair $x_\alpha, x_\beta$ of $x$ can select at most one of the edges. Cumulatively, $x$ can select at most $\binom{k}{2}$ edges, thus $\Psi_\epsilon(x | D_E) \leq \binom{k}{2}$. $\square$

Up to this stage, we have shown that if $\epsilon < \frac{1}{2}$, then $\Psi_\epsilon(x | D_G) \leq k + \binom{k}{2}$, with equality achievable if there is a clique of size $k$ in $G$. To establish the FPT reduction, we need to establish the reverse direction, i.e., if $\Psi_\epsilon(x | D_G) = k + \binom{k}{2}$, then there is a $k$-clique in $G$. The following lemma shows that this can be assured by setting $\epsilon < \frac{1}{M+2}$.

Lemma 5 If $\epsilon < \frac{1}{M+2}$, then $\Psi_\epsilon(x | D_G) \leq k + \binom{k}{2}$, with equality achievable if and only if there is a clique of size $k$ in $G$.

Proof The ‘only if’ direction has already been proven. To prove the ‘if’ direction, we show that if $\epsilon < \frac{1}{M+2}$ and $\Psi_\epsilon(x | D_G) = k + \binom{k}{2}$, the subgraph $S(x) = \{ [x_1], \ldots, [x_k] \}$ is a $k$-clique, where each $[x_i]$, represents a vertex index in $G$. Since $\epsilon < \frac{1}{M+2}$, $[x_\alpha] = u$ if and only if $(a_\alpha, b_\alpha)$ is an inlier. Therefore, $S(x)$ consists of all vertices selected by $x$. From Lemmas 3 and 4, when $\Psi_\epsilon(x | D_G) = k + \binom{k}{2}$, $x$ is consistent with $k$ points in $D_V$ and $\binom{k}{2}$ points in $D_E$. The inliers in $D_V$ specify the $k$ vertices in $S(x)$. The ‘if’ direction is true if all selected $\binom{k}{2}$ edges are only edges in $S(x)$, i.e., for each inlier point $(a_\alpha, b_\alpha) \in D_E$, $(a_\beta, b_\beta)$ and $(a_\beta, b_\beta)$ are also inliers w.r.t. $x$. The prove is done by contradiction:

If $\epsilon < \frac{1}{M+2}$, given an inlier $(a_\alpha, v)$, from (16) we have:

\begin{align*}
    |x_\alpha - u| + M(x_\beta - v) \\
    = |([x_\alpha] - u) + M([x_\beta] - v)| + ([x_\alpha] - [x_\alpha]) + M(x_\beta - [x_\beta]) | \\
    < \frac{1}{M+2}.
\end{align*}

(20)

Assume at least one of $(a_\alpha, b_\alpha)$ and $(a_\beta, b_\beta)$ is not an inlier, from (15) and $\epsilon < \frac{1}{M+2}$, we have $|x_\alpha| \neq u$ or $|x_\beta| \neq v$, which means that at least one of $([x_\alpha] - u)$ and $([x_\beta] - v)$ is not zero. Since all elements of $x$ satisfy (15), both $([x_\alpha] - u)$ and $([x_\beta] - v)$ are integers between $\{- (M - 1), (M - 1)\}$. 

Fig. 1 The blue dots indicate the integer values in the dimensions $x_\alpha$ and $x_\beta$. If $\epsilon < \frac{1}{2}$, then the ranges defined by (15) for all $v = 1, \ldots, M$ do not overlap. Hence, $x_\alpha$ can select at most one vertex of the graph (Color figure online).

Fig. 2 The blue dots indicate the integer values in the dimensions $x_\alpha$ and $x_\beta$. If $\epsilon < \frac{1}{2}$, then any two slabs defined by (17) for different $(u_1, v_1)$ and $(u_2, v_2)$ pairs do not intersect. The figure shows two slabs corresponding to $u_1 = 1, v_1 = 5, u_2 = 2, v_2 = 5$ (Color figure online).
3.2.3 Completing the Reduction

If only one of \((|x_a| - u)\) and \((|x_b| - v)\) is not zero, then\n
\[ \|(x_a) - u\| + M(|x_b| - v) \geq |1 + M \cdot 0| = 1. \]

If both are not zero, then\n
\[ \|(x_a) - u\| + M(|x_b| - v) \geq |(M - 1) + M \cdot 1| = 1 \]

