Deterministically Maximizing Feasible Subsystem for Robust Model Fitting with Unit Norm Constraint

Yinqiang Zheng Shigeki Sugimoto Masatoshi Okutomi
Department of Mechanical and Control Engineering, Tokyo Institute of Technology
http://www.ok.ctrl.titech.ac.jp/

Abstract

Many computer vision problems can be accounted for or properly approximated by linearity, and the robust model fitting (parameter estimation) problem in presence of outliers is actually to find the Maximum Feasible Subsystem (MaxFS) of a set of infeasible linear constraints. We propose a deterministic branch and bound method to solve the MaxFS problem with guaranteed global optimality. It can be used in a wide class of computer vision problems, in which the model variables are subject to the unit norm constraint. In contrast to the convex and concave relaxations in existing works, we introduce a piecewise linear relaxation to build very tight under- and over-estimators for square terms by partitioning variable bounds into smaller segments. Based on this novel relaxation technique, our branch and bound method can converge in a few iterations. For homogeneous linear systems, which correspond to some quasi-convex problems based on L_∞-L_∞-norm, our method is non-iterative and certainly reaches the globally optimal solution at the root node by partitioning each variable range into two segments with equal length. Throughout this work, we rely on the so-called Big-M method, and successfully avoid potential numerical problems by exploiting proper parametrization and problem structure. Experimental results demonstrate the stability and efficiency of our proposed method.

1. Introduction

To identify the maximum cardinality feasible subsystem of an infeasible linear system $\mathbf{Ax} \leq \mathbf{b}$ with a real matrix $\mathbf{A} \in R^{m\times n}$ and a real column vector $\mathbf{b} \in R^m$ is known as the Maximum Feasible Subsystem (MaxFS) problem [6]. It has widespread application in mathematical programming, machine learning and signal processing. However, it is NP-hard to solve globally. There is an exact mixed integer linear programming (MILP) formulation for the MaxFS problem by introducing a binary variable $y_i, i = 1, 2, \cdots, m$, for each constraint. Specifically,

$$\begin{aligned} \max_{\mathbf{x},\mathbf{y}} \quad & \sum_{i=1}^{m} y_i, \\ s.t., \quad & \sum_{j=1}^{n} a_{ij}x_j \leq b_i + (1-y_i)M_i, \\ & y_i \in \{0,1\}, i = 1, 2, \cdots, m, \end{aligned} \tag{1}$$

where a_{ij} is the i-th row and j-th column element of $\mathbf{A}$, b_i is the i-th element in $\mathbf{b}$, and M_i is a sufficiently large constant that deactivates the i-th constraint when $y_i = 0$. This is called as the Big-M method [6], which is usually not appealing due to potential numerical problems. To avoid overly large M_i, one possible solution [3] is to solve the following linear programming (LP) in case of tightly bounded model parameter $\mathbf{x}$, i.e., $x_j^l \leq x_j \leq x_j^u, j = 1, 2, \cdots, n$,

$$\begin{aligned} M_i = \max_{\mathbf{x}} \quad & \{\mathbf{a}_i\mathbf{x} - b_i\}, \\ s.t., \quad & x_j^l \leq x_j \leq x_j^u, j = 1, 2, \cdots, n, \end{aligned} \tag{2}$$

where $\mathbf{a}_i$ is the i-th row of $\mathbf{A}$, x_j^l and x_j^u denote the lower and upper bounds of x_j, respectively. Assuming that numerical instability can be avoided, this MILP formulation is preferable for medium-scale problems, since existing MILP solvers can solve it to the globally optimal solution efficiently in presence of dozens or hundreds of binary variables. See [16] for the latest benchmark tests.

Another well-known formulation of MaxFS is the linear programme with equilibrium constraints (LPEC) [6], which is in essence a bilinear programming problem. The LPEC formulation reads

$$\begin{aligned} \max_{\mathbf{x},\mathbf{y}} \quad & \sum_{i=1}^{m} y_i, \\ s.t., \quad & y_i(\mathbf{a}_i\mathbf{x}) \leq y_i b_i, \\ & 0 \leq y_i \leq 1, i = 1, 2, \cdots, m. \end{aligned} \tag{3}$$

Although there is no binary variable involved, introducing $m \times n$ bilinear terms never makes the problem much easier.

Actually, the MaxFS problem is not isolated. The complementary problem of the MaxFS is to find the minimum number of linear constraints to remove such that the remaining subsystem becomes feasible, which is named as the Minimum Unsatisfied Linear Relation (MinULR) problem [2]. A simple mathematical formulation is to introduce m nonnegative variables $s_i, i = 1, 2, \cdots, m$, and minimize the L_0-norm of the vector **s**, i.e., the number of nonzero elements in **s**,

$$\begin{aligned} \min_{\mathbf{x},\mathbf{s}} \quad & \|\mathbf{s}\|_0 \\ s.t., \quad & a_i\mathbf{x} \le b_i + s_i, \\ & s_i \ge 0, i = 1, 2, \cdots, m. \end{aligned} \tag{4}$$

The MinULR problem is of the same importance, since it is implicitly related to sparsity in many topics like sparse coding and compressive sensing. In spite of the difference in a literal way, we actually have no need to make too much difference between MaxFS and MinULR, since they are usually interchangeable. In the remaining of this work, we shall focus primarily on the MaxFS problem. Readers can easily adapt our method to solve the MinULR.

