Non-smooth M-estimator for Maximum Consensus Estimation

Huu Le[1]
huu.le@qut.edu.au
Anders Eriksson[2]
anders.eriksson@qut.edu.au
Michael Milford[2]
michael.milford@qut.edu.au
Thanh-Toan Do[2]
thanh-toan.do@adelaide.edu.au
Tat-Jun Chin[2]
tat-jun.chin@adelaide.edu.au
David Suter[3]
d.suter@ecu.edu.au

[1] School of Electrical Engineering and Computer Science, Queensland University of Technology, Australia.
[2] School of Computer Science, The University of Adelaide, Australia.
[3] School of Science, Edith Cowan University, Australia.

Abstract

This paper revisits the application of M-estimators for a spectrum of robust estimation problems in computer vision, particularly with the maximum consensus criterion. Current practice makes use of smooth robust loss functions, e.g. Huber loss, which enables M-estimators to be tackled by such well-known optimization techniques as Iteratively Re-weighted Least Square (IRLS). When consensus maximization is used as loss function for M-estimators, however, the optimization problem becomes non-smooth. Our paper proposes an approach to resolve this issue. Based on the Alternating Direction Method of Multiplier (ADMM) technique, we develop a *deterministic algorithm* that is *provably convergent*, which enables the maximum consensus problem to be solved in the context of M-estimator. We further show that our algorithm outperforms other differentiable robust loss functions that are currently used by many practitioners. Notably, the proposed method allows the sub-problems to be solved efficiently in parallel, thus entails it to be implemented in distributed settings.

1 Introduction

Robust model estimation, especially with the maximum consensus criterion, is a crucial problem that underpins a large number of geometric estimation tasks in computer vision [13]. It is employed extensively as an intermediate step in many important applications; for instance, image stitching [4], Simultaneous Localization and Mapping (SLAM) [19], large-scale Structure from Motion (SfM)[24], to name a few. Due to the noisy and imperfect nature of data acquisition devices, the quality of estimated models depend heavily on the ability to discard bad measurements (outliers) present in the dataset. Practically, consensus maximization is arguably one of the most popular criteria for robust geometric fitting. It has been established that consensus maximization is NP-hard [5], thus finding its globally optimal solution is computationally challenging. Although there has been much research on developing exact algorithms for this problem with encouraging results [6, 9, 18, 20, 27], globally optimal algorithms are still impractical for many real-world applications where fast processing time is crucial. Therefore, consensus maximization is mainly approached by using randomized hypothesize-and-verify methods such as the well-known RANSAC algorithm [12], and its improved variants [7, 8, 17]. While those methods are quite efficient for input data containing low proportion of outliers, they can become exceptionally slow for highly contaminated measurements. Furthermore, besides the probabilistic stopping criterion, there exists no further information about the convergent characteristics for such randomized approaches.

In contrast to the class of randomized algorithms, M-estimators are preferred in many scenarios as its convergence – up to local optimality – can be guaranteed with proper choices of robust loss functions. Indeed, it has been shown in [1] that if the robust loss functions are designed in such a way that they satisfy some pre-defined criteria, IRLS can be used to solve M-estimators with provable convergent property. Among the criteria required by [1], being a smooth function is mandatory. Unfortunately, that does not apply to the maximum consensus problem as its loss function is non-smooth (which will be illustrated in the following sections). Consequently, the common practice is to estimate the model parameters with a smooth robust loss function, then evaluate the consensus set based on the returned estimate. Clearly, the real problem is not tackled with the above-mentioned approach.[1]

Contributions Currently, there exists no methods in the literature that can solve M-estimator with maximum consensus loss function such that the analytical convergence is guaranteed. This paper fills that gap by proposing a *provably convergent* ADMM-based algorithm that is *deterministic* to iteratively solve the consensus optimization problem up to local optimality. By introducing auxiliary variables, our algorithm splits the main problem into smaller sub-problems that can be solved efficiently using bisection. This allows our algorithm to be *easily parallelized* for efficient computation of large scale geometric estimation problems. Our method can be used as a refinement scheme to improve the solution quality for consensus maximization from a rough estimate provided by RANSAC [12] or other heuristic strategies.