Therefore, we have\n
\[ |(x_a) - u| + M(|x_b| - v)| \geq 1. \quad (21) \]

Also due to (15), we have\n
\[ |(x_a) - u| + M(|x_b| - v)| \leq (M + 1) \cdot \epsilon = \frac{M + 1}{M + 2} \cdot \epsilon. \quad (22) \]

Combining (21) and (22), we have\n
\[ \frac{|(x_a) - u| + M(|x_b| - v)|}{(M + 1) \cdot \epsilon} \geq \frac{1}{M + 2}. \quad (23) \]

which contradicts (20). It is obvious that \(S(x)\) can be computed within linear time. Hence, the “if” direction is true when \(\epsilon < \frac{1}{M + 2}\). \(\square\)

To illustrate Lemma 5, Fig. 3 depicts the value of \(\psi(x \mid D_G)\) in the subspace \((x_a, x_b)\) for \(\epsilon < \frac{1}{M + 2}\). Observe that \(\psi(x \mid D_G)\) attains the highest value of 3 in this subspace if and only if \(x_a\) and \(x_b\) select a pair of vertices that are connected by an edge in \(G\).

3.2.3 Completing the Reduction

We have demonstrated a reduction from k-CLIQUE to MAXCON-D, where the main work is to generate data \(D_G\) which has number of measurements \(N = k|V| + 2|E|\) that is linear in \(|G|\) and polynomial in \(k\), and dimension \(d = k\).

In other words, the reduction is FPT in \(k\). Setting \(\epsilon < \frac{1}{M + 2}\) and \(\psi = k + (\frac{3}{4})\) completes the reduction.

**Theorem 3** MAXCON is \(W[1]\)-hard w.r.t. the dimension \(d\).

**Proof** Since k-CLIQUE is \(W[1]\)-hard w.r.t. \(k\), by the above FPT reduction, MAXCON is \(W[1]\)-hard w.r.t. \(d\). \(\square\)

The implications of Theorem 3 have been discussed in Sect. 1.1.

3.3 FPT in the Number of Outliers and Dimension

Let \(f(C)\) and \(\hat{x}_C\) respectively indicate the minimised objective value and minimiser of LP[C]. Consider two subsets \(P\) and \(Q\) of \(D\), where \(P \subseteq Q\). The statement

\[ f(P) \leq f(Q) \]

follows from the fact that LP[P] contains only a subset of the constraints of LP[Q]; we call this property monotonicity.

Let \(x^*\) be a global solution of an instance of MAXCON, and let \(\mathcal{I}^* := \mathcal{C}(x^* \mid D) \subset D\) be the maximum consensus set. Let \(C\) index a subset of \(D\), and let \(B\) be the basis of \(C\). If \(f(C) > \epsilon\), then by Lemma 1

\[ f(D) \geq f(C) = f(B) > \epsilon. \]

The monotonicity property affords us further insight.

**Lemma 6** At least one point in \(B\) do not exist in \(\mathcal{I}^*\).

**Proof** By monotonicity,

\[ \epsilon < f(B) \leq f(\mathcal{I}^* \cup B). \]

Hence, \(\mathcal{I}^* \cup B\) cannot be equal to \(\mathcal{I}^*\), for if they were equal, then \(f(\mathcal{I}^* \cup B) = f(\mathcal{I}^*) \leq \epsilon\) which violates (26). \(\square\)

The above observations suggest an algorithm for MAXCON that iteratively removes basis points to find a consensus set, as summarised in Algorithm 1. This algorithm is a special case of the technique of Chini et al. (2015). Note that in the worst case, Algorithm 1 finds a solution with consensus \(d\) (i.e., the minimal case to fit \(x\)), if there are no solutions with higher consensus to be found.

**Theorem 4** MAXCON is FPT in the number of outliers and dimension.

**Proof** Algorithm 1 conducts a breadth-first tree search to find a sequence of \(basis\) points to remove from \(D\) to yield a consensus set. By Lemma 6, the longest sequence of basis points that needs to be removed is \(o = N - |\mathcal{I}^*|\), which is also the maximum tree depth searched by the algorithm (each descend
of the tree removes one point). The number of nodes visited is of order $(d + 1)^o$, since for non-degenerate problem instances, the branching factor of the tree is $|B|$, and by Lemma 1, $|B| \leq d + 1$.

