Approximate Least Trimmed Sum of Squares Fitting and Applications in Image Analysis

Fumin Shen, Chunhua Shen, Anton van den Hengel, Zhenmin Tang

***Abstract*—The least trimmed sum of squares (LTS) regression estimation criterion is a robust statistical method for model fitting in the presence of outliers. Compared with the classical least squares (LS) estimator, which uses the entire data set for regression and is consequently sensitive to outliers, LTS identifies the outliers and fits to the remaining data points for improved accuracy. Exactly solving an LTS problem is NP-hard, but as we show here, LTS can be formulated as a *concave minimization problem*. Since it is usually tractable to globally solve a convex minimization or concave maximization problem in polynomial time, inspired by [1], we instead solve LTS' approximate complementary problem, which is convex minimization. We show that this complementary problem can be efficiently solved as a second order cone program (SOCP). We thus propose an iterative procedure to approximately solve the original LTS problem. Our extensive experiments demonstrate that the proposed method is robust, efficient and scalable in dealing with problems where data are contaminated with outliers. We show several applications of our method in image analysis.**

***Index Terms*—Least trimmed sum of squares regression, robust model fitting, outlier removal, second order cone programming, semidefinite programming.**

I. Introduction

In image analysis and computer vision, regression analysis is a fundamental technique for many model fitting problems. For example, many multi-view geometry problems are essentially model fitting and when an algebraic criterion is used, they are standard regression problems [2]. Since in most cases the measurement data are contaminated by noise and outliers, RANSAC has been used as the *de facto* tool to remove outliers under the *consensus set maximization* criterion [3][1]. RANSAC has been successfully applied to robust estimation, in particular in multi-view geometry problems. It works by iteratively performing sampling and consensus testing. However, there are drawbacks for RANSAC and its variants [3], [4]. First, RANSAC is a stochastic process: one may obtain different results when running RANSAC multiple times. Second, RANSAC cannot be applied to high dimensional problems. The number of samples required to achieve a set probability of success is exponential in the dimension of the problem. Also, because it usually needs many sampling steps, RANSAC can only be used on problems for which a solution can be efficiently computed. For some computer vision problems such as linear regression based face recognition [5], which is very different from multi-view parameter estimation problems, it is not clear how to apply RANSAC to improve the recognition accuracy.

In this work, we proffer a method for outlier removal within a least squares setting, which is equivalent to model fitting under the LTS criterion. It can be viewed as an approximate solution to the original LTS method, and is thus labeled *approximate least trimmed sum of squares* (ALTS). At an algorithmic level, the method is based on the simple heuristic of solving the inverse of the original LS optimization problem, i.e., maximum squares fitting, with a constraint on the number of data points to fit. We then remove the measurements that contribute to the maximum squares fitting, and solving again; the process is repeated until some stopping criterion is met. Our method has some desirable properties: a) Unlike RANSAC, our method is deterministic and non-heuristic. b) Compared with methods that solve an NP-hard problem, which is usually very computationally expensive, our method solves a second-order cone programming (SOCP) problem at each iteration and therefore it is efficient and scalable.

Our method is largely inspired by the outlier removal approach of Sim and Hartley [6], in the sense that both methods iteratively remove outliers and at each step, the methods fit the data and remove the measurement (or measurements) with greatest residuals. However, the fitting criteria for the two methods are very different. Our method approximately solves the least trimmed squares fitting problem. The ℓ_∞ norm minimization based outlier removal of Sim and Hartley [6] can be viewed as approximately solving the least k-quantile fitting problem. Clearly, both least trimmed squares and least k-quantile are well-regarded robust regression criteria. LTS is known to be statistically more efficient than least k-quantile. Usually, for low-dimensional problems where these estimators can be computed, LTS should be preferred. For higher-dimensional problems, the only known solutions offer uncontrolled approximations. Therefore, statistical efficiency of the minimizer and quality of approximation may be competing

This article has been accepted for publication in a future issue of IEEE Transactions on Image Processing, but has not been fully edited. Content may change prior to final publication. Manuscript received 05 August 2011; revised 04 December 2011; accepted 14 December 2012.

F. Shen and Z. Tang are with the School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, China This work was done when F. Shen was visiting The University of Adelaide.

C. Shen and A. van den Hengel are with Australian Center for Visual Technologies, and School of Computer Science, The University of Adelaide, SA 5005, Australia (e-mail: {chunhua.shen, anton.vandenhengel}@adelaide.edu.au). Correspondence should be addressed to C. Shen. This work was in part supported by Australian Research Council Future Fellowship (FT120100969).

[1]We argue that "trimmed fitting", "consensus set maximization", and "outlier removal" in most computer vision problems have the same definition. So hereafter we may use these terms interchangeably.

concerns.

The main contributions of our work are as follows.

- We first show that the LTS problem can be formulated as a concave minimization problem, which is NP-hard. Inspired by the work of [1], we also approximately solve the NP-hard problem by iteratively solving its complementary concave maximization problem. More importantly, by exploiting the structure of the problem, we derive an approximate complimentary optimization problem which is formulated as an SOCP. Off-the-shelf solvers such as Mosek [7] can be used to efficiently solve the SOCP problem. Compared with [1], in which a semidefinite programming (SDP) problem is formulated, our SOCP formulation is much more efficient.
- We apply the proposed algorithm to several computer vision applications and show that encouraging results are obtained.

Next, we briefly review previous work that is relevant to ours.

II. Prior work

We review some existing work on the topic of outlier removal and robust regression fitting before we present the proposed algorithm. As pointed in [6], "*It is difficult to cite references to this [fitting the largest residual and removing it], since papers using this method are usually rejected*", therefore we only review some work that is closest to ours. The most popular outlier removal method in computer vision is RANSAC [3]. The pros and cons of RANSAC are well known and discussed above.

The hardness of exactly solving the LTS problem has been shown in [8]—generally it is NP-hard. Li [9] proposed a robust least squares fitting method, which can be seen to solve a similar LTS fitting problem. The robust fitting problem there is converted into a bilinear programming problem, which is non-convex and NP-hard in general. A branch-and-bound (BnB) method is then used to globally solve the bilinear optimization problem. The downside of this method is that it is extremely slow and not scalable. Agulló [10] also used BnB to globally solve the LTS problem. With a standard desktop, the BnB method of [9] can only solve low dimensional (e.g., 2D, 3D) problems with less than 100 data points.[2] In contrast, our method only needs to solve SOCPs, which scales to problems with tens of thousands data points. Actually, it is known that some types of bilinear optimization problems can be equivalently converted into concave minimization problems [11]. It is not difficult to show the equivalence between the bilinear formulation in [9] and our concave minimization formulation. The essential difference is that our concave minimization formulation leads to an efficient approximate polynomial algorithm. It remains unclear whether one can obtain an efficient solver starting from the bilinear formulation in [9].

Lee et al. removes outliers by solving a convex sum-of-infeasibilities [12] problem. Olsson et al. [13] improved the method of [6] by considering Lagrange dual problem of the ℓ_∞ minimization problem in [6], the advantage being that less number of inliers are mistakenly removed by solving the dual problem.

Since Hartley and Schaffalitzky [14] introduced ℓ_∞ norm minimization, which makes finding a globally optimal solution for many multi-view geometry problems tractable, there is a growing interest in seeking a global solution for many computer vision problems. Two main directions have been identified in terms of global optimization in computer vision [15]. One approach has been to reformulate an originally non-convex problem into a convex one (or quasi-convex). Many ℓ_∞ norm minimization problems belong to this category [14], [16]. The other has been to use global optimization method for solving the non-convex problem, mainly using BnB [9], [10]. Our work here belongs to the former category as we re-formulate a non-convex problem into a convex one and solve the original problem approximately.