Related work We would like to remark that besides the class of smooth M-estimators, there are other deterministic approaches for the problem [16, 20, 21]. However, those methods do not tackle the original cost function of the consensus maximization problem, as they either solve the relaxed version [21] or depend solely on heuristic that may remove genuine inliers, which can result in bad estimate if the data is not well balanced [20]. Recently, an exact penalty approach was introduced by the authors in [16] that converges to a local solution of the maximum consensus problem. Their approach, however, can only handle the ℓ_1 or ℓ_∞ norm of the residual functions. A similar refinement scheme was proposed by [23], where a surrogate function is used to approximate the original problem, which can be solved by Iterative Re-weighted ℓ_1 (IRL1). Our proposed method is distinguished from the rest by its capability to handle the ℓ_2 quasi-convex transfer errors and the ability to be implemented in a parallel manner.

[1] Later, we will show in the experiment section that using smooth loss functions for the maximum consensus problem may lead to poor solutions.

2 Preliminaries

2.1 Maximum Consensus

Given a set of N measurements, the maximum consensus problem aims to find an estimate $\boldsymbol{\theta}^*$ that is consistent with as many of the data points as possible. All the consistent data instances (inliers) with respect to the solution $\boldsymbol{\theta}^*$ form the optimal consensus set $\mathcal{I}^*$. Mathematically, consensus maximization can be written as the following optimization problem

$$\max_{\boldsymbol{\theta}, \mathcal{I} \subseteq \mathcal{P}(N)} \quad |\mathcal{I}|, \quad \text{subject to} \quad f_i(\boldsymbol{\theta}) \leq \varepsilon, \quad \forall i \in \mathcal{I}, \tag{1}$$

where $|\mathcal{I}|$ denotes the cardinality of the consensus set $\mathcal{I}$, and $\mathcal{P}(N)$ represents the *powerset* (set of all subsets) of the set $\{1, 2, \ldots, N\}$. The functions $f_i(\boldsymbol{\theta})$ are the residual functions and ε is the inlier threshold. In this work, we focus on the quasi-convex residual functions that has the fractional form of

$$f_i(\boldsymbol{\theta}) = \frac{\|\mathbf{a}_i \boldsymbol{\theta} + \mathbf{b}_i\|_2}{\mathbf{c}_i^T \boldsymbol{\theta} + d_i}, \quad \text{s.t. } \mathbf{c}_i^T \boldsymbol{\theta} + d_i > 0. \tag{2}$$

Such type of residuals can be found in many geometric estimation problems, e.g. homography estimation, triangulation, structure from motion with known rotation, camera resectioning, etc. Note that (2) can also be generalized well to many robust linear regression problems. Interested readers are referred to [15] for more detailed discussions about quasi-convexity and its applications.

To put consensus maximization into the context of M-estimation, let us first rewrite (1) as an outlier minimization problem:

$$\min_{\boldsymbol{\theta}} \quad \sum_i \Phi(f_i(\boldsymbol{\theta})). \tag{3}$$

Note that from the restriction on the denominator of (2), all the residuals functions must be non-negative, i.e., $f_i(\boldsymbol{\theta}) \geq 0 \;\; \forall i$. Taking advantage of this, in order for (3) to be equivalent to (1), the function Φ can be defined as:

$$\Phi(x) = \begin{cases} 0 \text{ if } 0 \leq x \leq \varepsilon, \\ 1 \text{ otherwise.} \end{cases} \tag{4}$$

Intuitively, a data point that is not an inlier with respect to the estimate $\boldsymbol{\theta}$ will be penalized by (4), thus by solving (3), one gets the estimate that maximizes the consensus size.

2.2 Traditional M-estimators

The formulation (3) of the consensus maximization problem does indeed resemble the commonly used formulation of a wide class of traditional M-estimators, where the Φ function (4) has been put in place of the smooth robust loss functions Ψ that are often discussed in many relevant works [1, 14]. For instance, the Huber loss function

$$\Psi_{\text{huber}}(x) = \begin{cases} \frac{1}{2}x^2 \;\text{ for } |x| \leq b, \\ b(|x| - b), \;\text{ otherwise,} \end{cases} \tag{5}$$

where b is a scalar, is quite popular for many practitioners. For smooth loss functions, iterative reweighted least squares (IRLS) is a commonly used technique to obtain the M-estimate. As proven by [1], if the selected robust function satisfies some conditions, IRLS is guaranteed to converge.

The Φ function (4) for the maximum consensus problem, however, does not enjoy the properties required for convergence due to its non-smoothness. Certainly, one can get over this problem by performing a robust fitting using one of the smooth loss functions, e.g. (5), then obtain the consensus set using the returned results. However, the cost function of the consensus maximization problem and the M-estimators with smooth loss functions are totally different in nature. Thus, using the approximation technique mentioned above does not really solve the maximum consensus problem.