At each node, LP[C] is solved, with the largest of these LPs having $d + 1$ variables and $N$ constraints. Algorithm 1 thus runs in $O((d + 1)^o\text{poly}(N, d))$ time, which is exponential only in the number of outliers $o$ and dimension $d$. \hfill \Box

### 3.4 A Faster FPT Algorithm

In Algorithm 1, repeated bases might be traversed, which may cause redundant branches being generated during the tree search. While this does not affect the FPT outcome of Algorithm 1, it does suggest a faster FPT algorithm can be constructed of the redundant paths can be avoided.

Algorithm 2 describes the redundancy avoidance variant of Algorithm 1. Following Chin et al. (2015), the idea is to use a hash table $H$ to store all the previously visited bases for efficient repetition detection. Given a basis $B$, we branch (i.e., generate child bases thereof) only when it is not in $H$.

Theorem 5 MAXCON is solvable by Algorithm 2 in $O((o + 1)^{(d+1)}\text{poly}(N, d))$ time.

**Proof** According to (Matoušek 1995, Theorem 2.3 (i)), the number of unique bases on and before level $o$ is $O((o + 1)^{(d+1)})$. And since branching is not allowed for repeated bases, the parent basis of any repeated basis must be unique. Therefore, the number of bases traversed by Algorithm 2 is $O((o + 1)^{(d+1)}(d + 1))$, which is the number of unique basis $O((o + 1)^{(d+1)})$ times the size $d + 1$ of a branch. And since the runtime spent on each basis is $\text{poly}(N, d)$, the runtime of Algorithm 2 is $O((o + 1)^{(d+1)}(d + 1)\cdot \text{poly}(N, d)) \leq O((o + 1)^{(d+1)}\text{poly}(N, d))$. \hfill \Box

Note that in the context of parametrised complexity, $O((o + 1)^{(d+1)}) \neq O((o + 1)^d)$ or $O(o(d+1))$ since both $o$ and $d$ are parameters. Also, in computer vision applications, $o$ can be much larger than $d$. Therefore, $O((o + 1)^{(d+1)})$ is generally much smaller than $O((d + 1)^o)$. The empirical validity and tightness of these two bounds will be shown later in Sect. 3.6.1.

### 3.4.1 Further Speedups with Heuristics

In Chin et al. (2015), an admissible heuristic is used to guide the tree search. This effectively changes the breadth first regime in Algorithm 2 to a*-tree search (Hart et al. 1968). While in practice, this significantly speeds up convergence, the worst case time complexity of the technique is not improved. Thus, for brevity we will not describe the a*-tree search algorithm here, and instead refer the interested reader to Chin et al. (2015). In the experiments in Sect. 3.6.1, however, we will use the a* version for practical reasons.

### 3.5 Relation Between FPT and Kernelisation

An important indication of a problem being FPT is the existence of the kernel, which is defined as follows for MAXCON following (Cygan et al. 2015, Definition 2.1).

**Definition 1** A kernelisation algorithm for MAXCON is a polynomial time algorithm whereby given an instance of MAXCON with size $|\mathcal{I}|$, returns an equivalent instance—called a kernel - whose size $|\mathcal{I}'| \leq g(o, d)$ for some computable function $g : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$.

Intuitively, a kernel is a polynomial time algorithm that finds the “core” structure of a given problem instance whose size is bounded by a function that depends only on the parameters of the problem. The existence of a kernel indicates that when $o$ and $d$ are small (such that $g(o, d) < |\mathcal{I}|$), we might still be able to solve MAXCON efficiently. The following
Lemma 7 \textit{MAXCON admits a kernel with }$g(o, d) = \mathcal{O}((o + 1)^{(d+1)})$.\]

\textbf{Proof} According to the proof of Theorem 5, MAXCON can be solved in $g(o, d) \cdot \text{poly}(N, d)$ time by Algorithm 2. Assume the input size is $|\mathcal{I}|$, following the proof of Cygan et al. (2015, Lemma 2.2), we can construct the kernel by simply running Algorithm 2 and limit the maximum number of (unique) bases to traverse to $|\mathcal{I}|$.