In robust statistics, much effort has been spent on robust least squares estimation [17], [18], [19], [10]. Rousseeuw and Leroy [18] have presented a thorough introduction to the LTS problem. They also proposed a re-sampling method termed PROGRESS for approximating the robust least median of squares (LMS) estimation. Rousseeuw and Van Driessen introduced the Fast-LTS method for approximately solving the LTS problem [19], which is faster than most robust LTS or LMS methods. In this work, we empirically show that our method is in general even faster than Fast-LTS.

The rest of the paper is structured as follows. In Section III, we show how to formulate the LTS problem as a concave minimization problem. Then in Section IV, we present our main contribution: we approximately solve the LTS problem using SOCP. We then apply the proposed method to a few image analysis applications in Section V. Finally, we conclude our paper in Section VI.

III. Problem setting

In this section, we present our main results. Our starting point is standard LS estimation. Robust linear regression is one of the most important problems in the area of statistics and many other applications. It is often conducted via minimizing the sum of squared residuals. Let $\mathbf{A} = [\mathbf{a}_1; \ldots; \mathbf{a}_n] \in \mathbb{R}^{n\times d}$ be the measurement data matrix. Here $\mathbf{a}_i \in \mathbb{R}^{1\times d}$ is the i-th row of the matrix. $\mathbf{y} \in \mathbb{R}^n$ is the vector of the model's response. $\boldsymbol{\beta} \in \mathbb{R}^d$ is the linear model's parameter that needs to be estimated.

The plain least squares (LS) estimation writes:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} (\mathbf{a}_i\boldsymbol{\beta} - y_i)^2, \tag{1}$$

or in a compact form,

$$\min_{\boldsymbol{\beta}} \|\mathbf{A}\boldsymbol{\beta} - \mathbf{y}\|^2, \tag{2}$$

for which we have a closed-form solution

$$\boldsymbol{\beta} = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{y}. \tag{3}$$

In the above minimization problem, the terms $(\mathbf{a}_i\boldsymbol{\beta} - y_i)^2$ are the squared residuals. Due to its simplicity and efficiency, least

[2] Personal communication with the author of [9].

squares estimation has been widely used. However, it utilizes the entire data set and therefore can be easily influenced by outliers. In some cases this can lead to extremely distorted estimates.

A variety of techniques have been developed to diminish the impact of outliers upon the final estimate. One of them is the LTS criterion, which minimizes the sum of the k smallest squared residuals while outliers with larger squared residuals are excluded.

The least trimmed squares fitting problem can be directly formulated as a mixed integer programming problem:

$$\min_{\boldsymbol{\pi}} \min_{\boldsymbol{\beta}} \sum_{i=1}^{n} (\mathbf{a}_i\boldsymbol{\beta} - y_i)^2 \cdot \pi_i \quad \text{s.t. } \mathbf{1}^\top\boldsymbol{\pi} = k,\ \pi_i \in \{0,1\}, \forall i = 1 \cdots n. \tag{4}$$

Ideally, when the squared residual $(\mathbf{a}_i\boldsymbol{\beta} - y_i)^2$ is larger, the corresponding π_i will be 0. The integer constraint can be replaced with a simple linear constraint as follows:

$$\min_{\boldsymbol{\pi}} \min_{\boldsymbol{\beta}} \sum_{i=1}^{n} (\mathbf{a}_i\boldsymbol{\beta} - y_i)^2 \cdot \pi_i \tag{5a}$$

$$\text{s.t. } \mathbf{1}^\top\boldsymbol{\pi} = k,\ 0 \le \boldsymbol{\pi} \le 1. \tag{5b}$$

Problems (4) and (5) are equivalent because of the following fact.

Theorem 1. *For a concave minimization problem, every local and global solution is attained at some extreme point of the feasible domain.*

The proof of this theorem can be found in [20]. When we fix $\boldsymbol{\beta}$ the optimization problem (5) is a linear program in $\boldsymbol{\pi}$. A linear function is also a concave function (convex at the same time). Therefore, the optimal solution for $\boldsymbol{\pi}$ must be achieved at an extreme point of the feasible set, for which $\boldsymbol{\pi}_i$ can only be either 0 or 1. So (4) and (5) have the same solution.

Let $\boldsymbol{\Pi}$ be a diagonal matrix with its diagonal elements being $\boldsymbol{\pi}$. If we fix $\boldsymbol{\pi}$, the above problem (the inner minimization) has a closed-form solution for the variable $\boldsymbol{\beta}$ and the problem can be largely simplified. We can rewrite the objective function into

$$F_{\text{LTS}} = (\mathbf{A}\boldsymbol{\beta} - \mathbf{y})^\top \boldsymbol{\Pi} (\mathbf{A}\boldsymbol{\beta} - \mathbf{y}),$$

and its gradient vanishes at the optimum, which leads to the solution for $\boldsymbol{\beta}$:

$$\boldsymbol{\beta}^\star = (\mathbf{A}^\top\boldsymbol{\Pi}\mathbf{A})^{-1}\mathbf{A}^\top\boldsymbol{\Pi}\mathbf{y}. \tag{6}$$

Now we can eliminate $\boldsymbol{\beta}$ from F_{LTS},

$$\begin{aligned} F_{\text{LTS}} &= (\mathbf{A}\boldsymbol{\beta}^\star - \mathbf{y})^\top \boldsymbol{\Pi} (\mathbf{A}\boldsymbol{\beta}^\star - \mathbf{y}) \\ &= 2[\mathbf{y}^\top\boldsymbol{\Pi}\mathbf{y} - \mathbf{y}^\top\boldsymbol{\Pi}\mathbf{A}(\mathbf{A}^\top\boldsymbol{\Pi}\mathbf{A})^{-1}\mathbf{A}^\top\boldsymbol{\Pi}\mathbf{y}]. \end{aligned} \tag{7}$$

Now the optimization problem writes

$$\boxed{\min_{\boldsymbol{\pi}}\ F_{\text{LTS}} \text{ as in (7), s.t. (5b).}} \tag{8}$$

The following theorem shows that the problem (8) is a concave minimization problem and in general, it is NP-hard. This is expected because the original problem (4) is a NP-hard combinatorial problem.

Theorem 2. *The objective function of* (7) *is concave in the variable* $\boldsymbol{\pi}$ *(or* $\boldsymbol{\Pi}$*).*

Proof: The first term is linear in $\boldsymbol{\Pi}$ and hence convex and concave. So we only need to check the second term, and all that we need to show is the convexity of $f(\boldsymbol{\Pi}) = \mathbf{y}^\top\boldsymbol{\Pi}\mathbf{A}(\mathbf{A}^\top\boldsymbol{\Pi}\mathbf{A})^{-1}\mathbf{A}^\top\boldsymbol{\Pi}\mathbf{y}$. We can establish the convexity of f via its epigraph:

$$\begin{aligned} \mathbf{epi}\, f &= \{(\boldsymbol{\Pi}, t) \mid \mathbf{y}^\top\boldsymbol{\Pi}\mathbf{A}(\mathbf{A}^\top\boldsymbol{\Pi}\mathbf{A})^{-1}\mathbf{A}^\top\boldsymbol{\Pi}\mathbf{y} \le t\} \\ &= \left\{ (\boldsymbol{\Pi}, t) \;\middle|\; \begin{bmatrix} \mathbf{A}^\top\boldsymbol{\Pi}\mathbf{A} & \mathbf{A}^\top\boldsymbol{\Pi}\mathbf{y} \\ \mathbf{y}^\top\boldsymbol{\Pi}\mathbf{A} & t \end{bmatrix} \succeq 0. \right\} \end{aligned} \tag{9}$$

Using the Schur complement condition for positive semidefiniteness of a block matrix [21], and because $\mathbf{A}^\top\boldsymbol{\Pi}\mathbf{A}$ must be p.s.d., the epigraph $\mathbf{epi}\, f$ must be a convex set. Therefore f is convex, which demonstrates that the function in (7) is concave. ∎

Note that this theorem has appeared in [1] and here we present a simpler proof. This result is not surprising. Many 0-1 integer programming problems can be equivalently converted into concave minimization problems [22]. For example, An integer program $\min_{\mathbf{x}}\ \mathbf{c}^\top\mathbf{x},\ \text{s.t. } \mathbf{A}^\top\mathbf{x} = \mathbf{b}, \mathbf{x} \in \{0,1\}^d$ can be written into $\min_{\mathbf{x}}\ \mathbf{c}^\top\mathbf{x} + M\sum_{i=1}^{d} x_i(1-x_i),\ \text{s.t. } \mathbf{A}^\top\mathbf{x} = \mathbf{b}, 0 \le \mathbf{x} \le 1$, with M an arbitrarily large positive number.