In this paper, we introduce a provably convergent approach to solve the problem (1) within the context of M-estimators using the non-smooth loss function (4). Our method relies on the Alternating Direction Method of Multiplier (ADMM) optimization scheme. The use of ADMM for computer vision problems has gained much interest recently [10, 11, 28] as ADMM provides a convenient way to solve an optimization problem by solving multiple sub-problems in parallel. Its use for robust geometric estimation, however, has not been thoroughly explored. To the best of our knowledge, this is the first work that apply ADMM technique for the task of robust model fitting by the M-estimator approach.

3 ADMM-Based formulation for Consensus Maximization

In this section, we will show how the consensus maximization problem (1) can be tackled by using ADMM and demonstrate that with this technique, the task of solving (1) can be performed in a distributed setting by tackling smaller sub-problems at the same time. Interestingly, the sub-problems are special instances of Quadratic Program (QP) with only one Quadratic constraint [2], which can be solved up to global optimality using bisection.

By introducing N auxiliary variables $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \ldots \boldsymbol{\theta}_N$, (1) can be rewritten equivalently as a constrained optimization problem:

$$\min_{\boldsymbol{\theta},\boldsymbol{\theta}_1,\ldots\boldsymbol{\theta}_N} \quad \sum_{i=1}^{N} \Phi(f_i(\boldsymbol{\theta}_i)), \quad \text{s.t.} \quad \boldsymbol{\theta} = \boldsymbol{\theta}_1 = \cdots = \boldsymbol{\theta}_N. \tag{6}$$

Furthermore, to support the convergence of our algorithm[2], observe that solution of (6) does not change if we write it in the following form

$$\min_{\boldsymbol{\theta},\boldsymbol{\theta}_1,\ldots\boldsymbol{\theta}_N} \quad \sum_{i=1}^{N} \left(\Phi(f_i(\boldsymbol{\theta}_i)) - \mu\|\boldsymbol{\theta}_i\|^2\right) + \mu N\|\boldsymbol{\theta}\|^2, \quad \text{s.t.} \quad \boldsymbol{\theta} = \boldsymbol{\theta}_1 = \cdots = \boldsymbol{\theta}_N. \tag{7}$$

Here, μ acts as a regularization parameter, which is fixed during the optimization process. In our experiments, this parameter is set to 0.001. Note, however, that at convergence, μ does not change the nature of the problem, i.e., the solution of (7) is also the solution that we would like to seek for problem (6). See the convergence proof provided in the supplementary material for more details.

[2] See supplementary material

The augmented Lagrangian of (7) with a penalty parameter ρ can be formulated as

$$\begin{aligned}\mathcal{L}_{\rho}(\{\boldsymbol{\theta}_i\},\boldsymbol{\theta},\{\boldsymbol{\lambda}_i\}) = \; & \sum_{i=1}^{N}\left(\Phi(f_i(\boldsymbol{\theta}_i)) - \mu\|\boldsymbol{\theta}_i\|^2\right) + N\mu\|\boldsymbol{\theta}\|^2 \\ & + \rho\sum_{i=1}^{N}\left(\|\boldsymbol{\theta}_i - \boldsymbol{\theta} + \boldsymbol{\lambda}_i\|^2 - \|\boldsymbol{\lambda}_i\|^2\right)\end{aligned} \tag{8}$$

where each $\boldsymbol{\lambda}_i$ represents the scaled Lagrangian multiplier that associates with the auxiliary variable $\boldsymbol{\theta}_i$. Note that (8) was written in the scaled form of ADMM [3].

3.1 ADMM update

The core idea behind ADMM is to iteratively update the variables, starting from the auxiliary variables $\{\boldsymbol{\theta}_i\}$, then the original variables $\boldsymbol{\theta}$, by minimizing the augmented Lagrangian (8) with respect to that particular variable, while the rest of the other variables are fixed. Finally, the Lagrangian multipliers $\boldsymbol{\lambda}_i$ are also updated by accumulating the difference between the auxiliary variable and the original ones. Specifically, the update steps at the $(t+1)-$th iteration include:

$\boldsymbol{\theta}_i$ update:

$$\boldsymbol{\theta}_i^{(t+1)} \leftarrow \arg\min_{\boldsymbol{\theta}_i} \mathcal{L}_{\rho}(\{\boldsymbol{\theta}_j^{t+1}\}_{j=1}^{i-1},\ \boldsymbol{\theta}_i,\ \{\boldsymbol{\theta}_k^{t}\}_{k=i+1}^{N}, \boldsymbol{\theta}^t, \boldsymbol{\lambda}^t) \tag{9}$$