1.1. Related Works

In the past few decades, numerous methods haven been proposed to solve the MaxFS problem and its variants in other research fields. The majority of existing algorithms can only solve them heuristically, without a guaranty of global optimality, like [15, 3] and the references therein. Only a few works aimed at solving them deterministically, typically the branch and cut method proposed by Pfetsch [20]. However, these methods can not be easily used for computer vision problems, since the constraints are generally regarded as being independent. In computer vision problems, however, one image measurement usually determines two or even four linear constraints, and we prefer to maximize the number of measurements while ensuring that the resulting linear system is feasible.

Now we take a closer view at the robust fitting problem in computer vision. Although the terminology of MaxFS seems relatively new in the computer vision community, the basic conception of MaxFS is identical to many longstanding robust estimation techniques, such as the well-known RANSAC [9] and its numerous recent extensions. Due to their random nature, these methods would usually not find the maximum feasible set exactly.

Recently, L_∞-norm aroused a lot of attention due to the quasi-convex structures in a class of multiview geometry problems [11, 12]. In [12], Ke and Kanade pointed out that outlier removal is actually to find the minimum number of infeasibility, and alternatively minimized the sum of infeasibility, which is actually the L_1-norm approximation to the L_0-norm. Dalalyan and Keriven [8] theoretically defended the effectiveness of L_1-norm for this class of problems. Based on the quasi-convexity, Kim and Hartley [21] proposed to remove the measurements with the largest reprojection error recursively, until feasibility is reached. Olsson *et al*. [18] accelerated the process significantly by using duality. All these methods are different from random sampling, and they can find a feasible subset, but not necessarily the largest one.

One of the recent trends is to solve the robust fitting problem with guaranteed optimality. For problems with pseudo-convexity, Olsson *et al*. [17] presented a hit-or-miss strategy with a polynomial-time bound. When the problem can be described by a linear system, the robust fitting problem becomes the MaxFS. Using the LPEC formulation in eq.(3), Li [13] proposed a branch and bound method and applied it to a wide class of problems based on Direct Linear Transformation (DLT). However, too many bilinear terms are involved and a quite rough initial branching region used, both of which make the convergence quite slow even for slightly larger problems. In [23], Yu *et al*. directly solved the MILP formulation in eq.(1) by using a MILP solver, and reported satisfactory performance for calibrated photometric stereo. However, this idea can not be directly extended for other problems, like those in [13], since we usually do not have tight variable bounds at hand. Actually, Li [13] chose quite rough variable bounds ($\pm 100 \sim \pm 1000$) to make sure that the globally optimal solution is included. Using such wild bounds in the MILP formulation would cause numerical problems quite frequently.

1.2. Overview of Our Proposed Method

In this work, we are aiming at the global optimization for the MaxFS problem, in which the model variables are subject to the unit norm constraint. The unit norm constraint provides natural variable bounds with moderate width, and improves numerical stability in the Big-M method of the MILP formulation. Note that this desirable property comes at the costs of introducing additional nonlinearity. We use a branch and bound method to solve this challenging optimization problem. The novelty lies in the piecewise linear under- and over-estimators for square terms. To tighten the relaxation, we partition the variable ranges into smaller segments and transform the resulting disjunctive programming into a MILP problem using the Big-M method again. Based on the tight relaxation, our branch and bound method can usually converge in a few iterations by choosing proper number of segments. More interestingly, for homogeneous linear systems, which correspond to some L_∞-L_∞-norm based quasi-convex problems, our branch and bound method certainly converges to the globally optimal solution at the root node, by partitioning each variable range into two segments only. This actually means that our branch and bound method is non-iterative for this special class of problems.

1.2.1 To Which Problems Our Method Applicable?

One might think the application of our method is limited due to the unit norm constraint. However, this unit norm constraint actually poses little restriction.

DLT Based Geometry Fitting Problems. The model variables in all the problems in [13] are defined up to a scale. It is natural to eliminate this scale ambiguity by introducing the unit norm constraint. Therefore, our method can solve all those problems, including line fitting, similarity estimation, fundamental matrix estimation and so on.

Quasi-Convex Problems Using L_∞-L_∞-norm. Other than the DLT, we can also use the geometrically meaningful L_∞-L_∞-norm, under which the robust estimation of some quasi-convex problems, such as homography estimation and camera resection, is to find the MaxFS of a homogeneous linear system $\mathbf{Ax} \leq \mathbf{0}$ with the unit norm constraint.

The triangulation problem is also quasi-convex. If the translations of all the cameras are fully calibrated, the triangulated point would usually not satisfy the unit norm constraint. However, after introducing an additional variable, the unit norm can be satisfied. Note that it generally leads to a nonhomogeneous system. Refer to the following case for the basic idea.