IV. Maximum residual outlier identification

Observing that the optimization problem (8) is a concave minimization problem, we instead solve its complementary problem, which is a convex minimization problem:

$$\max_{\boldsymbol{\pi}} \mathbf{y}^\top\boldsymbol{\Pi}\mathbf{y} - \mathbf{y}^\top\boldsymbol{\Pi}\mathbf{A}(\mathbf{A}^\top\boldsymbol{\Pi}\mathbf{A})^{-1}\mathbf{A}^\top\boldsymbol{\Pi}\mathbf{y} \quad \text{s.t. } \mathbf{1}^\top\boldsymbol{\pi} = p, 0 \le \boldsymbol{\pi} \le 1. \tag{10}$$

Thus the global optimum is easily obtained in polynomial time. We call the above problem as approximate least trimmed sum of squares (ALTS) fitting. Ideally, here p is the number of outliers that one wants to remove. Different from (4) or (8), by maximizing (rather than minimizing) the expression in (10), one can identify the outliers which are corresponding to π_i with value 1.

As we show in Theorem 2, using (9), we can easily reformulate the ALTS problem into a semidefinite program (SDP). Although polynomial-time interior-point algorithms for SDP have been well studied, they are usually not scalable. Next we show that (10) is essentially a second-order cone program (SOCP), which is much more efficient and scalable.

First, we introduce a new variable $\mathbf{w} \in \mathbb{R}^n$ such that $\mathbf{A}^\top\mathbf{w} = \mathbf{A}^\top\boldsymbol{\Pi}\mathbf{y}$. Problem (10) becomes

$$\min_{\boldsymbol{\pi},\mathbf{w}} \mathbf{w}^\top\boldsymbol{\Pi}^{-1}\mathbf{w} - \mathbf{y}^\top\boldsymbol{\Pi}\mathbf{y} \tag{11a}$$

$$\text{s.t. } \mathbf{A}^\top\mathbf{w} = \mathbf{A}^\top\boldsymbol{\Pi}\mathbf{y}, \mathbf{1}^\top\boldsymbol{\pi} = p, 0 \le \boldsymbol{\pi} \le 1. \tag{11b}$$

The objective function is actually $\sum_{i=1}^{n} w_i^2/\pi_i - \mathbf{y}^\top\boldsymbol{\Pi}\mathbf{y}$. We interpret $0/0 = 0$ here. Now we introduce another set of variables $\mathbf{t}$ such that $w_i^2/\pi_i \le t_i,\ t_i > 0$ for $i = 1, \ldots, n$. The hyperbolic constraint can be represented as a second-order cone constraint:

$$w^2 \le \pi t, \pi \ge 0, t \ge 0 \Longleftrightarrow \left\| \begin{bmatrix} 2w \\ \pi - t \end{bmatrix} \right\| \le \pi + t. \tag{12}$$

We have converted the ALTS problem into an SOCP:

$$\begin{aligned} &\min_{\boldsymbol{\pi},\mathbf{w},\mathbf{t}} \mathbf{1}^\top \mathbf{t} - \sum_{i=1}^{n} y_i^2 \pi_i \\ &\text{s.t.} \left\| \begin{bmatrix} 2w_i \\ \pi_i - t_i \end{bmatrix} \right\| \leq \pi_i + t_i \ (i = 1 \ldots n); \\ &\mathbf{A}^\top \mathbf{w} = \mathbf{A}^\top \boldsymbol{\Pi} \mathbf{y}, \ \mathbf{1}^\top \boldsymbol{\pi} = p, 0 \leq \boldsymbol{\pi} \leq 1. \end{aligned} \tag{13}$$

Note that $\mathbf{A}^\top \boldsymbol{\Pi} \mathbf{y} = \sum_{i=1}^{n} \pi_i y_i \mathbf{a}_i^\top$.

Note that the solution of (13) for $\boldsymbol{\pi}$ is not necessarily an integer because (13) is not exactly complementary to the original integer programming problem. It is possible that a fractional value in $[0, 1]$ is assigned to a π. Some heuristics can be used to round up the non-integer solution, as commonly applied in integer programming. It is guaranteed that, *the solution of* (13) *for* $\boldsymbol{\pi}$ *has at most* $d+1$ *non-integers. Here* d *is the dimension of the input data.* This result can be obtained by looking at the KKT conditions and the complementary conditions [21] of the original ALTS problem (10). These conditions show that $\boldsymbol{\pi}$ is determined by a system of equations with $d+1$ degrees of freedom. So there are at most $d+1$ non-integer solutions in $\boldsymbol{\pi}$. This result suggests that we may simply treat all non-integer components of $\boldsymbol{\pi}$ as outliers when d is not very large.

Also it is not difficult to see that at each iteration, at least one outlier can be identified.

To run ALTS for outlier removal, we need to determine the value for p first, which corresponds the targeted number of outliers to be removed at each iteration. This value may be empirically set to: the number of total outliers divided by the maximum iteration number.

V. Experimental results

Extensive experiments on a variety of fitting problems using data with different levels of random noise and varying fractions of outliers are described below in order to illustrate the effectiveness of the proposed approach. We also test the efficiency of our approach (ALTS by SOCP) on different problems comparing to other two LTS methods: ALTS by SDP and Fast-LTS [19]. The code for Fast-LTS is downloaded from `ftp://ftp.win.ua.ac.be/pub/software/agoras/newfiles/fastlts.gz`. All the experiments are conducted using Matlab, running on a PC with a Quad-Core 3.07G Hz CPU and 12GB RAM.

We test the proposed method on various problems in computer vision and image analysis, including robust face recognition, line fitting, circle fitting, and homography estimation. Despite the diversity of the problems, minimal modifications are required for each. We need only adjust the measurement data matrix $\mathbf{A}$, the response vector $\mathbf{y}$, and the number of outliers p to be removed for each new problem formulation.

A. Outlier detection

In order to evaluate the effectiveness of the method in identifying outliers a set of face images have been augmented with a variety of occlusions. We then follow linear regression classification (LRC) of [5] to train a regression model. Fig. 1 shows example images from the dataset, and the results of outlier detection. The first two rows represent typical images from the AR face dataset and show the forms of occlusion caused by scarves and sunglasses. The third row shows images from the Extended Yale B dataset which have had 40% of their pixels replaced with those of another image (see [23]).

Fig. 1. Examples demonstrating the performance of the method in detecting outliers. The first and third columns are the original corrupted facial images. The detected outlier pixels are set to white in the second and fourth columns.

With respect to the formulation given above each image of one subject in the dataset represents a column vector $\mathbf{a}_i$ and the measurement data matrix $\mathbf{A}$ the cumulation of these column vectors. The column vector $\mathbf{y}$ represents the query image. We have empirically set the value of the parameter p to around 30% of the image size for the sunglasses tests and 38% for the scarf tests. For the random block occlusion case, we set the parameter p as the size of the block dimension. The detected outlier pixels are set to be white as shown in the images on the right of the original corrupted images.

Note that ALTS does not assume knowledge of the position or other features of the corruption. The results show that most pixels corresponding to sunglasses or scarves are correctly detected. A small number of pixels around the eyes are misclassified as inliers, however, which may be due to the misalignment of the original images during cropping.

The fact that the pixel values of the monkey face image are similar to those of the human face image makes a per-pixel occlusion decision far more challenging than the previous examples. As in Fig. 1, however, ALTS detects most of the block correctly.

