Consensus Maximisation Using Influences of Monotone Boolean Functions

Ruwan Tennakoon[†], David Suter[‡], Erchuan Zhang[‡], Tat-Jun Chin[§] and Alireza Bab-Hadiashar[†]

[†]RMIT University, Melbourne Australia. [‡]Edith Cowen University, Perth Australia.

[§]University of Adelaide, Adelaide Australia.

{ruwan.tennakoon, abh}@rmit.edu.au, {d.suter, erchuan.zhang}@ecu.edu.au,

tat-jun.chin@adelaide.edu.au

Abstract

Consensus maximisation (MaxCon), which is widely used for robust fitting in computer vision, aims to find the largest subset of data that fits the model within some tolerance level. In this paper, we outline the connection between MaxCon problem and the abstract problem of finding the maximum upper zero of a Monotone Boolean Function (MBF) defined over the Boolean Cube. Then, we link the concept of influences (in a MBF) to the concept of outlier (in MaxCon) and show that influences of points belonging to the largest structure in data would generally be smaller under certain conditions. Based on this observation, we present an iterative algorithm to perform consensus maximisation. Results for both synthetic and real visual data experiments show that the MBF based algorithm is capable of generating a near optimal solution relatively quickly. This is particularly important where there are large number of outliers (gross or pseudo) in the observed data.

1. Introduction

The popular Maximum Consensus (MaxCon) criterion for robust fitting (as typified by that of RANSAC [11]), seeks the maximum sized *feasible* set. Here feasible means that all data points belonging to a "structure" (the inlier set) fits its model within a tolerance level.

Given a set of n data points $\mathcal{D} = \{\mathbf{p}_i\}_{i=1}^n$ and a tolerance level ϵ, the MaxCon criterion for robust fitting can be written as:

$$\max_{\theta, \mathcal{I} \subseteq \mathcal{D}} |\mathcal{I}|$$

$$\text{subject to } r_{\mathbf{p}_i}(\theta) \leq \epsilon \quad \forall \mathbf{p}_i \in \mathcal{I}$$

where $r_{\mathbf{p}_i}(\theta)$ is the distance of $\mathbf{p}_i$ from the model θ.

A subset $\mathcal{I}$ can be represented by length-n bit-vector, $\mathbf{x}$, where the i'th position of bit vector denote the inclusion ($x_i = 1$) or exclusion ($x_i = 0$) of the data point i.

The above shows that, each subset can be represented by a vertex of the n-dimensional Boolean Cube. Therefore, any statement about which of the subsets are feasible, is a statement on the evaluation of a Boolean function over the n-dimensional Boolean Cube. This Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$, outputs 1 (for infeasible), or 0 (for feasible) for any vertex of the Boolean Cube. Since MaxCon is a search for the maximum sized feasible subset of the data, it is inherently a search on the Boolean Cube.

Such a view immediately opens the huge theory and very many associated mathematical tools, developed during the study of Boolean functions; for the purposes of the analysis of the MaxCon problem and for devising algorithms to solve this problem.

Also, there is an additional significant observation. The Boolean function associated with a MaxCon problem belongs to a special class of Boolean functions called the Monotone Boolean Functions (MBF) [15].

Definition 1.1. A MBF of n variables is a mapping $f : \{0,1\}^n \rightarrow \{0,1\}$ such that $\alpha \prec \beta$ implies that $f(\alpha) \leq f(\beta)$. Here we apply the natural ordering relation on n-dimensional Boolean Cube: β follows α ($\alpha \prec \beta$) if for any i, the equality $\alpha_i \leqslant \beta_i$ is satisfied.

This is easy to see: If a subset is feasible, then adding more data to that subset can only move the function towards infeasibility and once infeasible adding more points will not change the subset back to feasible. Likewise, deleting points from a subset can move only towards feasible. MaxCon, when viewed in the above sketched framework, is nothing more than the search for the *maximum upper zero* (see the definition below) of the above-mentioned monotone infeasibility function.

Definition 1.2 (Upper zero of a MBF). Upper zero of the MBF $f(\cdot)$ is a vertex α for which $f(\alpha) = 0$ and, for all $\alpha \prec \beta$ the relation $f(\beta) = 1$ is satisfied.

Definition 1.3 (Maximum Upper zero of a MBF [16]). Maximum Upper zero of the MBF $f(\cdot)$ is a vertex α for
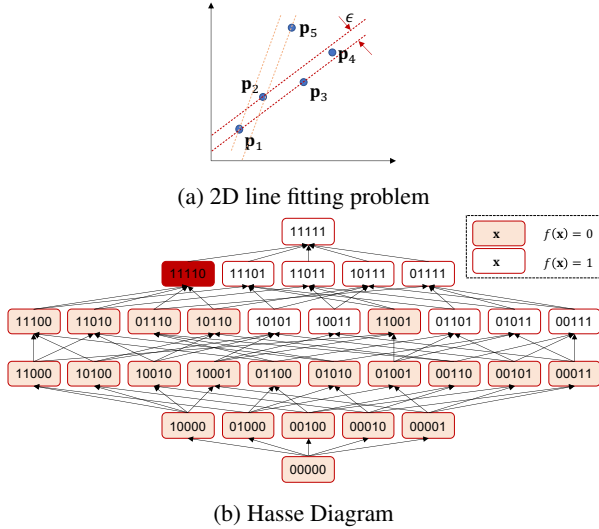
(a) 2D line fitting problem

(b) Hasse Diagram

Figure 1: (a) An example of a 2D line fitting problem with 5 data points together with (b) the associated Boolean Cube and MBF - represented in Hasse diagram format. The 5D Boolean Cube is drawn flattened onto 2D and oriented so that the higher up a vertex appears, the larger is the number of 1's in the coordinates. A Boolean Function maps the Boolean cube to 0 or 1. We illustrate by colouring: "white" nodes map to 1, coloured nodes map to 0. This example is a *Monotone* Boolean Function since moving up the picture, the value of the function only ever increases, it *never* translates in the opposite direction. The red node is the maximum upper zero in this example.

which $f(\alpha) = 0$ and, for all $\|\beta\|_1 > \|\alpha\|_1$ the relation $f(\beta) = 1$ is satisfied. Here, $\|\alpha\|_1 = \sum_i \alpha_i$.