PCA Based Model Fitting. Linear subspace methods are widespread in computer vision, and the easiest underlying mathematical tool is principle component analysis (PCA). One example is the well-known Active Shape Model (ASM) [7]. Assuming that the ASM has been trained through PCA, we want to fit this model to an observation, such that $\phi = \Lambda\theta + \phi_0$, where Λ is the known transformation matrix and ϕ_0 is the known translation vector. ϕ is a noisy observation, and θ is the unknown low dimensional latent variable. Generally, θ does not satisfy the unit norm constraint. However, we can multiply an arbitrary positive variable λ on both sides as $\lambda(\phi - \phi_0) = \lambda\Lambda\theta$. Let $\psi = \lambda\theta$, and now it is reasonable to enforce that the augmented vector (λ, ψ) satisfy the unit norm constraint. Given a noise tolerance δ, the ASM fitting problem can be easily transformed into a MaxFS problem.

Our method can also be easily adapted to solve the photometric stereo problem in [23]. Certainly, there exist some other applications to be discovered.

1.2.2 Our Contribution

Our contribution can be summarized as follows:

1. We introduce novel piecewise linear under- and over-estimator for square terms. In terms of relaxation tightness, this piecewise relaxation mechanism is superior over the popular convex and concave relaxations in existing literature, such as [13, 5, 19].

2. We point out that some L_∞-L_∞-norm based quasi-convex problems correspond to homogeneous linear systems, for which our method becomes non-iterative and surely reaches the optimal solution at the root node by uniformly partitioning each variable bound into two segments.

3. Throughout this work, we rely heavily on the ”notorious” Big-M method [6]. We show that it is actually a very convenient tool in modeling MILP problem and the numerical instability can be avoided by taking advantage of proper parametrization and problem structure.

2. Global Optimization to MaxFS with Unit Norm Constraint

2.1. Problem Formulation

We are in favor of the Big-M based MILP formulation in eq.(1). Unlike [23], in which this formulation is directly used, we introduce the unit norm constraint $||\mathbf{x}||_2 = 1$, i.e.,

$$\sum_{j=1}^{n} x_j^2 = 1. \tag{5}$$

From eq.(5), all the variables are naturally bounded by

$$-1 \leq x_j \leq 1, j = 1, 2, \cdots, n. \tag{6}$$

Undoubtedly, the globally optimal solution lies in these bounds. Using these moderate bounds, $M_i, 1 \leq i \leq m$ can be easily determined by solving m linear programming problems in eq.(2). Due to these mild bounds, in all our experiments, we found that M is quite small, and numerical problems never appeared. Note that the success in avoiding numerical instability comes with costs, since we have to handle the nonlinearity in the unit norm constraint (eq.(5)).

After introducing n additional variables w_j such that $w_j = x_j^2$, the MaxFS problem becomes

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y}} \quad & -\sum_{i=1}^{m} y_i \\ s.t., \quad & \mathbf{a}_i\mathbf{x} \leq b_i + (1 - y_i)M_i, \\ & y_i \in \{0, 1\}, i = 1, 2, \cdots, m, \\ & \sum_{j=1}^{n} w_j = 1, w_j = x_j^2, \\ & -1 \leq x_j \leq 1, j = 1, 2, \cdots, n. \end{aligned} \tag{7}$$

It is a challenging mixed integer nonlinear programming problem, for which we develop a branch and bound method. At each node, we relax the square terms to obtain a MILP relaxation problem. Note that we do not directly relax the binary variables y_i, but leave them to a MILP solver. Currently, some decent softwares [16] can solve MILP problems efficiently, especially for medium-scale problems with dozens or hundreds of binary variables.

2.2. Relaxation for Square Terms

Given a region $\Phi = \{\mathbf{x} \in R^n | x_j^l \leq x_j \leq x_j^u, j = 1, 2, \cdots, n\}$, to build tight relaxation is the key to accelerate the convergence of a branch and bound method. In existing computer vision literature ([13, 5, 19]), the mainstream idea is to build convex and concave relaxations for nonconvex terms. In the sense of convexity and concavity, the tightest possible relaxations are the convex and concave envelopes, the bilinear envelopes in [13, 5, 19] as typical representatives. However, if going beyond convexity and concavity, we are able to build much more tighter relaxations.

Now we show how to build the under-estimator $under(w)$ and over-estimator $over(w)$ of a square term $w = x^2, x^l \leq x \leq x^u$ such that $under(w) \leq w \leq over(w)$. Note that we omit the subscript j of x_j in this whole section for simplification.