B. Robust face recognition

We now describe the results of experiments performed on five public face recognition datasets: the AT&T dataset, the Georgia Tech. dataset, the AR [24] dataset, Extended Yale B [25], and CMU-PIE [26]. To evaluate the proposed approximate least trimmed sum of squares algorithm, we compare it with both the well-known traditional classifiers PCA [27], LDA [28], and three related state-of-the-art methods for robust face recognition: linear regression classification (LRC) [5],

sparse representation-based classifier (SRC) [23] and the linear regression classification with collaborative representations (referred to as LRC-C) [29]. Note that our approach to robust face recognition is different from robust regression based methods such as [30], where usually a non-convex robust regression loss is used to replace the convex least squares loss.

Linear regression classification selects the class corresponding to $\boldsymbol{\beta}_i$ which has the smallest reconstruction error. A similar method is suggested in [29], which uses the training data belonging to all classes (instead of only the corresponding class) to train the model $\boldsymbol{\beta}$ and then set $\boldsymbol{\beta}_i$ as the ith subvector of $\boldsymbol{\beta}$ that corresponds to the ith class. The reconstruction error is then calculated using $\boldsymbol{\beta}_i$ for each class i.

Two SRC models were presented in [23]:

1) SRC0 denotes the standard SRC in [23] which solves: $\min \|\boldsymbol{\beta}\|_1, \text{ s.t. } \mathbf{y} = \mathbf{A}\boldsymbol{\beta}$.
2) SRC1 denotes the extended SRC [23] handling occlusion and corruption: $\min \|\mathbf{w}\|_1, \text{ s.t. } \mathbf{y} = \mathbf{B}\mathbf{w}$, where $\mathbf{B} = [\mathbf{A}, \mathbf{I}]$, and $\mathbf{w} = [\boldsymbol{\beta}, \mathbf{e}]$. $\mathbf{I}$ and $\mathbf{e}$ are the identity matrix and error vector respectively.

Both methods are included in the comparison testing below. Note that similar to LRC-C, SRC takes all training images (represented in column vectors) to formulate a larger measurement data matrix $\mathbf{A}$.

For the purpose of the face recognition experiments, outlier pixels are first removed using ALTS, leaving the remaining inlier set for processing by any regression based classifier. In the experiments listed below LRC has been used for this purpose.

In addition, three datasets are used to compare computation efficiency between our method and other robust LTS algorithms on robust face recognition problems. The first two databses, AR and Extended Yale B contains faces with disguises; and the third dataset is for artificial occlusion example. Extensive experiments are conducted using different image sizes and different training sample numbers.

The AT&T dataset The AT&T dataset, also known as the ORL database, consists of 10 images each of 40 subjects. The images have been taken at different times, with varying lighting conditions, multiple facial expressions (smiling or not smiling, open or closed eyes), adornments (glasses or no glasses) and rotations up to 20 degrees.

For the purpose of testing we have taken the first five images of each subject as a training set and the remaining five as a test set. All the images with dimension 112×92 are down sampled to 10×10. A comparison between the results of our method and those of several others is summarized in Table I. For PCA and LDA, four reduction dimensions are selected: 5, 10, 20 and 40. We set p in ALTS as 15% of the total pixels. As shown in Table I, ALTS achieves the best accuracy of 94%, which is 1.5% better than LRC and 0.5% better than the best SRC. It also outperforms the best result of LDA and PCA by 1% and 4% respectively. Althogh no occlusion or corruption appears on this dataset, ALTS still achieves a better recognition accuracy than LRC. This may be mainly because ALTS can effectively remove noisy pixels caused by lighting or other variations.

TABLE I
CLASSIFICATION ACCURACY (%) ON THE AT&T DATASET. FOR PCA AND LDA, WE HAVE REDUCED THE ORIGINAL DIMENSION TO FOUR DIFFERENT DIMENSIONS (5, 10, 20, AND 40). *NOTE THAT FOR LDA, THE MAXIMUM DIMENSION IS 39 BECAUSE THIS DATASET HAS 40 CLASSES.

Approach	D-5	D-10	D-20	D-40
PCA	72	85	88.5	90
LDA	66	81.5	91.5	93*
LRC	92.5			
SRC0	90.5			
SRC1	93.5			
ALTS	94			

These results show that ALTS is producing state-of-the-art performance on un-occluded face data, as we would hope.

The Georgia-Tech dataset The Georgia-Tech dataset consists of 15 images each of 50 subjects. The images were taken in two or three sessions at different times with different facial expressions, lighting conditions, scale and background.

The original 480×640 images are down sampled to 15×15 pixels. The first 8 images of each subject were used for training and the last 7 images are used as a test set. All experiments are conducted on the original images without any cropping or normalization. We set p as 15% of the total pixel number for ALTS. A comparison of ALTS with a variety of other methods is shown in Table II. For PCA and LDA, four reduced dimensions are selected: $5, 10, 20$ and 40. ALTS achieves the best recognition accuracy of 92.9% which is slightly higher than LRC, LRC-C, SRCs and the best PCA, LDA results. This is consistent with the results on the AT&T dataset.

Faces occluded by sunglasses The AR dataset [24] consists of over 4000 facial images from 126 subjects (70 men and 56 women). For each subject 26 facial images were taken in two separate sessions. The images exhibit a number of variations including various facial expressions (neutral, smile, anger, and scream), illuminations (left light on, right light on and all side lights on), and occlusion by sunglasses and scarves.

Of the 126 subjects available 100 have been randomly selected for testing (50 males and 50 females) and the images cropped to 112×92 pixels. 8 images of each subject with various facial expressions but without occlusions were selected

TABLE II
CLASSIFICATION ACCURACY (%) ON THE GEORGIA TECH. DATASET. FOR PCA AND LDA, WE HAVE REDUCED THE ORIGINAL DIMENSION TO FOUR DIFFERENT DIMENSIONS (5, 10, 20, AND 40).

Approach	D-5	D-10	D-20	D-40
PCA	90	91.7	92.3	92
LDA	92.3	92	92	92
LRC	92			
LRC-C	92			
SRC0	92			
SRC1	92.3			
ALTS	92.9			

Fig. 2. Two examples demonstrating the performance of ALTS in reconstructing face images occluded by sunglasses from the AR dataset. Faces from left to right are original occluded images, the images reconstructed by ALTS, and the difference between the original image and the reconstructed images.

Fig. 3. Two examples demonstrating the performance of ALTS in reconstructing images of faces occluded by scarves from the AR dataset. faces from left to right are the original occluded images, the images reconstructed by ALTS, and the difference between the original images and the reconstructed images.

for training. Testing was carried out on 2 images of each of the selected subjects wearing sunglasses.

Two example test images of subjects wearing sunglasses from the AR dataset and their corresponding reconstructed images are shown in Figs. 2 and 3. Fig. 4 (top) shows a comparison of the recognition rates of LDA, PCA, LRC, SRCs and ALTS with respect to various feature spaces of dimension 42, 154, 644, and 2576. The SRC algorithms have been conducted on the first three feature spaces only due to their extremely high computation time. PCA was carried out with a final dimension of $5, 10, 20, 40, 80, 160$, and 320, and the best result reported. For LDA, only the first 5 dimension options are selected as there are only 100 classes in this task. It is obvious from Fig. 4 that the method proposed here significantly outperforms the other methods listed. In addition it is worth noting that on a feature space of dimension 644 ALTS achieved a recognition accuracy of 94.5% which outperforms SRC1 by 32.5% and LDA by 74%. For LRC-C [29], a similar technique to SRC1 is used to model occlusions, which writes $\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{B}\boldsymbol{\beta}\|_2^2$ where $\mathbf{B} = [\mathbf{A}, \mathbf{M}]$. $\mathbf{M}$ is a matrix designed for this dataset, which has fewer columns than the identity matrix $\mathbf{I}$ and is better than $\mathbf{I}$ to represent non-face objects [29]. However LRC-C does not perform well in this experiment, and only achieved an accuracy of 53.5% at dimension 2576.