$\boldsymbol{\theta}$ update:

$$\boldsymbol{\theta}^{(t+1)} \leftarrow \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{\rho}(\{\boldsymbol{\theta}_i^{t+1}\}_{j=1}^{N},\ \boldsymbol{\theta}, \boldsymbol{\lambda}^t) \tag{10}$$

$\boldsymbol{\lambda}_i$ update:

$$\boldsymbol{\lambda}_i^{t+1} \leftarrow \boldsymbol{\lambda}_i^{t} + \boldsymbol{\theta}_i^{t+1} - \boldsymbol{\theta}^{t+1} \tag{11}$$

3.2 Detailed update steps

3.2.1 $\boldsymbol{\theta}_i$ update

Interestingly, by dissecting further into the sub-problems, it can be observed that the updating steps for $\{\boldsymbol{\theta}_i\}$ can be solved efficiently up to global optimality.

From (9), updating $\boldsymbol{\theta}_i$ amounts to solving the problem:

$$\min_{\boldsymbol{\theta}_i} \Phi(f_i(\boldsymbol{\theta}_i)) - \mu\|\boldsymbol{\theta}_i\|^2 + \rho\|\boldsymbol{\theta}_i - \boldsymbol{\theta} + \boldsymbol{\lambda}_i\|^2 \tag{12}$$

Clearly, (12) is a non-smooth optimization problem caused by the non-smoothness of the Φ function. Fortunately, since the outcome of the Φ function can only be either 0 or 1, the solution of (12) can be achieved by considering the solutions of the following two sub-problems:

$$\min_{\boldsymbol{\theta}_i} \quad -\mu\|\boldsymbol{\theta}_i\|^2 + \rho\|\boldsymbol{\theta}_i - \boldsymbol{\theta} + \boldsymbol{\lambda}_i\|^2 \quad \text{s.t.} \quad 0 \le f_i(\boldsymbol{\theta}_i) \le \varepsilon, \tag{13}$$

$$\min_{\boldsymbol{\theta}_i} \quad 1 - \mu\|\boldsymbol{\theta}_i\|^2 + \rho\|\boldsymbol{\theta}_i - \boldsymbol{\theta} + \boldsymbol{\lambda}_i\|^2 \quad \text{s.t.} \quad f_i(\boldsymbol{\theta}_i) > \varepsilon \text{ or } f_i(\boldsymbol{\theta}_i) < 0 \tag{14}$$

The intuition behind the sub-problems (13), (14) lies in the fact that in other to solve (12) to update $\boldsymbol{\theta}_i$, we search for its solution $\boldsymbol{\theta}_i^*$ over two sub-domains, where the first sub-domain contains $\boldsymbol{\theta}_i$ such that $\Phi(\boldsymbol{\theta}_i) = 0$ (or $0 \leq f_i(\boldsymbol{\theta}_i) \leq \varepsilon$) and the other sub-domains contain $\boldsymbol{\theta}_i$ such that $\Phi(\boldsymbol{\theta}_i) = 1$ (or $f_i(\boldsymbol{\theta}_i) > \varepsilon \vee f_i(\boldsymbol{\theta}_i) < 0$). Generally speaking, these sub-problems are non-convex. However, due to the fact that each of them contains only one quadratic constraint, they are special cases of quadratic programs with one quadratic constraint [2]. Therefore, the optimal solutions to these subproblems can be achieved by employing well-known methods such that SDP relaxation, or bisection [2, 22]. In this work, we employ the bisection approach described in [22]. After obtaining the solutions to (13) and (14), the solution that induces smaller objective value is then used to update $\boldsymbol{\theta}_i$. More details are provided in the supplementary material.

3.2.2 $\boldsymbol{\theta}$ update

Finally, after all $\{\boldsymbol{\theta}_i\}$ are updated, $\boldsymbol{\theta}$ can be revised by solving a convex quadratic optimization problem:

$$\min_{\boldsymbol{\theta}} \quad \mu N \|\boldsymbol{\theta}\|^2 - \rho \sum_i \|\boldsymbol{\theta}_i - \boldsymbol{\theta} + \boldsymbol{\lambda}_i\|^2, \tag{15}$$

which, then can be computed by:

$$\boldsymbol{\theta} = \frac{\rho}{N(\rho - \mu)} \sum_i (\boldsymbol{\theta}_i + \boldsymbol{\lambda}_i) \tag{16}$$

3.3 Convergence

Theorem 1 *With a sufficiently large ρ, the ADMM iterations in (9), (10) and (11) converge after a finite number of steps:*

$$\lim_{t \to \infty} \|\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}^{(t-1)}\|_2 = 0$$

Also, it converges to a stationary point $(\{\boldsymbol{\theta}_i^*\}, \boldsymbol{\theta}^*)$ of the Lagrangian (8) such that

$$\boldsymbol{\theta}_1^* = \boldsymbol{\theta}_2^* = \cdots = \boldsymbol{\theta}_N^* = \boldsymbol{\theta}^*$$

Proof This section outlines the proof. The detailed proof is provided in the supplementary material. Firstly, we prove that with a sufficiently large ρ, the Lagrangian function (8) is monotonically non-increasing.