2.2.1 Traditional Convex and Concave Relaxation

Let's first consider the concave relaxation (or over-estimator). Considering that the function $f(x) = x^2, x^l \leq x \leq x^u$ is convex, the concave over-estimator $over(w)$, as shown in Fig.1(a), is the chord connecting the two endpoints $(x^l, (x^l)^2)$ and $(x^u, (x^u)^2)$ [19]. Specifically,

$$w \leq (x^l + x^u)x - x^l x^u. \quad (8)$$

The linear convex under-estimator $under(w)$ is composed of the two tangents at the two endpoints (shown in Fig.1(c))

$$w \geq 2x^l x - (x^l)^2, w \geq 2x^u x - (x^u)^2. \quad (9)$$

We are interested in the relaxation quality. The gap between the original function and the concave relaxation $G(x)$ reads

$$G(x) = -x^2 + (x^l + x^u)x - x^l x^u, \quad (10)$$

and it is easy to recognize that the maximum gap is $(x^u - x^l)^2/4$. The same result holds for the convex relaxation. The observation that the maximum gap shrinks with smaller bound width leads us to a novel piecewise relaxation method, in which each variable range is partitioned into smaller segments.

2.2.2 Piecewise Linear Relaxation

We partition the variable range $[x^l, x^u]$ into K smaller segments. For simplicity, we use uniform partitioning, so that all these segments are of equal length. Let $\{[x(k), x(k+1)], k = 1, 2, \cdots, K\}$ denote these segments, where

$$x(k) = x^l + \frac{x^u - x^l}{K}(k-1), k = 1, 2, \cdots, K. \quad (11)$$

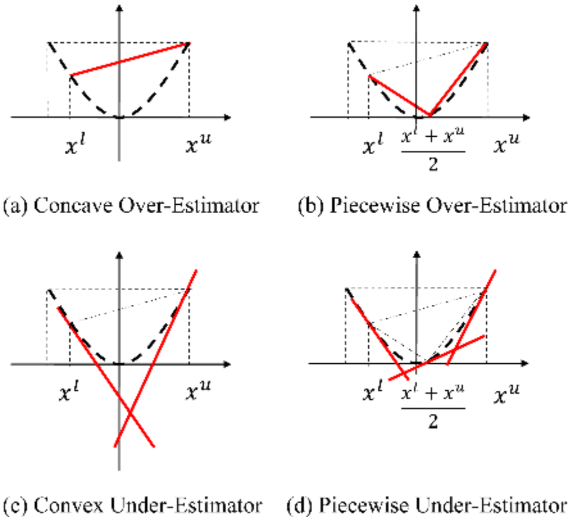

x^l
x^u
x^l
$\frac{x^l + x^u}{2}$
x^u
(a) Concave Over-Estimator
(b) Piecewise Over-Estimator
x^l
x^u
x^l
$\frac{x^l + x^u}{2}$
x^u
(c) Convex Under-Estimator
(d) Piecewise Under-Estimator

Figure 1. Concave and convex relaxations v.s. piecewise linear relaxations for a square term. (a) and (c) are the concave and convex linear over- and under-estimator, respectively. (b) and (d) are the piecewise linear over- and under-estimator using K=2 segments, respectively.

Clearly, x must lie in one of the K segments or at the boundary of two neighboring segments. Now, using the concave relaxation in eq.(8) for each segment, we obtain the piecewise linear over-estimator $over(w)$ (see Fig.1(b) for an example with two segments), which can be described as the following disjunctive programming [4]

$$OR \left| \begin{array}{c} W(k) \\ w \leq [x(k) + x(k+1)]x - x(k)x(k+1) \\ x(k) \leq x \leq x(k+1) \end{array} \right|. \quad (12)$$

$W(k)$ is the boolean variable (*true* or *false*) indicating the status of the k-th disjunction. The OR logic dictates that only one disjunction holds.

In order to transform the disjunctive logic into a concrete mathematical programming model, we need to introduce binary variables and obtain a mixed integer program. Now we introduce K binary variables such that

$$\lambda(k) = \begin{cases} 1, & if \;\; x(k) \leq x \leq x(k+1) \\ 0, & otherwise \end{cases} \quad (13)$$

and

$$\sum_{k=1}^{K} \lambda(k) = 1. \quad (14)$$

We use the Big-M method again to obtain a MILP model as follows

$$w \leq [x(k)+x(k+1)]x - x(k)x(k+1) + [1-\lambda(k)]M, \forall k, \quad (15)$$

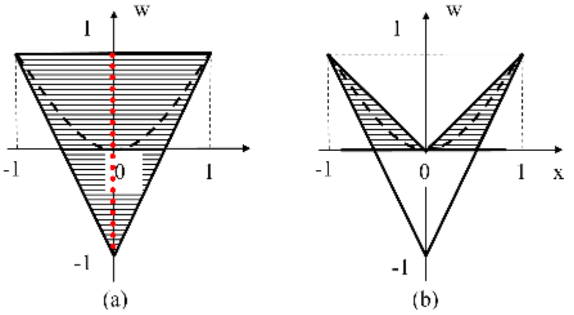


Figure 2. Using piecewise linear relaxations to avoid degeneracy for a homogeneous linear system. (a) The degeneracy can not be avoided, if using traditional convex and concave relaxation. (b) Using two segments for each variable range, the piecewise linear relaxation can prevent the degenerate solution. The shaded area indicates the feasible range after relaxing.

where M is a sufficiently large number so as to deactivate the constraint when $\lambda(k) = 0$. However, to avoid numerical instability, we should keep M as small as possible.