A simple example in Figure 1 illustrates the concepts involved. The figure shows that, along paths going up the Hasse diagram (a directed version of the Boolean cube), the function can only stay constant or increase (never decrease). The vertex "11110" (representing the subset $\{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4\}$) is the MaxCon solution for this problem as it is the highest feasible subset in the Hasse diagram (maximum upper zero). The vertex "11001" is an example of an upper zero as there are no feasible nodes that follows it ($\succ$ '11001') on the Hasse diagram (can be seen as a local optimum).

Whilst Monotone Boolean Functions (MBFs) have been extensively studied for a variety of application domains, including learning theory [1] (and computer vision is, these days, highly dominated by learning style approaches), there appears to be relatively little attention to MBFs in computer vision: and more specifically in model-based computer vision. Reference [27] appears to be a recent exception - but even this is tackling very different considerations from our main areas of interest (namely, [27] is concerned with ef-

ficient Conditional Random Field (CRF) calculations, and CRF modelling is very different to the geometric modelling we refer to).

In this paper, we present a novel view point on the Max-Con problem using the Monotone Boolean Functions theory and use that to develop some efficient solutions. For this purpose, we concentrate on a property of all Boolean Functions, called Influence [22] (definition in Section 2.1); a property that has a particular relationship with the Fourier Transform of a Boolean Function *when that Boolean Function is Monotone*. In summary our main contributions include:

1. We provide a precise (and abstract) definition of the MaxCon problem in terms of finding the maximum upper zero of a Monotone Boolean Function defined over the Boolean Cube where, vertices correspond to subsets of data and the output of the MBF correspond to whether the subset can be feasibly covered by a model with set tolerance.

2. We link the concept of influences in a MBF to the concept of outlier in MaxCon and provide theoretical analysis to show that influences of points belonging to the largest structure in data would be smallest under "ideal" conditions (defined in Section 3).

3. Based on the above analysis, we derive a greedy algorithm that searches for the maximum upper zero of a MBF to efficiently solve the MaxCon problem.

2. Background

2.1. Boolean Functions and Influences

For any Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$, the probability that the i'th coordinate in a random length-n binary vector, $\mathbf{x}$, "affects" the output of the function is defined as the influence [22]:

$$\text{Inf}_i[f] = \Pr_{\mathbf{x} \sim \{0,1\}^n} \left[f(\mathbf{x}) \neq f(\mathbf{x}^{\oplus i}) \right] \quad (1)$$

where $\mathbf{x}^{\oplus i}$ is equal to $\mathbf{x}$ with the i'th coordinate flipped. For a MBF, the influence of the i'th coordinate is equal to the degree-1 Fourier coefficient [22].

The most obvious strategy for influence approximation would be to uniformly sample the cube.

$$\text{Inf}_i[f] = \mathbb{E}_{\mathbf{x} \sim \{0,1\}^n} \delta \left[f(\mathbf{x}) \neq f(\mathbf{x}^{\oplus i}) \right] \quad (2)$$

where $\delta[\cdot] = 1$ if the condition inside is true (0 otherwise). However, in Section 4.1, we empirically show that this may not be the most efficient when it comes to solving MaxCon. A better approach would be to use the notion of weighted influence [9]. In such an approach one typically defines a Bernoulli measure $\mu_q(\mathbf{x})$ over the vertices of the Boolean

Cube. Operationally, this can correspond to sampling by selecting to "turn bits i" independently and with probability q. Uniform sampling corresponds to $q = 0.5$. Sampling with low q concentrates the measure towards the bottom of the cube (small Hadamard norm) and high q towards the top. The influences under μ_q distribution can now be defined as:

$$\text{Inf}_i^{(q)}[f] = \underset{\mathbf{x} \sim \mu_q(\mathbf{x})}{\mathbb{E}} \delta\left[f(\mathbf{x}) \neq f\left(\mathbf{x}^{\oplus i}\right)\right]. \qquad (3)$$

The influences under different q values are analysed further in Section 4.1.

2.2. Determining feasibility and the concept of a basis

Searching for the upper zeros of a MBF will require the evaluation of the function on a given subset. In the case of MaxCon, this involves finding if a given subset is feasible or infeasible. The feasibility/infeasibility of a subset can be obtained efficiently via solving the following minmax problem and checking if the resulting objective value is within ϵ [14, 5]:

$$\min_\theta \max_i r_{\mathbf{p}_i}(\theta) \qquad (4)$$

The above can be solved exactly for cases where $r_{\mathbf{p}_i}(\theta)$ is quasiconvex using algorithms based on bisection [10]. The solution of equation (4) for a subset $\mathcal{I}^{(j)}$, will return three quantities: the optimal objective value $g^{(j)}$, basis $\mathcal{B}^{(j)}$ and the fitted model $\theta^{(j)}$. A basis for a set, is a small subset such that the value of the final objective on that subset is equal to the final objective on the whole set [7]. Intuitively, a basis in the MaxCon setting is a subset of points "that prevents the enclosing structure from shrinking further". The cardinality of the basis in MaxCon is connected to the concept of "combinatorial dimension" and we denote it by $p + 1$ (p is the minimum number of data points required for the unique determination of the model).