Consider the $(t+1)$-th update iteration. As the update steps of $\boldsymbol{\theta}_i$ can be solved up to global optimality, it follows that

$$\mathcal{L}_\rho(\{\boldsymbol{\theta}_i\}^{(t+1)}, \boldsymbol{\theta}_i^t, \{\boldsymbol{\lambda}_i\}^t) \leq \mathcal{L}_\rho(\{\boldsymbol{\theta}_i\}^t, \boldsymbol{\theta}_i^t, \{\boldsymbol{\lambda}_i\}^t). \tag{17}$$

Then, after $\boldsymbol{\theta}$ and all the variables $\{\boldsymbol{\lambda}_i\}$ are updated, it can be proven that with a sufficiently large ρ, the following inequality holds [see supplementary material]

$$\mathcal{L}_\rho(\{\boldsymbol{\theta}_i\}^{(t+1)}, \boldsymbol{\theta}_i^{(t+1)}, \{\boldsymbol{\lambda}_i\}^{(t+1)}) \leq \mathcal{L}_\rho(\{\boldsymbol{\theta}_i\}^{(t+1)}, \boldsymbol{\theta}_i^t, \{\boldsymbol{\lambda}_i\}^t) \tag{18}$$

From (17) and (18), it can be assured that after each update iteration,

$$\mathcal{L}_\rho(\{\boldsymbol{\theta}_i\}^{(t+1)}, \boldsymbol{\theta}_i^{(t+1)}, \{\boldsymbol{\lambda}_i\}^{(t+1)}) \leq \mathcal{L}_\rho(\{\boldsymbol{\theta}_i\}^t, \boldsymbol{\theta}_i^t, \{\boldsymbol{\lambda}_i\}^t), \tag{19}$$

given that ρ is sufficiently large.

Moreover, it can also be proven that the Lagrangian is lower-bounded. Therefore, it convergence is guaranteed. See supplementary material for more details. ∎

3.4 Main algorithm

Algorithm 1 ADMM-based M-estimator (AMES).

```
Require: Data $\{\mathbf{a}_i, \mathbf{b}_i, \mathbf{c}_i, d_i\}_{i=1}^{N}$, initial model parameter $\boldsymbol{\theta}_0$, initial penalty value $\rho^0$, increment rate $\sigma$, threshold $\delta$.
$t \leftarrow 0$
$\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}_0$.
$\boldsymbol{\theta}_i{}^{(t)} \leftarrow \boldsymbol{\theta}_0$;  $\boldsymbol{\lambda}_i{}^{(t)} \leftarrow \mathbf{0}\ \forall i = 1 \ldots N$.
while true do
    $t \leftarrow t + 1$
    Update $\boldsymbol{\theta}_i{}^{(t)}$ by solving (9) $\forall\, i = 1..N$
    Update $\boldsymbol{\theta}^{(t)}$ using (16)
    Update $\boldsymbol{\lambda}^{(t)}$ using (11)
    if $\|\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}^{(t-1)}\| \le \delta$ then
        $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}^{(t)}$
        Break.
    end if
    $\rho^{(t)} \leftarrow \sigma \cdot \rho^{(t-1)}$.
end while
return $\boldsymbol{\theta}^*$.
```

Based on the analysis developed in the previous sections and the convergence proof discussed in Sec. 3.3, this section provides the pseudo-code for the main algorithm. Note, however, that although we have proved that the algorithm converges with a sufficiently large ρ, setting ρ to a large value at the first few iterations may lead to a poor solution as the problem is non-convex in general. Therefore, similar to other ADMM-based algorithms, we propose to initialize the parameter ρ with a small value of ρ^0 ($0.1 \le \rho^0 \le 5.0$) and increase ρ after each iteration by an incremental rate $\sigma > 1.0$. The algorithm terminates once first order conditions are sufficiently satisfied, i.e., the norm of the difference between the two successive $\boldsymbol{\theta}$ is less than a threshold δ.