Through simple differential and limit operations, we note that among all the over-estimators over the initial variable range $[-1, 1]$, the steepest one has a slope close to 2 (or -2), and reaches its minimal value, i.e., -3, when $x = -1$ (or $x = 1$). The maximum of w is 1, therefore, it is sufficient to set M to 4. Note that such a M always holds effective as the branch and bound proceeds.

The piecewise linear under-estimator $under(w)$ is much easier, without involving the disjunctive programming (see Fig.1(d) for an example with two segments). Applying the linear convex relaxation in eq.(9) to each segment, the $under(w)$ is composed of the following $K + 1$ linear inequalities

$$\begin{aligned} &w \geq 2x(k)x - x(k)^2, k = 1, 2, \cdots, K, \\ &w \geq 2x(K+1)x - x(K+1)^2. \end{aligned} \tag{16}$$

Introducing $n \times K$ additional binaries and establishing the piecewise relaxations for all n square terms in the same way, we obtain the MILP relaxation problem. It is easy to recognize that the relaxation becomes tighter when using more segments, thus expectedly less iterations needed for convergence. On the other hand, more binary variables should be introduced and the relaxation problem at each node becomes harder to solve. To reduce the total time of the whole branch and bound iterations, we should choose properly the number of segments and trade off between the relaxation quality and the size of the relaxation problem.

2.3. Branch and Bound Algorithm

Our branch and bound algorithm is similar to those in many other computer vision works [13, 5, 19]. We should note the difference between *branch* and *partition* mentioned above. Actually, *partition* is a sub-step in the *bound* process, and independent of the *branch* process.

Initial Branching Region. The region defined by the natural variable bounds in eq.(6) is the initial region.

Lower Bound and Upper Bound. When solving the MILP relaxation problem, we obtain the optimal objective and regard it as the lower bound (LB) for the current region. We also obtain the optimizer of **x**, which usually does not satisfy the unit norm constraint until the globally optimal solution is reached. However, we can normalize this optimizer to unit norm and then count the number of feasible constraints. We use this number, actually multiplied by -1, as the upper bound (UB) for the current region. Unfortunately, we experimentally found that such a simple strategy causes slow convergence. To remedy this problem, we propose to refine the current optimizer through local optimization. We solve the nonlinear optimization problem composed of the LPEC formulation in eq.(3) and the unit norm constraint in eq.(5), starting from the optimizer of the MILP relaxation problem. Through this local optimization, we get a refined **x** usually with unit norm. We normalize it to unit norm just for safety, and count the number of feasible constraints. One might worry about the correctness of using the LPEC formulation, while our objective is to solve the MILP formulation. We argue that it would not cause any problem, since we always normalize **x** to unit norm and then count the feasible constraints as the upper bound.

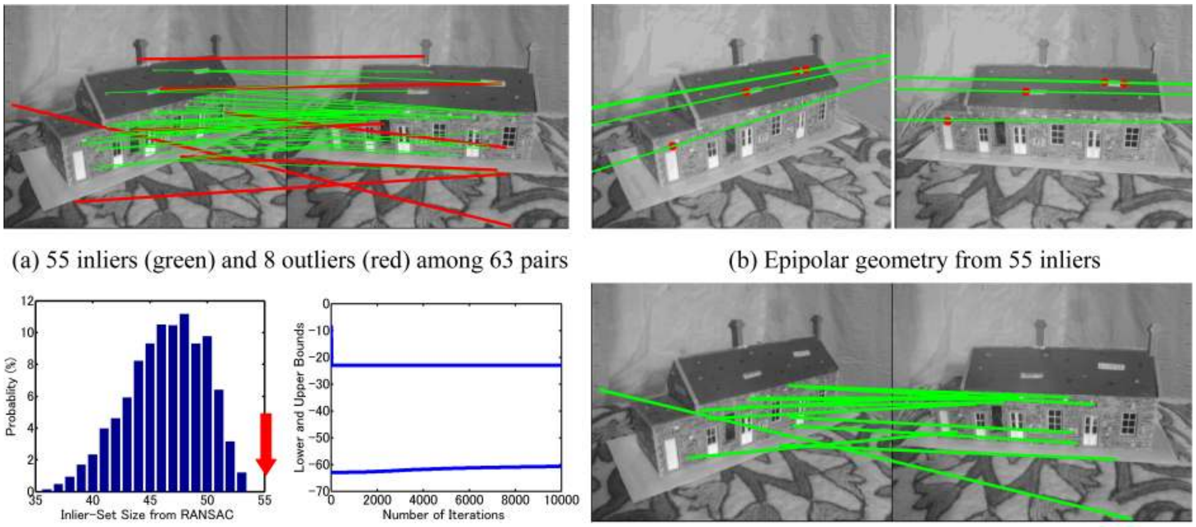
Branching Strategy. In the list of sub-regions, we choose to branch on the one with the lowest lower bound at the midpoint of its longest side. This simple heuristic strategy works well in all our experiments.