In Fig. 5 (top), we show the computational time comparison of detecting outlier pixels in a typical face image with sunglasses using Fast-LTS and the ALTS algorithms which are implemented by both an SDP and SOCP. We can clearly see that ALTS by SOCP is much faster than Fast-LTS on all feature dimensions. When the feature dimension is low (less than about 600), ALTS implemented by SDP is only slightly slower than ALTS by SOCP and much faster than Fast-LTS. However, when the feature dimension is higher, ALTS by SDP becomes much more time consuming than that by SOCP and also slower than Fast-LTS. Note that in this experiment settings there are only 8 samples in each class. *When the problem dimension d is larger, the difference of efficiency between our method and ALTS by SDP is much more significant*, which is shown next.

Faces occluded by scarves The test set for the scarf occlusion tests were 2 images of each of the selected subjects from the AR dataset wearing scarves. Fig. 4 (bottom) shows a comparison of recognition rates for LDA, PCA, LRC, LRC-C, SRCs and ALTS with various feature spaces. Similar as the case for sunglasses (top), the proposed method significantly outperforms its competitors. Note also that on the feature space of dimension 504, ALTS achieved 84% recognition accuracy which is better than SRC0 by 54.5% and better than LRC by 76%.

Since LRC-C admits a closed-form solution and it is in general very efficient [29], we also test LRC-C on images without down-sampling at dimension 112×92 and it achieves a recognition accuracy of 80%. LRC-C performs much better than SRC and LRC at high dimensions. However, it is still worse than ALTS.

In Fig. 5 (bottom) we compare the computation time of

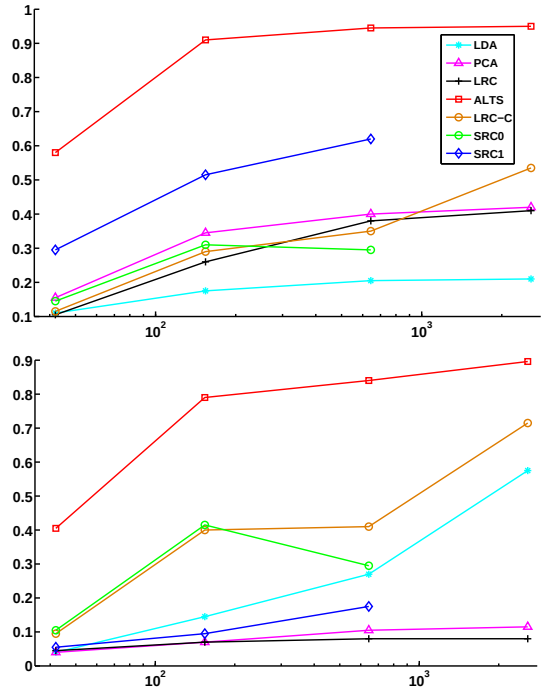

Fig. 4. Recognition accuracy (y-axis) for 100 subjects with sunglasses (top), and scarves (bottom) on the AR dataset. The x-axis is the feature dimension in log scale.

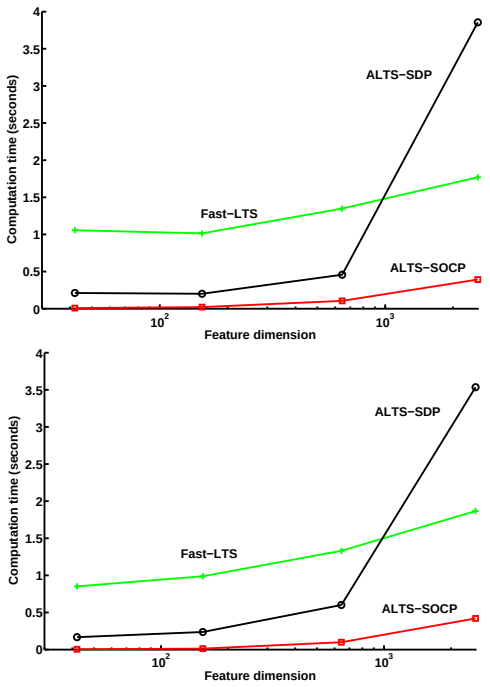

Fig. 5. Computational time comparison of ALTS implemented by SDP and SOCP as well as Fast-LTS for robust face recognition with different feature dimensions: 42, 154, 644, 2576. Plots shown on the top and bottom are results on the "sunglasses" and "scarves" subsets of the AR dataset, respectively.

detecting outlier pixels in a face image with scarf for different methods. Similar as in the time comparison of the sunglasses experiment (the first plot of Fig. 5), we can see that ALTS-SOCP runs much faster than the other two methods in higher feature space.

Contiguous block occlusions In order to evaluate the performance of the algorithm in the presence of artificial noise and larger occlusions we now describe testing involving replacing image pixels with those from another source. The Extended Yale B dataset [25] consists of 2414 frontal face images from 38 subjects under various lighting conditions. The images are cropped and normalized to 192×168 pixels [31]. Half of the images were randomly selected for training (about 32 images per subject), and the remaining half for testing. In our experiment, all the images are down sampled to 24×21 pixels.

Two example occluded images with 20% and 40% (by area) occlusion by randomly placed monkey faces, and their ALTS reconstructions are shown in Fig. 6. In order to evaluate the performance of the various methods on this data each was run on 3 sets of images with randomly placed occlusions at 3 sizes each. For ALTS, we set p as the outliers number divided by iteration number T. Results corresponding to two values for $T = 1$ and 5 are reported. Table III shows that ALTS and SRC1 achieve comparable accuracies ($92.6 \pm 0.6\%$ and $92.3 \pm 1.0\%$) on the 20% occlusion data which is a significant improvement over other compared methods. When $T = 1$, SRC1 outperforms ALTS with 30% and 40% occlusion, however, the performance of ALTS is significantly better than that of standard SRC and LRC. When $T = 5$, it is not surprising

Fig. 6. Two examples demonstrating the performance of ALTS in reconstructing face images occluded by monkey faces in Extended Yale B dataset. Faces from left to right are the images occluded by monkey face, the images reconstructed by ALTS, and the difference image between the occluded image and reconstructed images. The top row is with 20% occlusion and bottom row is with 40% occlusion.

that ALTS achieves better recognition accuracies (89.1% and 82.2% on the 30% and 40% occlusion data respectively), which are even better than those of SRC1 and significantly better than SRC0 and LRC. When it is run only one iteration ($T = 1$), our method can remove more inliers.

We then test the efficiencies of our method ALTS by SOCP and other two LTS methods, ALTS by SDP and Fast-LTS, on the Extended Yale B dataset in tow scenarios. Fig. 7 shows the computation time comparisons of detecting a random block of a typical face image from the Extended Yale B database by these three methods. Fig. 7 (top) demonstrates that with the number of training samples being fixed, ALTS by SOCP performs much better than that by SDP and Fast-LTS on most feature dimensions: 12×10, 24×21, and 48×42. Note that we do not report the running time of Fast-LTS on image size 6×5 because Fast-LTS does not allow the problem dimension (number of training samples per subject for face recognition) d larger than the observation number n and here $n = 30, d = 32$.

From Fig. 7 (bottom) we can see that with the feature dimension fixed (on 24×21) ALTS by SOCP perform much faster than that by SDP when with a larger number of training images. Similar as in [1], Fig. 7 (bottom) shows that with a relatively low dimension ($d < 20$ and $n = 504$ here) ALTS by SDP performs faster than Fast-LTS. However when d is larger, ALTS-SDP becomes much more time consuming than the other two methods due to SDP's high polynomial computation complexity.

Partial face features on the CMU-PIE dataset As shown above, occlusion can deteriorate face recognition performance.

TABLE III
RECOGNITION RATES (%) IN THE PRESENCE OF RANDOMLY PLACED BLOCK OCCLUSIONS OF IMAGES FROM THE EXTENDED YALE B DATASET.