4 Experiments

Our proposed algorithm (AMES) can be used for a wide range of robust model fitting applications in computer vision, including problems with linear constraints (algebraic errors) and quasi-convex residuals with ℓ_2 norm. In this section, the performance of AMES compared to other approaches will be evaluated. As the main goal of this work is to develop an algorithm for M-estimator with the maximum consensus loss function, we focus on benchmarking our algorithm against other smooth robust loss functions, including Huber (HB) and Cauchy (CC) loss. In addition, we also compare AMES against other representative deterministic methods that solve consensus maximization sub-optimally, which include ℓ_∞ outlier removal [25], ℓ_1 approximate method [21], IRL1 [23] and exact penalty (EP) method [16]. We execute RANSAC [12](RS) to get the baseline results and use them as initializations for all other local methods. Note that RANSAC's variants are not compared as we would like to focus on benchmarking the proposed algorithm among the class of deterministic algorithms. All methods were implemented in MATLAB on a Ubuntu machine with 8 cores and 32GB of RAM. For M-estimators with smooth loss function, we use the robust fitting packages from MATLAB's toolboxes. Our AMES implementation is available at: https://github.com/intellhave/AMES.

4.1 Linear regression with synthetic data

First, we test our proposed algorithm on the problem of robust linear regression with synthetic data. The input data $\mathbf{X} \in \mathbb{R}^{N\times 6}$, where $N = 2000$, and an estimate $\boldsymbol{\theta} \in \mathbb{R}^6$ are generated randomly, with each row $\mathbf{x}_i$ of $\mathbf{X}$ represents a data point. The vector $\mathbf{y}$ is generated by first computing $\mathbf{y} = \mathbf{X}\boldsymbol{\theta}$, then each y_i is perturbed with a Gaussian noise of $\sigma_{\text{noise}} = 0.1$. Outliers are simulated by randomly selecting a subset of $p\%$ elements of the vector $\mathbf{y}$ and corrupt them by adding a Gaussian noise of $\sigma_{\text{outliers}} = 1$. The data is then shifted to make it unbalanced, i.e., most of the points are distributed on one side of the hyperplane [16]. The residual function with respect to an estimate $\boldsymbol{\theta}$ is $f_i(\boldsymbol{\theta}) = |\mathbf{x}_i\boldsymbol{\theta} - y_i|$. It can be realized that this is a special case of the quasi-convex residual (2) by setting all $\mathbf{c}_i$ to a vector of all zeros and all d_i to 1.

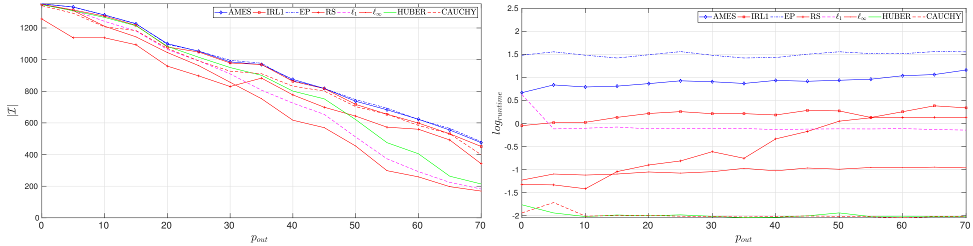

Figure 1: Consensus size (top) and run times in log scale (bottom) for different methods in linear estimation experiment. RS: RANSAC [12], ℓ_1: [21], ℓ_∞: [25], IRL1: [23], EP: [16]

Figure 1 plots the consensus size and the runtime at termination for all the methods with different values of the outlier rate p, where $p = 10, 15, 20, \ldots, 70$. It can be observed that AMES outperformed other M-estimators with smooth loss functions. This justifies the fact that one can get a better estimation quality by using the right loss function. Approximate methods such as ℓ_1 and ℓ_∞ outlier removal, as anticipated, perform well when the outlier rate is low. However, when the number of outliers increases, as shown in Figure 1, their performance start to deteriorate.

Our algorithm is comparable to the exact penalty method proposed in [16], although our algorithm can converge faster. We would like to emphasize, however, that the weakness of [16] is that they can not handle the residuals functions in the form of (2) (with ℓ_2 norm), which makes our method a strong candidate for applications that require ℓ_2 norm. Henceforth, we focus on experiments with quasi-convex residuals, namely, the homography estimation and affine image registration.