2.3. Related work in computer vision

Consensus maximisation is widely used for robust fitting in computer vision. Early work mostly focused on solving the consensus maximisation using randomised hypothesise-and-test techniques [11, 18, 26]. More recently, some focus has shifted towards finding globally optimal solutions [5, 33, 2, 19, 4, 20]. As MaxCon is known to be NP-hard, the run time of the global methods scale exponentially in the general case [6]. This has led to the search for deterministic and/or near optimal algorithms [3, 25, 17, 32].

To the best of our knowledge, ours is the first work in computer vision that explicitly poses the MaxCon problem in terms of finding the maximum upper zero of a MBF. An investigation into Quantum Computing calculations of Influence for Maxcon [8] was recently published. Moreover, there are several works that are related, in some sense, to
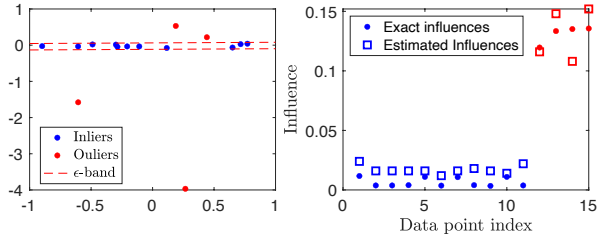

Figure 2: An example 2D line-fitting problem and the corresponding influences. The data generation procedure is described in detail in Section 4.1. Influences are estimated using equation (3) with $m = 1000$ and $q = 0.5$. Data points are sorted such that inliers comes before outliers.

the proposed framework. Though not expressed that way in the original works, the A* "tree searches" of [2] and [5] are in fact searches on this cube (where nodes reached by different paths become *repeated nodes* in the tree constructed by starting from the top of the Hasse Diagram). In this context, it is interesting to note that RANSAC (and it's many derivatives) search amongst subsets towards the bottom of the Hasse Diagram (minimal sized subsets or p-sized); and use these to "index" up to higher subsets by greedily including all data points that fit within tolerance of the model found on the p-subset. Furthermore, the methods that use the L_∞-Norm (equation 4) for outlier removal in the context of geometric fitting, can also be seen as traversing the Boolean cube [28, 23, 20, 24]. However, the above methods do not explicitly utilize the theory and tools associated with MBF.

3. Proposed Method

As described in the introduction, MaxCon can be seen as finding the *maximum upper zero* of a MBF. This paper is devoted to the exploitation of information embedded in the Influence function to find the maximum upper zero, efficiently. To explain the method, we first note that the influence of an inlier data point is likely to be smaller than the influence of an outlier data point. An example of this relationship for a 2D line-fitting problem is shown in Figure 2. The essential and intuitive reason is that inclusion of an outlier, into a feasible subset, most likely turns that subset infeasible (thereby "influencing" the function). In contrast adding an inlier leaves the set feasible. It is the "breaking" or "creation" of an infeasible $p + 1$ sized basis that is responsible for flipping the outcome of the MBF and therefore the addition/deletion of an outlier triggers this more than that of an inlier. Below, we prove that the above property of influences hold under "ideal" conditions. The proofs of theorems in the following text are available in the supplementary materials.

3.1. Ideal single structure case

We first formally define an ideal structure in data. Let $L_k := \{\mathbf{x} \in \{0,1\}^n : \|\mathbf{x}\|_1 = k\}$ be the level k in the n-dimensional Boolean cube and, $L_{\leq k}$ the levels below $k+1$.

Definition 3.1. Given a monotone Boolean function f, for $\mathbf{x}^k \in L_k$ ($p < k \leq n$), f is called ideal with respect to $\mathbf{x}^k$ if

$$f(\mathbf{x}) = \begin{cases} 0 & \forall \mathbf{x} \in B_{\mathbf{x}^k} \cup L_{\leq p} \\ 1 & \text{others} \end{cases} \tag{5}$$

where $B_{\mathbf{x}^k} = \left\{ \mathbf{x} \in \{0,1\}^n : d\left(\mathbf{x}, \mathbf{x}^k\right) = l, \mathbf{x} \in L_{k-l} \right\}$ for all $0 \leq l \leq k - p - 1$, is the Boolean sub-cube determined by $\mathbf{x}^k$. Here, $d(\cdot, \cdot)$ is the Hamming distance.

Intuitively, an ideal structure has only one structure: points are either inlier or outliers to that single structure, and no other structure exists other than the trivial with p or less points.

Theorem 3.1. If a monotone Boolean function f is ideal with respect to $\mathbf{x}^k \in L_k$, then

$$\text{Inf}_i[f] = \begin{cases} \dfrac{C_p^{n-1} - C_p^{k-1}}{2^n} & \text{if } i \text{ is inlier} \\ \dfrac{C_p^{n-1} + \sum_{l=p+1}^{k} C_l^k}{2^n} & \text{if } i \text{ is outlier} \end{cases} \tag{6}$$

Corollary 3.1.1.

$$\text{Inf}_j[f] - \text{Inf}_i[f] = \frac{1}{2^n} \sum_{l=p+1}^{k} C_l^k + C_p^{k-1} > 0 \tag{7}$$

where i is any inlier and j is any outlier.

The above shows that for an ideal structure case the influence of an inlier is strictly smaller than the influence of an outlier.

3.2. Ideal K-structure case

In what follows, we generalize the above result to the ideal K-structure ($K \geq 1$), namely there are K upper zeros only and the Boolean sub-cubes determined by these upper zeros are disjoint above level p.