First, since $|\mathcal{I}| = \mathcal{O}(N \cdot d)$ is the number of bits required to represent all points,\footnote{\url{https://en.wikipedia.org/wiki/Computational_complexity_theory#Measuring_the_size_of_an_instance.}} which is polynomial to $N$ and $d$, the constructed algorithm runs in polynomial time. Meanwhile, of-the-art FPT algorithm\footnote{\url{https://github.com/ZhipengCai/Demo---MAXCON-hardness.}} (Chin et al. 2015) (called A*-tree search in the rest of this paper) and the two bounds. Specifically, the exact value of the bound for Algorithm 1 is the maximum number of (repeated) nodes traversed during tree search, which is

$$
\frac{d + 1)^{(o+1)} - 1}{d} = 1 + (d + 1) + \cdots + (d + 1)^{o}
$$

(27)

And the bound for Algorithm 2 is $e \cdot (o + 1)^{(d+1)}$ (see the proof of Matoušek (1995, Theorem 2.3 (i)) for details), where $e$ is the exponential constant.

The synthetic data was generated randomly with $d \in \{1, \ldots, 10\}$ and $o \in \{1, \ldots, 20\}$. For each fixed $d$ and $o$, we generated 100 problem instances with $N = 200$. To control $o$, we first sampled $N$ points from a random linear model. Then we added noise uniformly distributed in $[-\epsilon, \epsilon]$ to the $b_i$ channel of all randomly selected inliers. Finally, outliers were created by adding truncated Gaussian noise distributed between $[-\infty, -\epsilon) \cup (\epsilon, \infty]$.

As shown in Fig. 4, the FPT bound for Algorithm 2 was much tighter than the one for Algorithm 1. Both FPT bounds were valid and the A*-tree search algorithm generated much smaller number of unique bases for all problem instances. This indicates that in practice, MAXCON can be solved much faster than the theoretical bounds.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig4.png}
\caption{Two FPT bounds and the actual number of unique bases generated by the A*-tree search. For each fixed $o$ and $d$, both the maximum and the average number of unique bases generated over all 100 synthetic data are shown (Color figure online).}
\end{figure}

3.6.1 Empirical Tightness of the FPT Bounds

To show the validity and tightness of the two FPT bounds derived in Sect. 3.3, we test on synthetic data the difference between the number of unique nodes generated by the state-

\textbf{c1} If Algorithm 2 solves MAXCON before traversing $|\mathcal{I}|$ bases, the global optimal solution can be returned as the output of the kernel.

\textbf{c2} Otherwise, we have $g(o, d) = \mathcal{O}((o + 1)^{(d+1)}) \leq |\mathcal{I}|$.

Hence, we can directly return the input itself as the output of the kernel since its size is already smaller than $g(o, d)$.

Therefore, the output size $|\mathcal{I}'|$ of this constructed algorithm is less than $g(o, d)$.

\begin{itemize}
\item Though unlike the polynomial sized kernel (Cygan et al. 2015, Chapter 2) where $g(o, d)$ is polynomial and is usually derived from some heuristics, the kernel constructed from FPT algorithms cannot reduce the size of the problem, it indicates a way to test the practicality of an FPT algorithm. Namely, we can run the constructed kernel and see how often case c1 happens in practice. If it happens for most of the problem instances, we can still hope to solve MAXCON exactly in polynomial time. The result of this test on real-world data will be demonstrated later in Sect. 3.6.2.
\end{itemize}
3.6.2 Is Exact Solution Practical?

Since the A*-tree search was much faster than the worst case runtime, a natural question to ask is that “is it fast enough to solve MAXCON exactly in practice?” To answer this question, we used the kernel constructed in Sect. 3.5. Specifically, we ran the corresponding kernel of A*-tree search and computed the frequency that case \( c_1 \) (in the proof of Lemma 7) happened. Theoretically, the size of the problem \( |I| \), which was the maximum number of nodes allowed to be generated by the kernel, should be the number of bits required to represent the input data, which was \( 32 \cdot N \cdot d \) following the single-precision floating-point format.

The experiment computed the linearised fundamental matrix (Hartley and Zisserman 2003) \( (d = 8 \) and the rank-2 constraint was ignored) for a sequence of image pairs from the KITTI odometry dataset (Geiger et al. 2012). For each image pair, the input to the kernel was a set of SIFT (Lowe 1999) correspondences (with \( N \) varied from 150 to 400) computed by VLFeat toolbox (Vedaldi et al. 2010). The inlier threshold was set to 0.04.

To examine the practicality of A*-tree search on image pairs with both small and large relative motions, the experiment was executed on two sequences of image pairs. In the first sequence, each image pair was made of two consecutive frames, while in the second sequence, the two images in each pair were 5 frames away. The example inputs generated from both 1-frame-gap data and 5-frame-gap data are provided in Fig. 5.