Approach	20%	30%	40%
LRC	82.3 ± 0.7	69.1 ± 1.6	53.9 ± 1.5
SRC0	80.1 ± 1.1	66.5 ± 0.6	54 ± 0.5
SRC1	92.6 ± 0.6	88.5 ± 0.2	82.1 ± 1.5
ALTS ($T = 1$)	92.3 ± 1	80.2 ± 0.8	65.6 ± 0.9
ALTS ($T = 5$)	93.2 ± 0.8	89.1 ± 1.6	82.2 ± 1.5

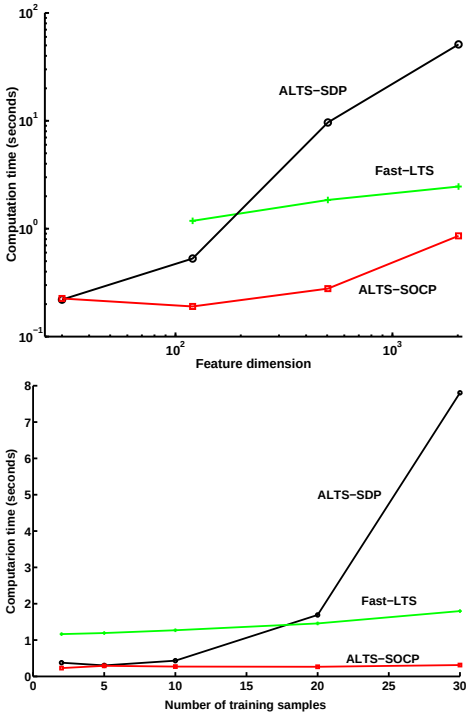

Fig. 7. Time comparison of ALTS implemented by SDP and SOCP and Fast-LTS for face recognition problem on the Extended Yale B dataset in two scenarios: different feature dimensions (6×5, 12×10, 24×21, and 48×42) with fixed number (32) of training samples of each subject; and different numbers of training samples of each subject with fixed feature dimension of 24×21. Both the x-axis and y-axis of the left chart are in log-scale.

Intuitively, occlusion at different face locations will perform differently. In this experiment, we investigate the effectiveness of using only partial face features. Here we compare our approach with other methods using two different partial features: eyebrows and eyes, mouth and chin. Different from [23], in which only the extracted partial features are used and other parts of a face image are discarded, we take an opposite strategy: the pixels corresponding to one of these partial features are set to zeros and we keep the remaining pixels. Following this strategy, if a partial feature is more important than others, worse performance must be achieved with this feature being dead pixels.

In this experiment, we use the CMU-PIE dataset [26]. It contains 68 subjects with 41368 face images, each person under 13 different poses, 43 different illumination conditions, and with 4 different expressions. In our experiment all of the face images are aligned and cropped, and resized to 24×24 pixels. Here we use the subset containing images of pose C27 (a nearly front pose) and we only use the data from the first 20 subjects, each subject with 21 images.

We normalize each image to be a unit vector. The first 15 images of each subject are used for training and the rest 6 for test. The test images are preprocessed so that one part of faces (about $1/3$ pixels) are set to be dead pixels. See Fig. 8 for illustration. The parameter p is set to $1/3$ of feature dimension. Results are reported in Table IV. We do not report SRC0 because here $n=576$ and $d=300$ ($n>d$) makes the SRC equation system $\mathbf{y}=\mathbf{A}\boldsymbol{\beta}$ overdetermined. This problem

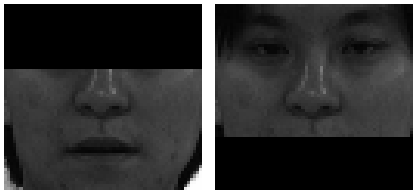

Fig. 8. Typical face images from the CMU-PIE dataset. Around $1/3$ of the image pixels are set to zeros at locations of "eyebrows and eyes" or "mouth and chin".

TABLE IV
RECOGNITION ACCURACIES (%) OF VARIOUS METHODS ON THE CMU-PIE DATASET WITH DIMENSION 24×24 CONTAINING 1/3 DEAD PIXELS AT DIFFERENT POSITIONS (SHOWN IN THE FIRST ROW). FOR ALTS, RESULTS FOR TWO VALUES OF T: 33% AND 43% OF THE NUMBER OF ALL PIXELS ARE REPORTED IN THE LAST TWO ROWS.

Approach	eyes & eyebrows	mouth & chin
LRC	30.83	40
SRC1	56.67	65
ALTS (33%)	70	88.33
ALTS (43%)	75.83	94.17

is solved by SRC1 which replaces $\mathbf{A}$ with $\mathbf{B}=[\mathbf{A},\mathbf{I}]$ [23]. In order to eliminate the adverse influence of black pixels, here for our algorithms, we remove one pixel at each iteration, i.e., $p=1$ for ALTS and in total T iterations are needed. Results for different values of T: 33% and 43% of all pixels are reported. From Table IV, we can easily see that ALTS performs the best. In particular, with the mouth and chin part replaced with black pixels, when T is set as 43% of pixels number ALTS achieves a recognition accuracy 94.17% which is considerably higher than that of LRC and SRC1.

It is noteworthy that all these algorithms obtain a worse performance when the test images lose the information of eyes and eyebrows than in the other situation. This is consistent with the argument from [32] that the eyes and eyebrows part of human faces is most informative for face recognition.

For these robust LTS methods, the computation time is reported in Table V. Consistent with the results of previous experiments, ALTS-SOCP is much more efficient than the other two methods.

C. Geometric model fitting

Having shown that ALTS is effective in reconstructing partially occluded images we now investigate its performance in more classical parameter estimation. The first problem we approach in this vein is line fitting.

Line fitting Fig. 9 shows the performance of our algorithm (marked with label "ALTS") and least squares (marked with label "LS") on data generated under two different models.

TABLE V
COMPUTATION TIME (IN SECONDS) OF VARIOUS LTS METHODS FOR REMOVING OUTLIER PIXELS OF A FACE IMAGE OF CMU-PIE DATASET.

Approach	eyes & eyebrows	mouth & chin
Fast-LTS	1.52	1.51
ALTS-SDP	0.78	0.86
ALTS-SOCP	0.09	0.06

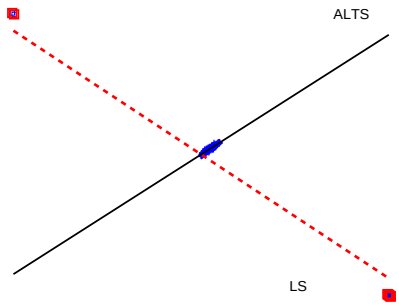

Fig. 10. An example demonstrating the performance of our algorithm on an extremely case. Outliers detected by our algorithm are marked as squares. The solid line is fitted by our method and the corresponding least squares fit is the dashed line. In this case, we set $n = 210$ and $k = 200$.

We generate $\mathbf{A} \in \mathbb{R}^{n\times d}$ randomly and $\boldsymbol{\beta} \in \mathbb{R}^d$ randomly. We then set the first k error terms $\epsilon_j, j \in [1, \ldots, k]$ as independent standard normal random variables. We set the last $n-k$ error terms as independent chi squared random variables with 5 degrees of freedom. We also test using the two-sided contamination model which sets the sign of the last $n-k$ variables randomly such that the outliers lie on both side of the true regression line. In both cases we set $\mathbf{y} = \mathbf{A}\boldsymbol{\beta} + \epsilon$. For ALTS, p is set $n-k$ for two-sided contamination model. In order to remove all the outliers in one side contamination model, only one outlier is removed and totally $n-k$ iterations are needed. As can be seen in Fig. 9, ALTS detects all of the outliers and consequently generates a line estimate which fits the inlier set well for both noise models, where LS achieves a reasonable estimate only for the two-sided contamination case where the inlier and outlier distributions are centered about the same line. Fig. 10 shows a more extreme example, where 10 outliers are located far from the inliers. As would be expected LS produces a skewed result, where ALTS has correctly identified the outliers and produced a perfect line estimate. For this experiment, most existing approaches fail, including RANSAC and the method of Sim and Hartley [6].