Convergence. We terminate the branch and bound iteration when the absolute gap between the current best upper bound and lower bound converges to zero. This zero-gap tolerance guarantees that our method reports the globally optimal solution, in contrast to a ε-suboptimal solution in the majority of existing works, like [5, 19].

Actually, many works using the branch and bound method have appeared in the computer vision community, like [13, 5, 19]. Therefore, the detailed procedures are omitted here.

2.4. A Special Case - Homogeneous Linear Systems

When $\mathbf{b} = \mathbf{0}$, the linear system degenerates into a homogeneous linear system $\mathbf{Ax} \leq \mathbf{0}$. An important property is that the homogeneous system always holds when multiplying the left-hand side by any positive constant, which means that, for this special case, our branch and bound method surely reaches the zero-gap and thus converges at the root node. However, we should take care of the degenerate case that the whole homogeneous linear system is trivially feasible when $\mathbf{x} = \mathbf{0}$. Interestingly, by using our piecewise linear relaxations, this trivial solution can be avoided. It is desir-



(a) 55 inliers (green) and 8 outliers (red) among 63 pairs

(b) Epipolar geometry from 55 inliers

(c) RANSAC

(d) Li's optimal method

(e) Yu *et al*.'s optimal method with 14 inliers (green)

Figure 3. Results for robust fundamental matrix estimation. (a) Among 63 tentative point pairs, our optimal method found 55 inliers and 8 outliers. (b) The epipolar lines estimated from the N8P method using 55 inliers (only 4 lines shown). (c) The statistics of RANSAC over 5000 runs. The red arrow indicates the position our result lies. (d) After 10,000 iterations, there still is a wide gap in Li's method [13]. (e) Yu *et al*.'s method [23] had numerical problems, and found 14 "inliers" only.

Relax.	Root Node	Whole Algorithm		
	UB/LB	UB/LB	Iterations	Time(sec.)
Con.	-1/-63	-55/-55	535	4729.3
K=2	-49/-56	-55/-55	21	84.0
K=4	-55/-55	-55/-55	0	12.6
K=6	-55/-55	-55/-55	0	31.8

Table 1. Convex/concave relaxations v.s. piecewise linear relaxations with varying segments.

able to use as few segments as possible, while making sure that the degeneracy can be avoided.

What's the minimum number of segments? The answer is two. In the MILP relaxation problem, the only constraint to prevent $\mathbf{x} = \mathbf{0}$ is the relaxed unit norm constraint $\sum_{i=1}^{n} w_i = 1$. Using the traditional convex and concave relaxation, the degenerate solution can not be avoided. As shown in Fig.2(a), when $x_j = 0, j = 1, 2, \cdots, n$, w_j can be any number in the range $[-1, 1]$, therefore, the relaxed unit norm constraint can be easily satisfied. In contrast, when using piecewise linear relaxations with two segments for each variable range (see Fig.2(b)), the relaxed unit norm constraint can never be satisfied when $\mathbf{x} = \mathbf{0}$. The reason is that w_j must be zero when $x_j = 0, j = 1, 2, \cdots, n$.

3. Experimental Results

In this section, we show some applications of our method in various practical problems. We introduce one binary variable for each image measurements, and maximize the number of measurements such that the resulting linear system is feasible. All codes are implemented in Matlab, and run on a desktop with Intel Quad 9650 3.0GHz CPU and 3GB RAM. We use the leading LP/MILP solver GUROBI [1] to solve all the LP and MILP problems. The GUROBI software solves MILP problem by using a parallel branch and cut method and reports the globally optimal solution if it exists. We enable the parallel mode and report the actual elapsed time, not the CPU time. As to the local optimization for refining the upper bound, we use the *fmincon* functionality in Matlab. In addition, whenever possible, we properly normalize the coordinates of image measurements to improve the numerical stability further.

3.1. Robust Fundamental Matrix Estimation

As a representative of the class of DLT-based fitting problems, robust fundamental matrix estimation is considered here. We assume a perspective fundamental matrix with 9 variables in the model parameter vector $\mathbf{x}$, in contrast to the affine fundamental matrix in [13]. The rank-2 constraint is not enforced. We use two images from the well-known Oxford *house* sequence. 63 tentative point correspondences are established by matching SIFT feature points with a matching threshold of 0.7 [14]. For this case, we use a noise tolerance of 0.25, i.e., setting T=0.25 in [13].