As shown in Table 1, the kernel terminated quickly and \( c_1 \) happened for all 1000 image pairs with 1 frame gap. This was because the relative motion was too small (see Fig. 5a for an example) and almost all SIFT correspondences were inliers \( (\omega = 0 \) for most pairs), as shown by the deepest tree level \( (\omega = 0 \) if \( c_1 \) happens) traversed by the kernel.

On the other hand, when the frame gap = 5 and the relative motion became large, \( c_1 \) never happened. To estimate the difference between \( \omega \) and the deepest tree level traversed by the kernel \( (\leq 16 \) as shown in Table 1), we computed the upper bound \( \bar{\omega} \) of \( \omega \) using the state-of-the-art deterministic local method (Cai et al. 2018), whose results were shown to be usually close to optimal. As shown in Table 1, the average value of \( \bar{\omega} \), \( \bar{\omega}_{\text{mean}} = 70.29 \), was much larger than 16. Hence, the kernel was still far from finding the optimal solution when terminated. Note that, the kernel took in average more than 1 h to terminate, which was unacceptable already even though it

### Table 1

Practicality of A*-tree search for linearised fundamental matrix estimation. The data was generated using the images in the “00” sequence of the KITTI odometry dataset. \( \bar{\omega}_{\text{max}} \) and \( \bar{\omega}_{\text{mean}} \) were respectively the mean and max of \( \bar{\omega} \) among all data with 5 frame gap. Note that when \( c_1 \) happened, the deepest tree level traversed by the kernel was exactly \( \omega \).

| Frame gap | 1 | 5 |
|-----------|---|---|
| Number of tested image pairs | 1000 | 200 |
| Number of \( c_1 \) | 1000 | 0 |
| Average runtime (s) of the kernel | 0.03 | 4340.70 |
| Max of the deepest tree level traversed by the kernel | 8 | 16 (\( \bar{\omega}_{\text{max}} = 114 \)) |
| Mean of the deepest tree level traversed by the kernel | 0.04 | 13.15 (\( \bar{\omega}_{\text{mean}} = 70.29 \)) |

---

\( \bar{\omega}_{\text{max}} \) and \( \bar{\omega}_{\text{mean}} \) were respec-

\[\text{tively the mean and max of } \bar{\omega} \text{ among all data with 5 frame gap. Note that when } c_1 \text{ happened, the deepest tree level traversed by the kernel was exactly } \omega.\]

https://en.wikipedia.org/wiki/Single-precision_floating-point_format.
could find the optimal solution at the end. Therefore, solving MAXCON exactly is unlikely to be practical in general.

4 Approximability

Given the inherent intractability of MAXCON, it is natural to seek recourse in approximate solutions. However, this section shows that it is not possible to construct PTAS (Vazirani 2001) for MAXCON.

Our development here is inspired by Amaldi and Kann (1995, Sec. 3.2). First, we define our source problem: given a set of $k$ Boolean variables $\{v_j\}_{j=1}^k$, a literal is either one of the variables, e.g., $v_j$, or its negation, e.g., $\neg v_j$. A clause is a disjunction over a set of literals, i.e., $v_1 \lor \neg v_2 \lor v_3$. A truth assignment is a setting of the values of the $k$ variables. A clause is satisfied if it evaluates to true.

Problem 5 [MAX-2SAT] Given $M$ clauses $\mathcal{K} = \{\mathcal{K}_i\}_{i=1}^M$ over $k$ Boolean variables $\{v_j\}_{j=1}^k$, where each clause has exactly two literals, what is the maximum number of clauses that can be satisfied by a truth assignment?

MAX-2SAT is APX-hard, meaning that there are no algorithms that run in polynomial time that can approximately solve MAX-2SAT up to a desired error ratio. Here, we show an L-reduction from MAX-2SAT to MAXCON, which unfortunately shows that MAXCON is also APX-hard.