Table VI shows a comparison of computation time of Fast-LTS and the ALTS algorithms implemented by SDP and SOCP under the two-sided contamination model. We place 10 percent of the points as outliers, and the parameter p is set as the number of outliers placed. We compare efficiencies of different methods for problems with dimension d varying from 2 to 50 (shown in the first column of Table VI) and $n \leq 2000$ (shown in the first row). It is clear that with all $100 \leq n \leq 2000$ ALTS implemented by SOCP is much faster than that by SDP. This is consistent with the theoretical analysis: the complexity of interior point methods for SOCP is much lower than that of SDP [21]. Fast-LTS also performs slower than ALTS by SOCP for all n and d and also slower than ALTS by SDP when the problem size is small, i.e., $n \leq 500, d \leq 5$. We notice that with observation number n and problem dimension d becoming larger, the SDP algorithm becomes more time consuming significantly and slower than Fast-LTS when $d \geq 20, n \geq 200$.

Then we use the average trimmed absolute residuals [1] for accuracy comparison of Fast-LTS, ALTS and least squares fitting, which is shown in Table VII. We can see that, consistent with the result in [1], the average trimmed absolute residuals of ALTS are on par compared with those of Fast-LTS, and are much better than those of traditional least squares.

Circle fitting An example of the performance of ALTS in circle fitting is shown in Fig. 11. The 100 inlier points were selected from a distribution uniformly around the perimeter of a circle centered at $(0, 0)$ with radius 10, and had noise sampled from $\mathcal{N}(0, 1)$ added. The 26 outliers ($p = 26$) were randomly generated approximately following a uniform distribution inside the box. Fig. 11 shows that ALTS has correctly identified the inlier and outlier sets, and demonstrates that the center and radius estimated by ALTS fit the true inlier data well.

Homography estimation The final problem for which the performance of the ALTS algorithm is presented is that of estimating homographies from real image data. Fig. 12 (top) shows the image pairs, keypoint correspondences and labels provided in [33]. There are totally 74 pairs of correspondences, 52 of them are good points (labeled as inliers) and 22 are noise (labeled as outliers). Comparison of performance and computation efficiency between RANSAC and ALTS is summarized in Table VIII. In this case ALTS and RANSAC were both based on the normalized Direct Linear Transformation (DLT) method for homography estimation [2]. We run ALTS to remove $p = 5$ outliers at each iteration till an all-inlier set is obtained. For RANSAC, different distance thresholds t are chosen (see Table VIII). For each t we run 10 times and the average results are reported.

We use the RANSAC implementation of [34]. The true inliers detected by ALTS and RANSAC are shown in the middle and bottom of Fig. 12 respectively. Both RANSAC and ALTS correctly identifies an inlier set and a good approximation of the correct homography, but rejects a proportion of the points labeled as inliers by [33]. In terms of computation efficiency, our ALTS implemented by SOCP is about eight times faster than RANSAC to obtain an all-inlier set.

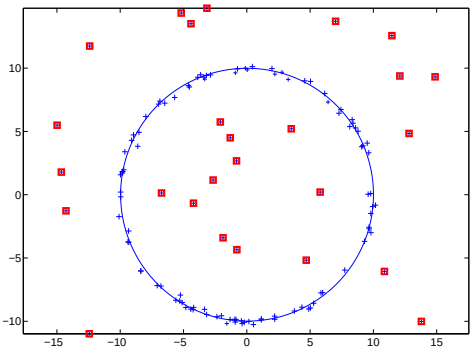

Fig. 11. An example of the performance of ALTS in circle fitting. Points identified by ALTS as outliers are marked by red squares.

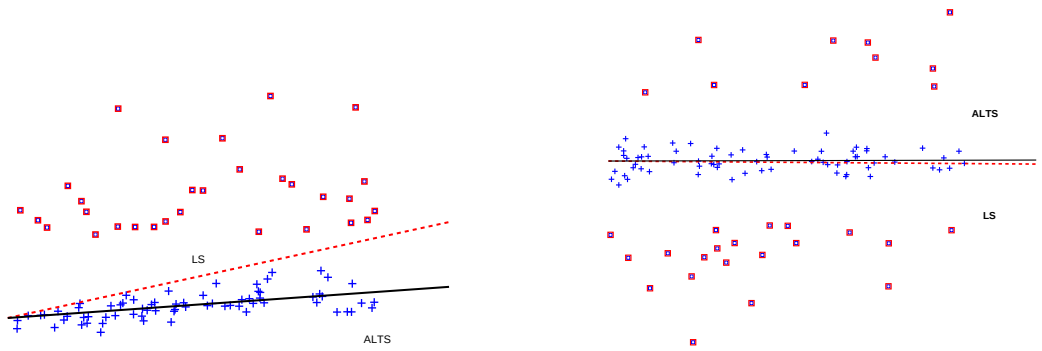

Fig. 9. Two examples demonstrating the performance of our algorithm on one sided contamination data (left) and the two sided contaminated data (right). Outliers detected by our algorithm are marked as squares. The solid line is fitted by our method and the corresponding least squares fit is the dashed line. In these two cases, we set $n = 100$ and $k = 70$.

TABLE VI

COMPUTATIONAL TIME (IN SECONDS) COMPARISON OF DIFFERENT ROBUST FITTING METHODS UNDER TWO SIDED CONTAMINATION MODEL. THE FIRST COLUMN IS THE PROBLEM DIMENSION d AND THE FIRST ROW IS THE OBSERVATION NUMBER n. "FAST", "SDP", AND "SOCP" REPRESENT FAST-LTS [19], ALTS-SDP [1] AND OUR PROPOSED FAST SOCP METHOD, RESPECTIVELY. WE CAN SEE THAT THE PROPOSED METHOD IS IN GENERAL MUCH FASTER THAN ITS COMPETITORS.

	100			200			500			1000			2000		
	Fast	SDP	SOCP	Fast	SDP	SOCP	Fast	SDP	SOCP	Fast	SDP	SOCP	Fast	SDP	SOCP
2	0.91	0.16	**0.01**	1.02	0.18	**0.01**	1.22	0.29	**0.04**	1.22	0.59	**0.11**	1.33	1.86	**0.29**
5	0.94	0.18	**0.01**	1.05	0.23	**0.02**	1.23	0.38	**0.05**	1.26	0.71	**0.13**	1.37	2.21	**0.32**
10	0.98	0.26	**0.01**	1.08	0.33	**0.02**	1.35	0.62	**0.07**	1.34	1.46	**0.16**	1.48	3.86	**0.40**
20	1.05	0.27	**0.02**	1.20	1.21	**0.03**	1.53	2.64	**0.89**	1.54	5.61	**0.22**	1.64	12.47	**0.53**
30							1.92	9.82	**0.12**	1.79	20.51	**0.31**	1.91	61.23	**0.69**
50										2.92	248.37	**0.50**	3.12	391.20	**1.17**

TABLE VII

AVERAGE ABSOLUTE RESIDUALS COMPARISON OF DIFFERENT FITTING METHODS: THE PROPOSED ALTS, FAST-LTS (SHOWN AS "FAST"), LS_{in} AND LS. THE THIRD RESULT LS_{in} IS THE AVERAGE RESIDUAL OF LEAST SQUARES FITTING USING ONLY INLIER OBSERVATIONS, I.E., GROUND TRUTH. LS IS THE LEAST SQUARES FITTING USING ALL OBSERVATIONS.