First of all, we used our method and found the maximal inlier set with 55 point pairs(Fig.3(a)). Fig.3(b) shows the epipolar geometry estimated from the normalized 8-point (N8P) method [10] using 55 inliers. Table 1 shows the

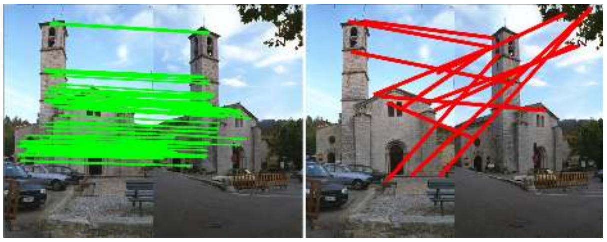


Figure 4. Results for robust homography estimation. Among 98 tentative correspondences, our optimal method took 21 seconds and found 83 inliers (shown in green at the left side) and 15 outliers (shown in red at the right side).

performance of piecewise linear relaxations using various number of segments K, as well as that of the traditional convex and concave relaxation. As expected, using more segments will generate tighter relaxation, and reduce the number of iterations, but not necessarily the running time of the whole branch and bound algorithm. Using K=4 segments (the best choice for this problem), our method reaches the globally optimal solution at the root node in 12.6 seconds, which is much faster than using the traditional convex and concave relaxation.

After that, we tested the performance of RANSAC by repeating 5000 times and counting the reported inliers for each run. Fig.3(c) shows the statistics, from which we see that RANSAC rarely reports the maximum inlier sets, and our result is better than those of RANSAC over 5000 runs.

We also implemented the branch and bound method by Li [13] and the direct solving of the MILP formulation by Yu *et al*. [23] for comparison. Since the variable ranges can not be easily determined for these two methods, we use the conservative range $[-500, 500]$ for each variable. Note that in [13], Li even used $[-1000, 1000]$. For the method in [13], we need to handle 63×8 bilinear terms, and branch in a very huge 8-D space defined by the variable ranges, after setting the last element of the fundamental matrix to 1. Both factors make the convergence quite slow. Fig.3(d) shows the progress of Li's method, from which we see that there is still a wide gap between the upper and the lower bound after 10,000 iterations (about 1200 seconds). Yu *et al*. [23] directly solved the MILP formulation in eq.(1). However, when solving a problem with so wide variable ranges, GUROBI reported numerical problems and returned an inaccurate solution with 14 feasible pairs only (Fig.3(e)), some of which are actually outliers.

3.2. Robust Homography Estimation

This can be regarded as an example for some L_∞-L_∞-norm based quasi-convex problems. The resulting linear system is homogeneous, thus we use two segments for each variable range. Our method certainly reaches the globally optimal solution at the root node. The traditional convex and concave relaxations are not appropriate due to degeneracy described in section 2.4.

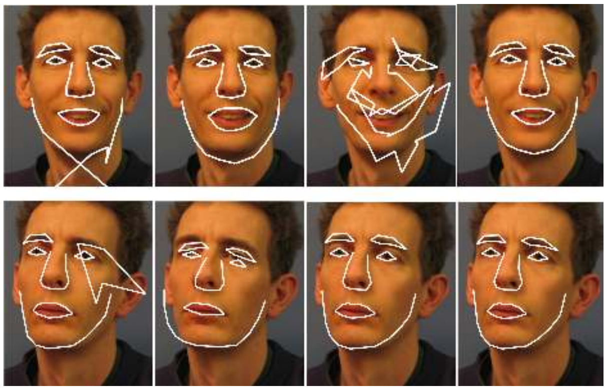


Figure 5. Results for robust ASM fitting. For each row, from left to right are the original noisy observation, (i) direct LLS result without outlier removal, (ii) LLS result using inliers from L_1-norm method and (iii) LLS result using inliers from our optimal method.

We used two views from the Oxford *Valbonne Church* sequence and established 98 tentative correspondences by setting the matching threshold to 0.7. A noise tolerance of 0.2 was used here. Among 98 tentative pairs, our optimal method took 21 seconds and found 83 inliers and 15 outliers as shown in Fig.4.

Then, we compared our result with those from the duality method in [18] and the L_1-norm method in [8]. The duality method retained 75 inliers, while the L_1-norm method 72 inliers. In spite of their efficiency, our method offers the best solution, certainly at the cost of reasonably higher computational burden.

Similar to the previous experiment, we used the conservative range $[-500, 500]$ for each model variable for Li's and Yu *et al*.'s methods. For this problem, Li's method converged to the optimal solution after 4251 iterations and took about 1920 seconds. The GUROBI solver reported numerical problems for Yu *et al*.'s method and returned an inaccurate solution, which is feasible to 26 point pairs only.

3.3. Robust ASM Fitting for Human Face

As an example of robust PCA based model fitting, we consider the ASM fitting problem for human face. We used the dataset provided by Cootes [22]. We used 52 markers on each image and trained an ASM model with 9 shape modes through eigenvalue decomposition. After that, we added some random outliers to simulate noisy observations. We fit the trained ASM model to each noisy observation separately by using the linear least square (LLS) method. We consider three different strategies: (i) using noisy data directly without removing outliers, (ii) using inliers reported by the L_1-norm method and (iii) using inliers from our optimal method. The L_1-norm method is an approximation to the L_0-norm in eq.(4). Fig.5 illustrates the results for two test cases. For the case shown in the first row, our optimal method took 67 seconds and found the maximum set with 48 inliers at the root node by using 6 segments, while the L_1-norm method found 16 inliers only. As expected, the LLS method using noisy data directly has poor performance, while L_1-norm method has the risk of removing too many inliers incorrectly and the final result is meaningless. For the case shown in the second row, our optimal method found 49 inliers as the maximum feasible set, while the L_1-norm method found 26 inliers. For this case, these reported 26 inliers provide sufficient information for a satisfactory fitting. Visually comparing the results shown in Fig.5, the LLS using inliers from our optimal method always has the best performance.