4.1 Generating the Input Data

Given an instance of MAX-2SAT with clauses $\mathcal{K} = \{\mathcal{K}_i\}_{i=1}^M$ over variables $\{v_j\}_{j=1}^k$, let each clause $\mathcal{K}_i$ be represented as $\{\pm v_{a_i}\} \lor \{\pm v_{b_i}\}$, where $a_i, b_i \in \{1, \ldots, k\}$ index the variables that exist in $\mathcal{K}_i$, and ± here indicates either a “blank” (no negation) or ¬ (negation). Define

\[
\text{sgn}(\alpha_i) = \begin{cases} +1 & \text{if } v_{a_i} \text{ occurs without negation in } \mathcal{K}_i, \\ -1 & \text{if } v_{a_i} \text{ occurs with negation in } \mathcal{K}_i; \end{cases}
\]

(28)

similarly for sgn($\beta_i$). Construct the input data for MAXCON as

\[
\mathcal{D}_\mathcal{K} = \{(a_i^p, b_i^p)\}_{i=1}^M, \quad \forall p = \{1, 6\},
\]

(29)

where there are six measurements for each clause. Namely, for each clause $\mathcal{K}_i$,

- $a_i^1$ is a $k$-dimensional vector of zeros, except at the $\alpha_i$-th and $\beta_i$-th elements where the values are respectively sgn($\alpha_i$) and sgn($\beta_i$), and $b_i^1 = 2$.
- $a_i^2 = a_i^1$ and $b_i^2 = 0$.
- $a_i^3$ is a $k$-dimensional vector of zeros, except at the $\alpha_i$-th element where the value is sgn($\alpha_i$), and $b_i^3 = -1$.
- $a_i^4 = a_i^3$ and $b_i^4 = 1$.
- $a_i^5$ is a $k$-dimensional vector of zeros, except at the $\beta_i$-th element where the value is sgn($\beta_i$), and $b_i^5 = -1$.
- $a_i^6 = a_i^5$ and $b_i^6 = 1$.

The number of measurements $N$ in $\mathcal{D}_\mathcal{K}$ is $6M$.

4.2 Setting the Inlier Threshold

Given a solution $\mathbf{x} \in \mathbb{R}^k$ for MAXCON, the six input measurements associated with $\mathcal{K}_i$ are inliers under these conditions:

- $(a_i^1, b_i^1)$ is an inlier $\iff |\text{sgn}(\alpha_i)x_{a_i} + \text{sgn}(\beta_i)x_{b_i} - 2| \leq \epsilon$, (30)
- $(a_i^2, b_i^2)$ is an inlier $\iff |\text{sgn}(\alpha_i)x_{a_i} + \text{sgn}(\beta_i)x_{b_i}| \leq \epsilon$, (31)
- $(a_i^3, b_i^3)$ is an inlier $\iff |\text{sgn}(\alpha_i)x_{a_i} + 1| \leq \epsilon$, (32)
- $(a_i^4, b_i^4)$ is an inlier $\iff |\text{sgn}(\alpha_i)x_{a_i} - 1| \leq \epsilon$, (33)
- $(a_i^5, b_i^5)$ is an inlier $\iff |\text{sgn}(\beta_i)x_{b_i} + 1| \leq \epsilon$,
- $(a_i^6, b_i^6)$ is an inlier $\iff |\text{sgn}(\beta_i)x_{b_i} - 1| \leq \epsilon$.

where $x_{a_i}$ is the $\alpha$-th element of $\mathbf{x}$. Observe that if $\epsilon < 1$, then at most one of (30), one of (31), and one of (32) can be satisfied. The following result establishes an important condition for L-reduction.

Lemma 8 If $\epsilon < 1$, then

\[
\text{OPT(MAXCON)} \leq 6 \cdot \text{OPT(MAX-2SAT)},
\]

(33)

OPT(MAX-2SAT) is the maximum number of clauses that can be satisfied for a given MAX-2SAT instance, and OPT(MAXCON) is the maximum achievable consensus for the MAXCON instance generated under our reduction.

Proof If $\epsilon < 1$, for all $\mathbf{x}$, at most one of (30), one of (31), and one (32), can be satisfied, hence OPT(MAXCON) cannot be greater than $3M$. For any MAX-2SAT instance with $M$ clauses, there is an algorithm (Johnson 1974) that can satisfy at least $\lceil \frac{M}{3} \rceil$ of the clauses, thus OPT(MAX-2SAT) $\geq \lceil \frac{M}{3} \rceil$. This leads to (33). \hfill \square

Note that, if $\epsilon < 1$, rounding $\mathbf{x}$ to its nearest bipolar vector (i.e., a vector that contains only $-1$ or $1$) cannot decrease the
consensus w.r.t. $D_K$. It is thus sufficient to consider $x$ that are bipolar in the rest of this section.