4.2 Homography estimation with quasi-convex residuals

With this experiment, we show that by using the right loss function, one gets an estimate with better quality (larger consensus size) for many popular computer vision applications. We randomly select the image pairs from the AdelaideRMF dataset [26], and the Zurich Building dataset[3] to test our proposed algorithm for the task of of estimating a homography matrix with quasi-convex residuals. The pairs of images are listed in Table 1. For each image pair, SIFT feature was extracted by VFleat toolbox to get a set of approximately 500 to 1000 putative correspondences. All the methods are initialized with the same starting point (using RANSAC). Table 1 summarizes the results for all the methods. AMES has demonstrated its ability to consistently upgrade the initial solution to better solutions with higher consensus size. Observe that, similar to the linear case, we are able to get higher consensus size compared to other methods, with faster run time. It can be seen from Table 1

[3] http://www.vision.ee.ethz.ch/showroom/zubud/

Algorithms		RS	ℓ_1	ℓ_∞	HB	CC	IRL1	EP	AMES
Union House	$\lvert\mathcal{I}\rvert$	434	430	409	453	447	440	**462**	**462**
N = 836	time (s)	0.64	6.95	1.14	9.75	25.11	4.86	22.84	12.67
Classic Wing	$\lvert\mathcal{I}\rvert$	512	425	411	507	504	450	512	**524**
N = 916	time (s)	0.72	8.53	1.18	19.03	22.97	3.65	26.74	16.74
Elder Hall	$\lvert\mathcal{I}\rvert$	333	299	276	**352**	337	296	**352**	345
N = 553	time (s)	0.72	8.53	1.18	19.03	22.97	3.65	26.74	16.74
Napier	$\lvert\mathcal{I}\rvert$	601	601	577	609	597	588	**618**	**618**
N = 792	time (s)	0.67	5.66	0.63	14.34	33.6	1.25	20.63	13.31
University	$\lvert\mathcal{I}\rvert$	516	575	484	549	534	**576**	**576**	**576**
N = 692	time (s)	0.54	3.88	0.96	17.73	14.2	0.63	13.77	10.54
Valbonne	$\lvert\mathcal{I}\rvert$	512	476	446	514	476	490	**520**	**520**
N = 789	time (s)	0.64	5.57	1.23	17.19	29.66	1.33	16.29	18.11
Invalides	$\lvert\mathcal{I}\rvert$	330	310	314	335	329	308	**342**	338
N = 558	time (s)	0.41	2.01	0.49	9.97	12.91	1.19	10.11	10.3
Building 39	$\lvert\mathcal{I}\rvert$	373	390	295	415	400	406	**412**	**412**
N = 660	time (s)	0.52	3.15	2.15	7.16	13.41	1.91	15.8	11.28

Table 1: Homography estimation results with quasi-convex residuals. RS: RANSAC [12], ℓ_1: [21], ℓ_∞: [25], IRL1: [23], EP: [16]. HB/CC: Huber/Cauchy loss function, respectively.

Algorithms		RS	ℓ_1	ℓ_∞	HB	CC	IRL1	EP	AMES
Bikes	$\lvert\mathcal{I}\rvert$	589	619	619	613	281	621	620	**622**
N = 666	time (s)	4.63	3.15	0.25	8.66	6.62	0.31	7.12	7.68
Tree	$\lvert\mathcal{I}\rvert$	475	596	**598**	477	277	361	**598**	**598**
N = 636	time (s)	4.63	3.15	0.25	8.66	6.62	0.31	7.12	7.68
Boat	$\lvert\mathcal{I}\rvert$	350	423	481	458	429	450	486	**488**
N = 896	time (s)	4.63	3.15	0.25	8.66	6.62	0.31	7.12	7.68
Graff	$\lvert\mathcal{I}\rvert$	279	203	334	331	280	330	335	**336**
N = 663	time (s)	4.66	3.14	0.7	10.26	8.39	2.56	9.21	9.09
Raglan	$\lvert\mathcal{I}\rvert$	160	132	165	160	163	180	192	**202**
N = 519	time (s)	3.52	1.33	1.55	7.37	4.9	0.75	9.5	9.6
Bld 192	$\lvert\mathcal{I}\rvert$	258	242	216	261	259	193	267	**276**
N = 507	time (s)	3.38	1.18	0.42	12.47	8.91	0.66	7.14	13.19
Bld 155	$\lvert\mathcal{I}\rvert$	269	265	**274**	269	270	259	**274**	**274**
N = 308	time (s)	2.01	0.21	0.1	5.54	5.53	0.13	2.71	9.12

Table 2: Affine image registration results with quasi-convex residuals. RS: RANSAC [12], ℓ_1: [21], ℓ_∞: [25], IRL1: [23], EP: [16]. HB/CC: Huber/Cauchy loss function, respectively.

that the performance of M-estimators with smooth loss functions are inconsistent, and with the same starting points, most of the time their solutions are worse than AMES.