Definition 3.2. Let f be a monotone Boolean function and $\left\{\mathbf{x}^{k_r}\right\}_{r=1}^{K}$ are upper zeros, where $p < k_1 \leq k_2 \leq \cdots \leq k_K \leq n$, then f is called K-ideal with respect to $\left\{\mathbf{x}^{k_r}\right\}_{r=1}^{K}$ if

1) $d\left(B_{\mathbf{x}^{k_i}} \setminus L_{\leq p}, B_{\mathbf{x}^{k_j}} \setminus L_{\leq p}\right) > 0 \quad \forall k_i \neq k_j$

2) $f(\mathbf{x}) = \begin{cases} 0 & \forall \mathbf{x} \in \bigcup_{r=1}^{K} B_{\mathbf{x}^{k_r}} \cup L_{\leq p} \\ 1 & \text{others} \end{cases} \tag{8}$

The first condition states that the sub-cubes determined by $\mathbf{x}^{k_r}$ have no elements in common. Let $\mathcal{S}_{\mathbf{x}^{k_r}}^c$ be the set of inlier (if $c = 1$) or the set of outliers (if $c = 0$) to $\mathbf{x}^{k_r}$. Then we can define

$$\mathcal{S}_{c_1 c_2 \cdots c_K} = \bigcap_{r=1}^{K} \mathcal{S}_{\mathbf{x}^{k_r}}^{c_r} \quad c_r \in \{0,1\} \tag{9}$$

which represent the index set of inlier to structures where bit string $c_1 c_2 \cdots c_K$ is one. For example, $\mathcal{S}_{11\cdots 1}$ is the index set of points that are inliers with respect to all $\mathbf{x}^{k_r}$, and $\mathcal{S}_{00\cdots 0}$ is the index set of points that are outliers with respect to all $\mathbf{x}^{k_r}$.

Theorem 3.2.

$$2^n \text{Inf}_{\mathcal{S}_{c_1 \cdots c_K}}[f] = C_p^{n-1} + \sum_{\substack{c_r=0 \\ 1 \leq r \leq K}} \sum_{l=p+1}^{k_r} C_l^{k_r} - \sum_{\substack{c_r=1 \\ 1 \leq r \leq K}} C_p^{k_r-1} \tag{10}$$

where $\text{Inf}_{\mathcal{S}_{c_1 \cdots c_K}}[f]$ denote the influence $\text{Inf}_i[f]$ of $i \in \mathcal{S}_{c_1 \cdots c_K}$.

Corollary 3.2.1. The influences have the following ordered relationship

$$\forall \alpha, \beta \in \{0,1\}^K, \alpha > \beta \Rightarrow \text{Inf}_{\mathcal{S}_\alpha}[f] < \text{Inf}_{\mathcal{S}_\beta}[f] \tag{11}$$

which means $\text{Inf}_x[f]$ is a real-valued monotone decreasing Boolean function. If any $\mathcal{S}_{\bullet} = \emptyset$ then $\text{Inf}_{\mathcal{S}_{\bullet}}[f]$ is not defined.

Following Corollary 3.2.1 and the Definition 3.2 we can see that the influence of a point that belongs to a structure with a higher upper zero is smaller than that of a point that belongs to a structure with a lower upper zero.

3.3. Finding the maximum upper zero

The above theoretical analysis shows that influences of points belonging to the largest structure in data would be smallest under "ideal" conditions. Thus, in the ideal case, a very simple process of estimation of the influences, followed by thresholding, would immediately lead to the sought after MaxCon solution. However, we need to recognise that:

- The non-ideal case (which applies to all realistic data sets) will be a perturbation away from this, where the influences of inlier and outlier come closer together.

- We can only work with estimated influences, not the actual influences. Thus we need to assume that the estimated influences largely follow the ordering given mathematically in our derivations, and according with the above intuition.

Nonetheless, continuity of behaviour arguments suggest the departure from ideal will be only partial for many data sets; and that relatively standard ways for attacking such scenarios, should still be viable. We empirically verify that this is often true in our experiments.

Using the above intuition ("*influences of points belonging to the largest structure in data would be smallest*") we formulate an algorithm - MBF-MaxCon (Algorithm 1) that starts with the set of all data points and then gradually removes one data point at a time (data point with the largest influence) until the remaining set of points is within the ϵ band. In the algorithm, we also use the notion that: *If there are outliers in a subset of data, then at least one of them should belong to the basis returned by solving equation* (4) [28]. This additional constraint enables us to compute only $p + 1$ influences per elimination of a data point (rather than for all points in subset) which leads to a more efficient algorithm.

The estimation of Influence introduces some noise to the proposed algorithms, and the solution returned by them may not include all the inlier points of a given structure. To partially overcome this, we introduce a local expansion step (Algorithm 2). In this step, starting from the initial solution, at each iteration, the distance-1 upper neighborhood (Hasse Diagram) of the current solution is explored and the current solution is updated if there is any feasible set. This process is repeated until there are no feasible subsets in the distance-1 upper neighborhood. This local update guarantees that the algorithm will find an upper zero (local optimal).

Algorithm 1 MBF-MaxCon - algorithm for finding the maximum consensus set using influences of BMFs.

Require: $\{\mathbf{p}_i\}_{i=1}^n, \epsilon, m, q.$
1: $\mathbf{x} \leftarrow [1, \ldots, 1]_{[1 \times n]}$
2: **repeat**
3: $\quad \mathcal{I}^{(t)} \leftarrow \{i : x_i = 1\}$
4: $\quad$ Solve equation (4) for subset $\mathcal{I}^{(t)}$ and obtain $\mathcal{B}^{(t)}$
5: $\quad$ Estimate $\mathrm{Inf}_i^{(q)}[f] \quad \forall i \in \mathcal{B}^{(t)}$ using equation (3)[1].
6: $\quad e \leftarrow \underset{i \in \mathcal{B}^{(t)}}{\mathrm{argmax}}\, \mathrm{Inf}_i^{(q)}[f]$
7: $\quad x_e \leftarrow 0$
8: **until** $f(\mathbf{x}) = 0$
9: $\mathbf{x} \leftarrow$ Run local expansion step in (Algorithm 2)
10: **return** $\mathcal{I} \leftarrow \{i : x_i = 1\}$

It is important to note that one needs to re-estimate the influences at each iteration, t of algorithm 1. If $\mathrm{Inf}_i^{[t]}[f]$ is the influence of point i at iteration t of the algorithm, then $\mathrm{Inf}_i^{[t]}[f] \neq \mathrm{Inf}_i^{[t-1]}[f]$. This is because the function at level

[1]When estimating influences, one can use the monotonic nature of f to save some computations (In a MBF, the function value before the bit flip has some information regarding the value after). The algorithm used for estimating the influences is available in supplementary materials.