4. Conclusion

We have shown a branch and bound method for the maximum feasible subsystem (MaxFS) problem with application to robust model fitting (parameter estimation). The novelty lies in the unit norm constraint to improve numerical stability and the piecewise linear relaxations for square terms to tighten relaxation. Due to this novel relaxation technique, our branch and bound method can generally converge in a few iterations. Particularly, for homogeneous linear systems, our method certainly converges to the globally optimal solution at the root node by partitioning each variable range into two segments with equal length. Experimental results demonstrated its superiority over two existing optimal methods in terms of numerical stability and (or) computational efficiency.

Currently, our method is reasonably efficient for medium-scale problems with about one hundred image measurements and ten variables in the model. For lower dimensional problems like line fitting and similarity estimation, our method can easily handle hundreds of measurements. One possible method to improve the scalability further is to warm-start the MILP solver by using non-optimal methods, like RANSAC and L_1-norm method.

Acknowledgement. This work was partly supported by Grant-in-Aid for Scientific Research (21240015) from the Japan Society for the Promotion of Science.

References

[1] Gurobi optimization. *http://www.gurobi.com/*, 2010. 1830

[2] E. Amaldi. From finding maximum feasible subsystems of linear systems to feed-forward neural network design. *PhD Thesis,EPFL,Switzerland*, 1994. 1826

[3] E. Amaldi, M. Bruglieri, and G. Casale. A two-phase relaxation-based heuristic for the maximum feasible subsystem problem. *Computers & Operations Research*, 35:1465–1482, 2008. 1825, 1826

[4] E. Balas. Disjunctive programming. *Annals of Discrete Mathematics*, 5:3–51, 1979. 1828

[5] M. Chandraker, S. Agarwal, D. Kriegman, and S. Belongie. Globally optimal algorithms for stratified autocalibration. *IJCV*, 90(2):236–254, 2010. 1827, 1828, 1829

[6] J. Chinneck. *Feasibility and infeasibility in optimization: algorithms and computational methods*. Springer, first edition, 2007. 1825, 1827

[7] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models-their training and application. *CVIU*, 61(1):38–59, 1994. 1827

[8] A. Dalalyan and R. Keriven. L1-penalized robust estimation for a class of inverse problems arising in multiview geometry. *In Proc. NIPS*, 2009. 1826, 1831

[9] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *In Commun. ACM*, 24(6):381–395, 1981. 1826

[10] R. I. Hartley. In defense of the eight-point algorithm. *IEEE TPAMI*, 19(6):580–593, 1997. 1830

[11] F. Kahl and R. Hartley. Multiple view geometry under the l_∞-norm. *IEEE TPAMI*, 30(9):1603–1617, 2008. 1826

[12] Q. Ke and T. Kanade. Quasiconvex optimization for robust geometric reconstruction. *IEEE TPAMI*, 29(10):1834–1847, 2007. 1826

[13] H. Li. Consensus set maximization with guaranteed global optimality for robust geometry estimation. *In Proc. ICCV*, 2009. 1826, 1827, 1828, 1829, 1830, 1831

[14] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1830

[15] O. Mangasarian. Misclassification minimization. *Journal of Global Optimization*, 5(4):309–323, 1994. 1826

[16] H. Mittelmann. Decision tree for optimization software. *http://plato.asu.edu/guide.html*, 2010. 1825, 1827

[17] C. Olsson, O. Enqvist, and F. Kahl. A polynomial-time bound for matching and registration with ouliers. *In Proc.CVPR*, 2008. 1826

[18] C. Olsson, A. Eriksson, and R. Hartley. Outlier removal using duality. *In Proc.CVPR*, 2010. 1826, 1831

[19] C. Olsson, F. Kahl, and M. Oskarsson. Branch-and-bound methods for euclidean registration problems. *IEEE TPAMI*, 31(5):783–794, 2009. 1827, 1828, 1829

[20] M. Pfetsch. Branch-and-cut for the maximum feasible subsystem problem. *SIAM Journal on Optimization*, 19(1):21–38, 2008. 1826

[21] K. Sim and R. Hartley. Removing outliers using the l_∞-norm. *In Proc. CVPR*, 2006. 1826

[22] T.Cootes. Modelling and search software. *http://personalpages.manchester.ac.uk/staff/timothy.f.cootes/*. 1831

[23] C. Yu, Y. Seo, and S. Lee. Photometric stereo from maximum feasible lambertian reflections. *In Proc. ECCV*, 2010. 1826, 1827, 1830, 1831