Intuitively, $x$ is used as a proxy for truth assignment: setting $x_j = 1$ implies setting $v_j = \text{true}$, and vice versa. Further, if one of the conditions in (30) holds for a given $x$, then the clause $K_i$ is satisfied by the truth assignment. Hence, for $x$ that is bipolar and $\epsilon < 1$,

$$\Psi_\epsilon(x \mid D_K) = 2M + \sigma,$$

where $\sigma$ is the number of clauses satisfied by $x$. This leads to the final necessary condition for L-reduction.

**Lemma 9** If $\epsilon < 1$, then

$$|\text{OPT}(\text{MAX-2SAT}) - \text{SAT}(t(x))|$$

$$= |\text{OPT}(\text{MAXCON}) - \Psi_\epsilon(x \mid D_K)|,$$

where $t(x)$ returns the truth assignment corresponding to $x$, and $\text{SAT}(t(x))$ returns the number of clauses satisfied by $t(x)$.

**Proof** For any bipolar $x$ with consensus $2M + \sigma$, the truth assignment $t(x)$ satisfies exactly $\sigma$ clauses. Since the value of $\text{OPT}(\text{MAXCON})$ must take the form $2M + \sigma^*$, then $\text{OPT}(\text{MAX-2SAT}) = \sigma^*$. The condition (35) is immediately seen to hold by substituting the values into the equation. \(\square\)

We have demonstrated an L-reduction from MAX-2SAT to MAXCON, where the main work is to generate $D_K$ in linear time. The function $t$ also takes linear time to compute. Setting $\epsilon < 1$ completes the reduction.

**Theorem 6** MAXCON is APX-hard.

**Proof** Since MAX-2SAT is APX-hard, by the above L-reduction, MAXCON is also APX-hard. \(\square\)

See Sect. 1.1 for the implications of Theorem 6.

## 5 Conclusions and Future Work

Given the fundamental difficulty of consensus maximisation as implied by our results (see Sect. 1.1), it would be prudent to consider alternative paradigms for optimisation, e.g., deterministically convergent heuristic algorithms (Le et al. 2017; Purkait et al. 2017; Cai et al. 2018) or preprocessing techniques (Svärmt et al. 2014; Parra Bustos and Chin 2015; Chin et al. 2016).

**Acknowledgements** This work was supported by ARC Grant DP160103490.

References

Amaldi, E., & Kann, V. (1995). The complexity and approximability of finding maximum feasible subsystems of linear relations. *Theoretical Computer Science*, 147, 181–210.

Aronov, B., & Har-Peled, S. (2008). On approximating the depth and related problems. *SIAM Journal on Computing*, 38(3), 899–921.

Bazin, J. C., Li, H., Kweon, I. S., Demonceaux, C., Vasseur, P., & Ikeuchi, K. (2013). A branch-and-bound approach to correspondence and grouping problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7), 1565–1576.

Ben-David, S., Eiron, N., & Simon, H. (2002). The computational complexity of densest region detection. *Journal of Computer and System Sciences*, 64(1), 22–47.

Bernholt, T. (2005). Robust estimators are hard to compute. *Technical report 52*, Technische Universität.

Cai, Z., Chin, T. J., Le, H., Suter, D. (2018). Deterministic consensus maximization with biconvex programming. In *European conference on computer vision (ECCV)*.

Campbell, D., Petersson, L., Kneip, L., Li, H. (2017). Globally-optimal inlier set maximisation for simultaneous camera pose and feature correspondence. In *IEEE international conference on computer vision (ICCV)*.

Cheney, E. W. (1966). *Introduction to approximation theory*. New York: McGraw-Hill.

Chin, T. J., Cai, Z., Neumann, F. (2018). Robust fitting in computer vision: easy or hard? In *European conference on computer vision (ECCV)*.

Chin, T. J., Kee, Y. H., Eriksson, A., Neumann, F. (2016). Guaranteed outlier removal with mixed integer linear programs. In *IEEE computer society conference on computer vision and pattern recognition (CVPR)*.

Chin, T. J., Purkait, P., Eriksson, A., Suter, D. (2015). Efficient globally optimal consensus maximisation with tree search. In *IEEE computer society conference on computer vision and pattern recognition (CVPR)*.

Choi, S., Kim, T., Yu, W. (2009). Performance evaluation of RANSAC family. In *British machine vision conference (BMVC)*.

Cyganski, M., Fomin, F. V., Kowalik, L., Lokshin, D., Marx, D., Pilipczuk, M., et al. (2015). *Parameterized algorithms* (Vol. 3). Berlin: Springer.