Algorithm 2 Local expansion step.

Require: $\{\mathbf{p}_i\}_{i=1}^n, m$, initial feasible set $\mathbf{x}$.
1: **repeat**
2: $\quad$ updated $\leftarrow$ false
3: $\quad$ **for all** $i : x_i = 0$ **do**
4: $\quad\quad \bar{\mathbf{x}} \leftarrow \mathbf{x}; \bar{x}_i \leftarrow 1$
5: $\quad\quad$ **if** $f(\bar{\mathbf{x}}) = 0$ **then** $\mathbf{x} \leftarrow \bar{\mathbf{x}}$; updated $\leftarrow$ true; break;
6: $\quad$ **end for**
7: **until** updated=false
8: **return** $\mathbf{x}$

---

t is a restricted version of the function at level $t - 1$ and the relationship between influences at different levels is derived in supplementary materials. Another allied intuition is that though, as mentioned before, noise (from both the estimation process and from the departure of the data from that of being ideal) will raise the level of the influence of some inliers (and decrease the values of some outliers) to the point where the estimated influences of some inliers will be above those of some outliers: at each stage we only remove the *largest* influence data point which will be away from the "polluted" data division; and that re-estimation afterwards allows the possibility for the re-estimated influences to be "cleaner".

4. Results

We evaluated the performance of the proposed algorithms, on both synthetic and real data experiments, and compared those with the state-of-the-art techniques. All experiments were executed in MATLAB on a computer with Intel Xeon E5-1650 CPU, 16GB RAM and Ubuntu 16.04 OS. The publicly available codes were used to obtain the results for improved A* tree search[2] (A*-NAPA-DIBP) [2] and Lo-RANSAC[3] [18]. Our implementation is publicly available at https://github.com/RuwanT/MBF-MaxCon.

4.1. Influence estimation

Estimating the influences via equation (3) requires two hyper-parameters: The number of randomly sampled bit-vectors m and, the sampling probability q of the Bernoulli measure $\mu_q(x)$. In this section we explore the effects of the two hyper-parameters on influence estimation using a 2D-line fitting problem. Here, the number of data points (n) is set to 15, out of which 25% would be outliers (n_o). The number of points is chosen to be relatively small as the exact influence calculation time increases exponentially with n. A subset of $(n - n_o)$ randomly selected points (inliers) were then perturbed with uniformly distributed noise in the

[2]Code: MaxConTreeSearch
[3]Code: Consensus-maximization-with-biconvex-programming

range $[-0.1, 0.1]$. The remaining n_o data points (outliers) were then perturbed with uniformly distributed noise from $[-5, -0.1) \cup (0.1, 5]$. The inlier threshold ϵ was set to 0.1 for all the experiments in this section.

First, we compare the estimated influences with the exact influences for different combinations of m and q. The exact influences, $\overline{\text{Inf}}_i[f]$, are computed by taking the expectation in equation (2) over all 2^n vertices of the Boolean cube. The estimation error of all influence values can then be calculated as: $\frac{1}{n} \sum_{i \in [n]} \left(\text{Inf}_i^{(q)}[f] - \overline{\text{Inf}}_i[f] \right)^2$.

The results in Figure 3 shows that, for all q values, the influence estimation error is high when only a few samples are used. However, the error decreases exponentially with increasing the number of samples. We can also see that the estimated influences are closest to their exact values when $q = 0.5$. When q is varied in either direction, the error increases. This is because at $q = 0.5$, the computed values are an unbiased estimate of the influences. Changing the sampling distribution changes the definition of the orthonormal basis and hence introduces a bias [9].

However, in solving MaxCon we do not seek for an unbiased estimate of the influences. What we are after is a definition of influences where the separation between influences of inliers and ourliers are maximum. We used the same synthetic experiment to analyse the separation. Here we define the separation as:

$$\text{Separation} = \min_{i \in \mathcal{D}_{out}} \text{Inf}_i^{(q)}[f] - \min_{j \in \mathcal{D}_{in}} \text{Inf}_j^{(q)}[f] \quad (12)$$

where $\mathcal{D}_{in}$ is the set of inlier data points and $\mathcal{D}_{out} = \mathcal{D} \setminus \mathcal{D}_{in}$. The results in Figure 4 show that the separation increase with m. However, in relation to q, the separation is maximum when the q value is small. Using $q < (p+1)/n$ will also decrease the separation as most samples at this probability will be trivially feasible. In summary, changing q leads to different measures of influence where the order (influences of inliers are smaller than outliers) is much the same, though the separation can vary. In our experiments, we use a q value between $(p+1)/n$ and 0.3.