	100				200				500				1000				2000			
	ALTS	Fast	LS_{in}	LS	ALTS	Fast	LS_{in}	LS	ALTS	Fast	LS_{in}	LS	ALTS	Fast	LS_{in}	LS	ALTS	Fast	LS_{in}	LS
2	0.96	0.96	0.96	3.60	0.89	0.88	0.90	5.40	1.00	1.00	1.00	1.66	1.02	1.02	1.02	1.45	1.00	1.00	1.00	1.27
5	0.98	0.97	0.95	6.43	0.88	0.87	0.88	4.89	0.98	0.97	0.97	3.31	0.98	0.98	0.98	3.08	0.97	0.97	0.97	1.75
10	1.00	1.16	1.00	9.46	0.98	1.00	0.98	5.20	1.08	1.05	1.05	5.38	0.95	0.95	0.95	3.33	1.01	1.01	1.01	2.36
20	0.88	2.32	0.66	10.41	1.07	1.07	0.92	10.67	1.05	0.98	0.98	4.88	0.99	0.98	0.98	3.59	1.03	1.03	1.03	3.16
30									1.18	0.93	0.92	5.70	1.10	0.97	0.97	5.16	1.04	0.96	0.96	4.07
50													0.96	0.95	0.94	13.18	1.00	1.00	1.00	13.15

TABLE VIII

COMPARISON OF PERFORMANCE AND COMPUTATION TIME BETWEEN RANSAC AND ALTS ON HOMOGRAPHY ESTIMATION. N_{det} IS THE NUMBER OF DETECTED INLIERS AND N_{fa} IS THE NUMBER OF DETECTED FALSE INLIERS. FOR ALTS, $p = 5$ OUTLIERS ARE REMOVED AT EACH ITERATION TILL AN ALL INLIERS SET IS OBTAINED. THE COMPUTATION TIME IS SHOWN IN SECONDS.

RANSAC			
residual threshold t	N_{det}	N_{fa}	computation time
0.01	40	2	0.08
0.005	37	1	0.13
0.001	26	0	0.59
Our ALTS			
iterations	N_{det}	N_{fa}	computation time (accumulated)
1	52	17	0.01
3	46	13	0.03
6	39	5	0.05
8	33	1	0.06
9	29	0	0.07

VI. CONCLUSION

We have proposed a second-order cone programming formulation of the approximate least squares estimation problem. We have demonstrated its efficiency over existing approximate LTS methods.

In solving this problem using second order cone programming, we arrive at a robust regression method suitable for application to a range of key problems in computer vision. Extensive synthetic and real image testing shows the method to be efficient and scalable to the extent required by the application domain, and robust to the types of noise exhibited.

Like many other robust fitting methods, the proposed method needs to know how many outliers to be removed. One may heuristically determine this value. In the future, we plan to investigate how to estimate the degree of noise in the contaminated data.

REFERENCES

[1] T. Nguyen and R. Welsch, "Outlier detection and least trimmed squares approximation using semi-definite programming," *Computational Statistics & Data Analysis*, vol. 54, pp. 3212–3226, 2010.

[2] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[3] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Comm. of the ACM*, vol. 24, pp. 381–395, 1981.

[4] O. Chum and J. Matas, "Optimal randomized RANSAC," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, pp. 1472–1482, 2008.

[5] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 11, pp. 2106–2112, 2010.

[6] K. Sim and R. Hartley, "Removing outliers using the l_∞ norm," in *Proc. IEEE Comp. Vis. Pattern Recog.*, 2006, pp. 485–494.

[7] Mosek ApS, "MOSEK version 6.0: Optimization toolbox for matlab manual," 2010, http://www.mosek.com/.

[8] D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "On the least trimmed squares estimator," Technical report. http://www.cs.umd.edu/~mount/pubs.html, 2007.

[9] H. Li, "Consensus set maximization with guaranteed global optimality for robust geometry estimation," in *Proc. Int. Conf. Comp. Vis.*, 2009, pp. 1074–1080.

[10] J. Agulló, "New algorithms for computing the least trimmed squares regression estimator," *Comput. Stat. & Data Anal.*, vol. 36, no. 4, pp. 425–439, 2001.

[11] T. V. Theiu, "A note on the solution of bilinear programming problems by reduction to concave minimization," *Math. Program.*, vol. 41, pp. 249–260, 1988.

[12] H. Lee, Y. Seo, and S. W. Lee, "Removing outliers by minimizing the sum of infeasibilities," *Image Vision Comput.*, vol. 28, no. 6, pp. 881–889, 2010.

[13] C. Olsson, A. Eriksson, and R. Hartley, "Outlier removal using duality," in *Proc. IEEE Conf. Comp. Vis. Pattern Recog.*, 2010, pp. 1450–1457.

[14] R. I. Hartley and F. Schaffalitzky, "l_∞ minimization in geometric reconstruction problems," in *IEEE Conf. Comp. Vis. Pattern Recog.*, 2004.

[15] R. Hartley and F. Kahl, "Optimal algorithms in multiview geometry," in *Proc. Asian Conf. Comp. Vis.*, vol. 1, 2007, pp. 13–34.

[16] F. Kahl and R. Hartley, "Multiple view geometry under the L_∞-norm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1603–1617, 2008.

[17] D. M. Hawkins, "The feasible solution algorithm for least trimmed squares regression," *Comput. Stat. & Data Anal.*, vol. 17, no. 2, pp. 185–196, 1994.

[18] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. John Wiley & Sons, Inc., 1987.

[19] P. J. Rousseeuw and K. Van Driessen, "Computing LTS regression for large data sets," *Data Mining & Knowl. Discov.*, vol. 12, pp. 29–45, 2006.

[20] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, 1970.

[21] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[22] M. Raghavachari, "On connections between zero-one integer programming and concave programming under linear constraints," *Operations Research*, vol. 17, no. 4, pp. 680–684, 1969.

[23] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, pp. 210–227, 2009.

[24] A. M. Martinez and R. Benavente, "The AR face database," *CVC, Tech. Rep.*, 1998.

[25] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, 2001.

[26] T. Sim, S. Baker, and M. Bsat, "The cmu pose, illumination, and expression database," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, pp. 1615–1618, 2003.

[27] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[28] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, pp. 711–720, 1997.

[29] Q. Shi, A. Eriksson, A. van den Hengel, and C. Shen, "Is face recognition really a compressive sensing problem?" in *Proc. IEEE Conf. Comp. Vis. Pattern Recog.*, 2011, pp. 553–560.

[30] I. Naseem, R. Togneri, and M. Bennamoun, "Robust regression for face recognition," *Pattern Recog.*, vol. 45, no. 1, pp. 104–118, 2012.

[31] K. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 684–698, 2005.

[32] P. Sinha, "Face recognition by humans: Nineteen results all computer vision researchers should know about," in *Proc. the IEEE*, 2006, pp. 1948–1962.

[33] T.-J. Chin, J. Yu, and D. Suter, "Accelerated hypothesis generation for multi-structure robust fitting," in *Eur. Conf. Comput. Vis.*, 2010, pp. 533–546.

[34] P. Kovesi, "MATLAB and Octave functions for computer vision and image processing," http://www.csse.uwa.edu.au/~pk/research/matlabfns/.

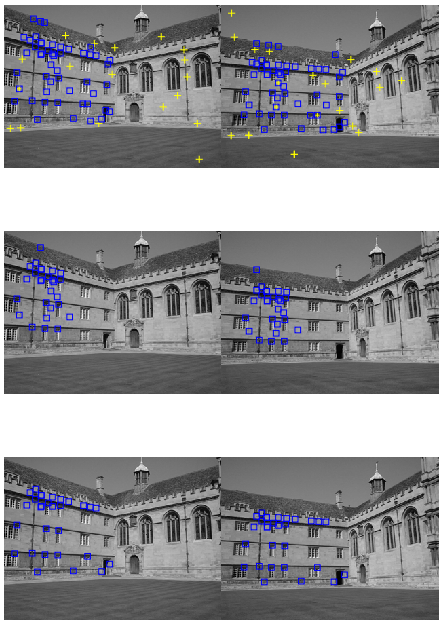

Fig. 12. Image pairs used for the homography fitting test. True inliers are marked as blue squares and outliers as yellow crosses (top). Points from correspondences detected as inliers by ALTS (middle) and RANSAC (bottom) are marked with blue squares.