4.3 Affine image registration

The proposed algorithm is also tested on affine image registration with the image pairs from the Oxford VGG dataset[4] and the Zurich Building dataset (the pairs are listed in Table 2). The list of methods in the homography estimation experiment are also compared agains AMES. Results are shown in Table 2. Similar to the homography experiments, with the same starting point, AMES consistently converges to better solution quality (larger consensus size) compared to other smooth robust functions and we also outperform other refinement schemes in term of consensus size with comparable run time.

5 Conclusions

We introduced an ADMM-based approach to solve the consensus maximization in the context of M-estimator. Unlike traditional M-estimators with smooth robust loss functions, our algorithm is the first algorithm to tackle the non-smooth loss function such that its convergence is guaranteed. We provided the proof for our algorithm together with the experiments to show that by using the right loss function, one gets a better solution quality than using smooth loss functions as approximation. Our algorithm can easily be parallelized, which makes it a promising candidate for deploying to distributed optimization frameworks.

Acknowledgements This research was supported by the Australian Research Council through the grant ARC FT170100072. Milford was partially supported by the ARC Future Fellowship FT140101229. Chin and Suter were partially supported by DP160103490.

References

[1] Khurrum Aftab and Richard Hartley. Convergence of iteratively re-weighted least squares to robust m-estimators. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 480–487. IEEE, 2015.

[2] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[3] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[4] Matthew Brown and David G Lowe. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74(1):59–73, 2007.

[5] Tat-Jun Chin and David Suter. *The Maximum Consensus Problem: Recent Algorithmic Advances*. Morgan & Claypool Publishers, 2017.

[6] Tat-Jun Chin, Pulak Purkait, Anders Eriksson, and David Suter. Efficient globally optimal consensus maximisation with tree search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2413–2421, 2015.

[4] http://www.robots.ox.ac.uk/~vgg/data/

[7] Ondrej Chum and Jiri Matas. Matching with prosac-progressive sample consensus. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 220–226. IEEE, 2005.

[8] Ondřej Chum, Jiří Matas, and Josef Kittler. Locally optimized ransac. In *DAGM*. Springer, 2003.

[9] Olof Enqvist, Erik Ask, Fredrik Kahl, and Kalle Åström. Robust fitting for multiple view geometry. In *European Conference on Computer Vision*, pages 738–751. Springer, 2012.

[10] Anders Eriksson and Mats Isaksson. Pseudoconvex proximal splitting for l-infty problems in multiview geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4066–4073, 2014.

[11] Anders Eriksson, John Bastian, Tat-Jun Chin, and Mats Isaksson. A consensus-based framework for distributed bundle adjustment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1754–1762, 2016.

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

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

[14] Peter J Huber et al. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

[15] Qifa Ke and Takeo Kanade. Quasiconvex optimization for robust geometric reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10): 1834–1847, 2007.

[16] Huu Le, Tat-Jun Chin, and David Suter. An exact penalty method for locally convergent maximum consensus. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017.

[17] Karel Lebeda, Jirı Matas, and Ondrej Chum. Fixing the locally optimized ransac–full experimental evaluation. In *British Machine Vision Conference*, pages 1–11. Citeseer, 2012.

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

[19] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5): 1147–1163, 2015.

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

[21] Carl Olsson, Anders P Eriksson, and Richard Hartley. Outlier removal using duality. In *IEEE Int. Conf. on Copmuter Vision and Pattern Recognition*, pages 1450–1457. IEEE Computer Society, 2010.

[22] Jaehyun Park and Stephen Boyd. General heuristics for nonconvex quadratically constrained quadratic programming. 2017.

[23] Pulak Purkait, Christopher Zach, and Anders Eriksson. Maximum consensus parameter estimation by reweighted l1 methods. In Marcello Pelillo and Edwin Hancock, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2018.

[24] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016.

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

[26] Hoi Sim Wong, Tat-Jun Chin, Jin Yu, and David Suter. Dynamic and hierarchical multi-structure geometric model fitting. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1044–1051. IEEE, 2011.

[27] Yinqiang Zheng, Shigeki Sugimoto, and Masatoshi Okutomi. Deterministically maximizing feasible subsystem for robust model fitting with unit norm constraint. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1825–1832. IEEE, 2011.

[28] Siyu Zhu, Runze Zhang, Lei Zhou, Tianwei Shen, Tian Fang, Ping Tan, and Long Quan. Very large-scale global sfm by distributed motion averaging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4568–4577, 2018.