4.2. Controlled experiments with synthetic data

To study the behaviour of the proposed algorithm under a controlled setting, similar to [2], we conducted experiments on an 8-dimensional robust linear regression problem with synthetically generated data. First, a set of $n = 200$ data points on a randomly instantiated model $\theta \in \mathbb{R}^8$ was generated. The data set was then perturbed using the same process described in Section 4.1 to get a data set that is corrupted by noise and outliers. In our experiments, the number of outliers, n_o, were varied in the range of $[5, 40]$ (upper bound determined by computation time of A*-NAPA-DIBP used for obtaining ground truth inliers/outliers). The error of a method is computed by comparing the cardinality of the
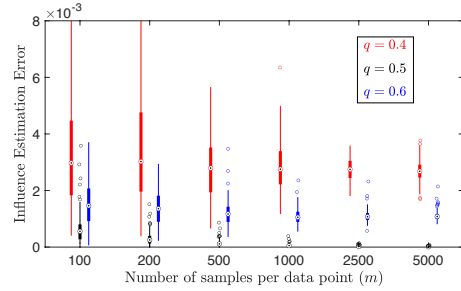

Figure 3: The variation of the influence estimation error with the number of samples used (m) and the sampling probability (q). The figure show the statistics across 100 random runs for each combination.
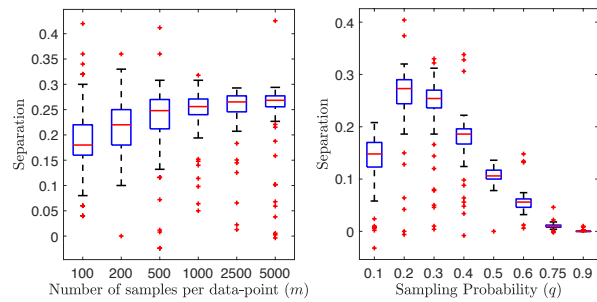

Figure 4: The variation in the separation (between influences of inliers and outliers) with the number of samples used (m) and the sampling probability (q). The figure show the statistics across 100 random runs for each combination.

MaxCon solution found by that method ($|\mathcal{I}_\bullet|$) to the ground truth cardinality obtained using A*-NAPA-DIBP ($|\mathcal{I}_{A^*}|$).

Ablation study: To identify the importance of each component in our overall algorithm, we conduct an ablation study using the above data. Here, we analyse four variants of our algorithm: 1) *MBF-MaxCon-nR*: Simple algorithm where all the influences are computed at the start and the data point with the largest estimated influence is removed iteratively until the remaining subset is feasible (no re-estimation of influences at each iteration). 2) *MBF-MaxCon-nB*: Same as MBF-MaxCon-nR but the influences of all remaining points are recomputed (not just points in the basis) at the end of each iteration. 3) *MBF-MaxCon-nL*: Same as algorithm 1 without the local expansion in line 9. 4) *MBF-MaxCon*: Proposed algorithm 1.

The error of each variant and the computation times are shown in Fig. 5. The results show that re-estimating influences at the end of each iteration has a significant effect on the final outcome (see section 3.3 for explanation). Furthermore, the results show that, on average, the use of both local expansion and basis has helped to move the solutions closer to the global optimal. The results also show that the most significant contribution in terms of the use of basis is
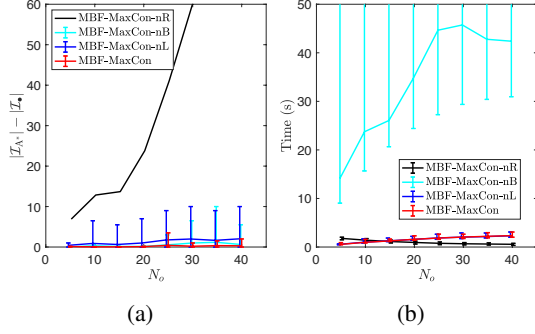
(a)   (b)

Figure 5: Results of the ablation study for 8-dimensional robust linear regression with synthetic data (a) Number of inliers found compared with the global optimal (obtained using A^*) and (b) Variation of computational time with number of outliers. The experiments were repeated 100 times and the error-bars indicate the 0.05-th and 0.95-th percentile.

in computational efficiently.

Comparative analysis: Next, we compare the performance of the proposed method with the most relevant methods in literature: A^*-NAPA-DIBP [2], RANSAC [11] and Lo-RANSAC [18]. Both RANSAC variants were run with the number of RANSAC iterations set to match the computation time of MBF-MaxCon. The computation time for the above methods are shown in Figure 6b. The figure shows that when the number of outliers are low (< 30) A^*-NAPA-DIBP converges to a solution relatively quickly. However, the computational time of A^*-NAPA-DIBP increases exponentially with the number of outliers. On the other hand, the computational time of the proposed algorithms increase linearly with the number of outliers[4]. This is clearly predictable as our algorithms take one step across each level and the deeper down is the MaxCon solution, proportionally longer is the "search". A^*-NAPA-DIBP has a much more sophisticated search that allows backtracking of routes explored and this causes the exponential behaviour when that is heavily exercised. Figure 6a shows the difference between the number of inliers returned by A^*-NAPA-DIBP and other methods. On average the proposed method MBF-MaxCon returns a solution with usually close to the same number of inliers as the A* method. The figure also shows the 0.05th and 0.95th percentile distances from A^* solution over 100 random runs. This shows that in few cases the solution returned by MBF-MaxCon can be up to 2-4 inliers away from the optimal solution (around 1% error). The main summary is that the proposed methods never "go exponential", unlike A^*, in runtime, but compares often

---

[4]Our main extra computational cost compared to other l_∞ methods, sampling the Boolean cube and counting "flips of feasibility", is also trivially parallelizable.
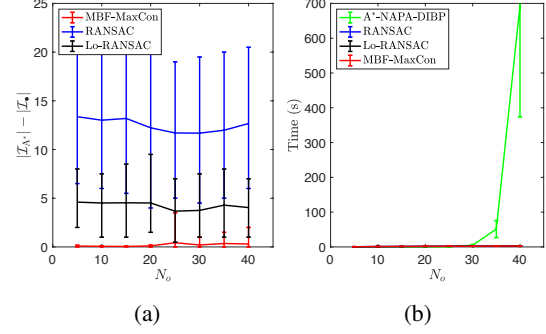

(a)   (b)

Figure 6: Results for 8 dimensional robust linear regression with synthetic data (a) Number of inliers found compared with the global optimal (obtained using A^*) and (b) Variation of computational time with number of outliers. The experiments were repeated 100 times and the error-bars indicate the 0.05-th and 0.95-th percentile.

favourably in terms of accuracy.