Downey, R. G., & Fellows, M. R. (1999). *Parametrized complexity*. New York: Springer.

Enqvist, O., Ask, E., Kahli, F., & Åström, K. (2012). Robust fitting for multiple view geometry. In *European conference on computer vision (ECCV)*.

Enqvist, O., Ask, E., Kahli, F., & Åström, K. (2015). Tractable algorithms for robust model estimation. *International Journal of Computer Vision*, 112(1), 115–129.

Erickson, J., Har-Peled, S., & Mount, D. M. (2006). On the least median square problem. *Discrete & Computational Geometry*, 36(4), 593–607.

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

Fukuda, K., Liebling, T. M., & Margot, F. (1997). Analysis of backtrack algorithms for listing all vertices and all faces of a convex polyhedron. *Computational Geometry*, 8, 1–12.

Garey, M. R., & Johnson, D. S. (1990). *Computers and intractability: a guide to the theory of NP-completeness*. New York: WH Freeman & Co.

Geiger, A., Lenz, P., Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on computer vision and pattern recognition (CVPR)*.
Giannopoulos, P., Knauer, C., Rote, G.: The parameterized complexity of some geometric problems in unbounded dimension. In *International workshop on parameterized and exact computation (IWPEC)* (2009).

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions Systems Science and Cybernetics, 4*(2), 100–107.

Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge: Cambridge University Press.

Johnson, D. S. (1974). Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences, 9*, 256–278.

Johnson, D. S., & Preparata, F. P. (1978). The densest hemisphere problem. *Theoretical Computer Science, 6*, 93–107.

Le, H., Chin, T. J., Suter, D. (2017). An exact penalty method for locally convergent maximum consensus. In *IEEE computer society conference on computer vision and pattern recognition (CVPR)*.

Li, H. (2009). Consensus set maximization with guaranteed global optimality for robust geometry estimation. In *IEEE international conference on computer vision (ICCV)*.

Lowe, D. G. (1999) Object recognition from local scale-invariant features. In *The proceedings of the seventh IEEE international conference on computer vision, 1999* (Vol. 2, pp. 1150–1157). IEEE

Matoušek, J. (1995). On geometric optimization with few violated constraints. *Discrete and Computational Geometry, 14*(4), 365–384.

Meer, P. (2004). Robust techniques for computer vision. In G. Medioni & S. B. Kang (Eds.), *Emerging topics in computer vision*. New York: Prentice Hall.

Parra Bustos, A., Chin, T. J., Suter, D.: Fast rotation search with stereographic projections for 3d registration. In *IEEE computer society conference on computer vision and pattern recognition (CVPR)* (2014)

Parra Bustos, A., Chin, T. J. (2015). Guaranteed outlier removal for rotation search. In *IEEE international conference on computer vision (ICCV)*.

Purkait, P., Zach, C., Eriksson, A. (2017) Maximum consensus parameter estimation by reweighted L1 methods. In *Energy minimization methods in computer vision and pattern recognition (EMMCVPR)*.

Raguram, R., Chum, O., Pollefeys, M., Matas, J., & Frahm, J. M. (2013). USAC: A universal framework for random sample consensus. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(8), 2022–2038.

Svárm, L., Enqvist, O., Oskarsson, M., Kahl, F. (2014). Accurate localization and pose estimation for large 3D models. In *IEEE computer society conference on computer vision and pattern recognition (CVPR)*

Tran, Q. H., Chin, T. J., Chojnacki, W., & Suter, D. (2014). Sampling minimal subsets with large spans for robust estimation. *International Journal of Computer Vision (IJCV), 106*(1), 93–112.

Vazirani, V. (2001). *Approximation algorithms*. Berlin: Springer.

Vedaldi, A., Fulkerson, B. (2010). Vlfeat: An open and portable library of computer vision algorithms. In *Proceedings of the 18th ACM international conference on Multimedia* (pp. 1469–1472). ACM.

Yang, J., Li, H., Jia, Y.: Optimal essential matrix estimation via inlier-set maximization. In *European conference on computer vision (ECCV)*

Zheng, Y., Sugimoto, S., Okutomi, M.: Deterministically maximizing feasible subsystems for robust model fitting with unit norm constraints. In *IEEE computer society conference on computer vision and pattern recognition (CVPR)* (2011)

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.