4.3. Linearized fundamental matrix estimation

In this section we examine the performances of the proposed methods for linearized fundamental matrix estimation. Provided that the point matches between two views are given as $[\mathbf{p}_1, \mathbf{p}_2]$ where $\mathbf{p}_j = (x_j, y_j, 1)^\top$ is a coordinate of a point in view j, each rigid motion in the scene can be modeled using the fundamental matrix $F \in \mathcal{R}^{3 \times 3}$ as [29]: $\mathbf{p}_1^\top F \mathbf{p}_2 = 0$. In our experiments we use the linearized version presented in [5] together with the algebraic error ([13] chapter 11). For each image pair, the input is a set of SIFT [21] feature matches generated using VLFeat [30].

Single dominant motion: Following [2], we used the first five crossroads image pairs from the sequence "00" of the KITTI Odometry dataset [12] in our experiments. The inlier threshold ϵ is set to 0.03 for all image pairs. The Number of inliers returned by each method (n_i) and the computation times are shown in the first part of Table 1. The results reported for the probabilistic methods are the mean (max, min) over 100 random runs. The results show that the proposed methods on average have produced solutions that are close to the optimum solution returned by A^*-NAPA-DIBP [2].

The distribution of errors by each algorithm over 100 repeated runs for all the frames in sequence "00" of the KITTI Odometry data set is shown in Figure 7. The main message is that we operate in a time cost regime a little better than A^*-NAPA-DIBP and around the same as we allowed for Lo-RANSAC but we generally get much closer to A^*-NAPA-DIBP performance - including often finding the optimal, which RANSAC or Lo-RANSAC rarely do in this experiment.

Table 1: Linearized fundamental matrix estimation result. "Same Comp." refers to running RANSAC with the same time budget as MBF-MaxCon and, "sp=0.99" refers to running RANSAC with the success probability 0.99.

|		A*-NAPA-DIBP	MBF-MaxCon	RANSAC Same Comp.	Lo-RANSAC Same Comp.	RANSAC sp=0.99	Lo-RANSAC sp=0.99
104-108	n_i	289	288.52 (289, 285)	282.03 (286, 277)	284.54 (287, 283)	271.18 (282, 254)	281.41 (284, 276)
	Time (s)	10.96	1.78	1.78	1.78	0.004	0.04
198-201	n_i	296	293.10 (296, 291)	291.88 (294, 290)	292.87 (294, 291)	287.00 (293, 272)	290.35 (293, 287)
	Time (s)	4.04	2.05	2.05	2.05	0.004	0.04
417-420	n_i	366	364.44 (366, 359)	363.54 (365, 361)	364.02 (365, 363)	357.38 (364, 343)	362.33 (364, 359)
	Time (s)	7.93	2.59	2.59	2.59	0.004	0.06
579-582	n_i	523	520.68 (523, 514)	518.04 (521, 511)	520.89 (522, 520)	502.31 (519, 463)	517.30 (512, 497)
	Time (s)	4.28	3.22	3.22	3.22	0.004	0.11
738-742	n_i	462	460.97 (462, 457)	455.02 (459, 451)	457.24 (460, 455)	438.55 (455, 409)	451.35 (459, 435)
	Time (s)	2.97	2.72	2.72	2.72	0.005	0.1
breadcube	n_i	$\sim$	64.36 (68, 58)	61.77 (65, 59)	63.66 (66, 61)	59.04 (65, 55)	62.17 (66, 59)
	Time (s)	>3600	19.07	19.07	19.07	0.993	1.07
breadtoy	n_i	$\sim$	107.48 (115, 102)	105.24 (111, 102)	107.23 (111, 104)	100.31 (107, 93)	105.21 (109, 101)
	Time (s)	>3600	38.37	38.37	38.37	0.526	0.68
cubetoy	n_i	$\sim$	58.51 (61, 52)	54.54 (58, 52)	56.14 (58, 55)	52.20 (56, 48)	55.24 (57, 52)
	Time (s)	>3600	15.38	15.38	15.38	0.45	0.65
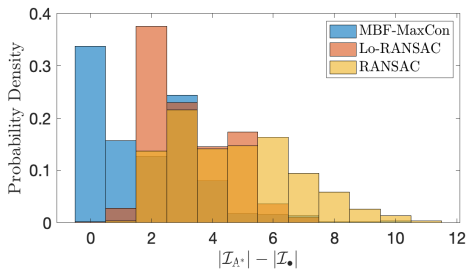
Figure 7: The distribution of errors by each algorithm over 100 repeated runs for all the frames in sequence "00" of the KITTI Odometry data set.

Multiple motions: The above data set contains a single dominant motion and the number of outliers are limited (around 13-22). To study the behaviour of the proposed algorithm in the presence of multiple structures, we conducted experiments on three sequences from the Adelai-deRMF data-set [31]. The inlier threshold ϵ was set to 0.015 for all image pairs. In these sequences, there are two rigidly moving objects. The results in the second part of Table 1 show that the method A*-NAPA-DIBP did not find a solution after 3600s (1 hour), where as the proposed methods found solutions in around 15-40 seconds. Once again, when compared to Lo-RANSAC, we generally obtain higher consensus sets (and thus implicitly closer to what A^* is capable of, but on these data sets, would require astronomically more computation.

5. Conclusion

We have applied a new perspective to the long standing problem of MaxCon. This perspective recognises that the underlying mathematical object is a Monotone Boolean Infeasibility function, defined over the Boolean Cube. Such a perspective immediately identifies a rich mathematical theory that can be applied. Very probably, we have only scratched that surface here. But we have been able to take at least one element of the theory (i.e. Influence) and link that concept to the concept of outlier (in MaxCon) and shown that already, without borrowing further from the rich theory, that we can derive algorithms that are already at least competitive, in some aspects. Specifically:

1. The approach sometimes achieves the true MaxCon, whereas RANSAC rarely achieves the MaxCon solution. Indeed, it is well known that this is a feature of RANSAC based methods in general.

2. The approach can (mostly) achieve close to A^* (provably optimal) in a similar time budget or faster.

3. Unlike A^* the approach will never go exponential in runtime or memory requirements, and - as above - unlike RANSAC variants, it can more reliably obtain the optimal or close to optimal result - given a similar time budget.

The proposed algorithm can be used on model fitting problems where the residuals, $r_{\mathbf{P}_1}(\theta)$, are quasi-convex and the run-time of the algorithm would depend on the existence of the an efficient oracle to evaluate the feasibility/infeasibility of a subset of points.

References

[1] Nader H Bshouty and Christino Tamon. On the fourier spectrum of monotone functions. *Journal of the ACM (JACM)*, 43(4):747–770, 1996. 2

[2] Z. Cai, T. Chin, and V. Koltun. Consensus maximization tree search revisited. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1637–1645, 2019. 3, 5, 6, 7

[3] Zhipeng Cai, Tat-Jun Chin, Huu Le, and David Suter. Deterministic consensus maximization with biconvex programming. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 685–700, 2018. 3

[4] Dylan Campbell, Lars Petersson, Laurent Kneip, and Hongdong Li. Globally-optimal inlier set maximisation for simultaneous camera pose and feature correspondence. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–10, 2017. 3

[5] T. Chin, P. Purkait, A. Eriksson, and D. Suter. Efficient globally optimal consensus maximisation with tree search. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2413–2421, 2015. 3, 7

[6] Tat-Jun Chin, Zhipeng Cai, and Frank Neumann. Robust fitting in computer vision: Easy or hard? In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 701–716, 2018. 3

[7] Tat-Jun Chin and David Suter. The maximum consensus problem: recent algorithmic advances. *Synthesis Lectures on Computer Vision*, 7(2):1–194, 2017. 3

[8] Tat-Jun Chin, David Suter, Shin-Fang Ch'ng, and James Quach. Quantum robust fitting. In Hiroshi Ishikawa, Cheng-Lin Liu, Tomas Pajdla, and Jianbo Shi, editors, *Computer Vision – ACCV 2020*, pages 485–499, Cham, 2021. Springer International Publishing. 3

[9] Irit Dinur and Samuel Safra. On the hardness of approximating minimum vertex cover. *Annals of mathematics*, pages 439–485, 2005. 2, 6

[10] David Eppstein. Quasiconvex programming. *Combinatorial and Computational Geometry*, 52(287-331):3, 2005. 3

[11] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. 1, 3, 7

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 7

[13] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 7

[14] Q. Ke and T. Kanade. Quasiconvex optimization for robust geometric reconstruction. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 986–993 Vol. 2, 2005. 3

[15] A D Korshunov. Monotone boolean functions. *Russian Mathematical Surveys*, 58(5):929–1001, oct 2003. 1

[16] E.G. Kul'yanov. An algorithm for finding the maximal upper zero of an arbitrary monotonic function of the algebra of logic. *USSR Computational Mathematics and Mathematical Physics*, 15(4):267 – 269, 1975. 1

[17] Huu Le, Tat-Jun Chin, and David Suter. An exact penalty method for locally convergent maximum consensus. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1888–1896, 2017. 3

[18] K Lebeda, J Matas, and O Chum. Fixing the locally optimized ransac. In *British machine vision conference, 2012*, 2012. 3, 5, 7

[19] Hongdong Li. Consensus set maximization with guaranteed global optimality for robust geometry estimation. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1074–1080. IEEE. 3

[20] Hongdong Li. A practical algorithm for l triangulation with outliers. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 3

[21] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999. 7

[22] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. 2

[23] Carl Olsson, Olof Enqvist, and Fredrik Kahl. A polynomial-time bound for matching and registration with outliers. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 3

[24] Carl Olsson, Anders Eriksson, and Richard Hartley. Outlier removal using duality. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1450–1457. IEEE, 2010. 3

[25] Thomas Probst, Danda Pani Paudel, Ajad Chhatkuli, and Luc Van Gool. Convex relaxations for consensus and non-minimal problems in 3d vision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10233–10242, 2019. 3

[26] Rahul Raguram, Ondrej Chum, Marc Pollefeys, Jiri Matas, and Jan-Michael Frahm. Usac: a universal framework for random sample consensus. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):2022–2038, 2012. 3

[27] Srikumar Ramalingam, Chris Russell, Lubor Ladicky, and Philip H.S. Torr. Efficient minimization of higher order submodular functions using monotonic boolean functions. *Discrete Applied Mathematics*, 220:1 – 19, 2017. 2

[28] Kristy Sim and Richard Hartley. Removing outliers using the l_∞ norm. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 485–494. IEEE, 2006. 3, 5

[29] P. H. S. Torr and D. W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, 24(3):271–300, 1997. 7

[30] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008. 7

[31] Hoi Sim Wong, Tat-Jun Chin, Jin Yu, and David Suter. Dynamic and hierarchical multi-structure geometric model fit-

ting. In *2011 International Conference on Computer Vision*, pages 1044–1051. IEEE, 2011. 8

[32] Heng Yang and Luca Carlone. A quaternion-based certifiably optimal solution to the wahba problem with outliers. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1665–1674, 2019. 3

[33] Y. Zheng, S. Sugimoto, and M. Okutomi. Deterministically maximizing feasible subsystem for robust model fitting with unit norm constraint. In *CVPR 2011*, pages 1825–1832, 2011. 3