Efficient algorithms for robust recovery of images from compressed data

Duc-Son Pham*, *Member, IEEE* and Svetha Venkatesh, *Senior Member, IEEE*

Abstract—Compressed sensing is an important theory for sub-Nyquist sampling and recovery of compressible data. Recently, it has been extended by Pham and Venkatesh [35] to cope with the case where corruption to the Cs data is modeled as impulsive noise. The new formulation, termed as robust CS, combines robust statistics and CS into a single framework to suppress outliers in the CS recovery. To solve the newly formulated robust CS problem, Pham and Venkatesh suggested a scheme that iteratively solve a number of CS problems, the solutions from which provably converge to the true robust compressed sensing solution. However, this scheme is rather inefficient as it has to use existing CS solvers as a proxy. To overcome limitation with the original robust CS algorithm, we propose in this paper more computationally efficient algorithms by following latest advances in large-scale convex optimization for non-smooth regularization. Furthermore, we also extend the robust CS formulation to various settings, including additional affine constraints, ℓ_1-norm loss function, mix-norm regularization, and multi-tasking, so as to further improve robust CS and derive simple but effective algorithms to solve these extensions. We demonstrate that the new algorithms provide much better computational advantage over the original robust compressed sensing method on the original robust CS formulation, and effectively solve more sophisticated extensions where the original methods simply cannot. We demonstrate the usefulness of the extensions on several CS imaging tasks.

Index Terms—robust compressed sensing, optimization algorithms, ADMM, FISTA, IRLS, ℓ_1 regularization

I. INTRODUCTION

Compressed sensing (CS) [7], [14] is a powerful sub-Nyquist sampling theory for the acquisition and recovery of sparse signals, that has received special attention in signal and image processing as well as other related fields such as statistics and computer science. The CS theory states that if the unknown signal is inherently sparse then it is possible to acquire and reconstruct signal (by solving a convex optimization problem) with a much lower number of measurements that would be otherwise needed using the existing Nyquist sampling scheme. In image processing, the CS theory is particularly relevant in several applications, such as magnetic resonant imaging (MRI) [29] or hyper-spectral imaging [11], [18], where acquisition time or sensing hardware cost plays a significant role. Also, the sparsity assumption typically holds due to, for example inherent wavelet structure in images [37].

In recent years, the CS literature has seen seen significant advances in both theory [3], [8], [15], [24] and applications

[17], [21], [23], [32], [36], [38], [42] (many of which are collected in the CS repository[1]). There are also a variety of specialized solvers for the CS recovery problem, which are developed from different angles, such as pursuit algorithms [13], [31], [33], optimization algorithms [19], [28], a complexity regularization algorithm [22], and Bayesian methods [27].

In this work, we focus on a particular aspect of CS recovery wherein the emphasis is on the robustness. This is originally raised by Pham and Venkatesh [35] who recognize that whilst existing CS recovery schemes may still be stable when the corruption of CS measurements is modeled as impulsive noise, they can be statistically inefficient. Such impulsive corruption can occur due to bit errors in transmission, malfunctioning pixels, faulty memory locations [10], and buffer overflow [20], and has been raised in many image processing works [2], [12], [40]. To address this problem, Pham and Venkatesh [35] have proposed a new formulation, known as robust CS, which combines traditional robust statistics [25] and existing CS into a single framework to effectively suppress outliers in the recovery. Whilst the focus of [35] is on the theoretical justification of the new formulation, they also suggest a provably convergent algorithm to solve their robust CS formulation. This majorization minimization (MM) algorithm finds the robust CS solution by iteratively solving a number of CS problems, which converge to the true solution. However, this is not computationally efficient because each iteration involves a full CS recovery, which is always iterative in nature.

To overcome the computational limitation of the original robust CS algorithm proposed in [35], we propose two new algorithms that iteratively majorize the original robust CS objective function, but they only have one main loop. One algorithm is adapted from the fast iterative shrinkage thresholding (FISTA) framework developed by Beck and Teboulle [4], which shares the same spirit as an unpublished work of Nesterov [34]. The other algorithm is based on a framework known as alternative direction method of multipliers (ADMM) [5]. Even though the original FISTA scheme was derived for the CS problem, it can be used for robust CS. Our contribution is a theoretical result that allows one to computes the Lipchitz constant for the application of FISTA. Additionally, we also derive a generalized ADMM algorithm for solving the robust CS formulation efficiently, which differs from the FISTA algorithm in that operator splitting and approximation updates are used. This results in a method that has same update complexity as FISTA, but is more flexible to extend.

Furthermore, we also extend robust CS in a number of

directions, including additional affine constraints, ℓ_1-norm loss function, mixed-norm regularization, and multi-tasking. We show that the ADMM is a powerful optimization framework for the robust CS problem as it can be modified or generalized to cope with these extensions where often other CS techniques, including FISTA, find impossible to do so. We show that the derived algorithms are simple to implement, provably convergent under the ADMM theory, and that effectively solve complex robust CS formulations.

The organization of the paper is as follows. Section II gives some background on robust CS, whilst Section III describes the FISTA and ADMM algorithms for solving robust CS formulation. Section IV presents four extensions of the robust CS formulation and derive computationally efficient algorithms for solving them. Section V contains numerical experiments to demonstrate the computational efficiency of the proposed algorithms. Finally, Section VI concludes the paper.

All Matlab code to implement our methods described in this paper and reproduce our results are readily available at the following website http://www.computing.edu.au/~dsp/code.php.

II. BACKGROUND

In compressed sensing (CS), one is interested in the recovery of a sparse signal $\mathbf{x} \in \mathbb{R}^N$ though the compressed measurement

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} + \mathbf{n}. \tag{1}$$

Here, $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ is the CS matrix that represents the compressive sampling operation and $\mathbf{n}$ is additive noise. The CS matrix is required to some stable embedding conditions for stable recovery [6]. As $M < N$ in the CS setting, the recovery of $\mathbf{x}$ from $\mathbf{y}$ is generally ill-posed. The CS theory has established that under an assumption that $\mathbf{x}$ is sparse, it is possible to recover $\mathbf{x}$ reliably from $\mathbf{y}$ with an error upper bounded by the noise strength. Among various approaches to solve the CS recovery problem, the optimization formulation often provides the best achievability for a given CS matrix

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x} \in \mathbb{R}^N} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1 \right\}. \tag{2}$$

In the normal CS setting, the noise in (1) is often considered Gaussian with bounded norm $\|\mathbf{n}\|_2 \leq \xi$ and thus the maximum error induced by a CS recovery is $\mathcal{O}(\|\mathbf{n}\|_2)$. However, Pham and Venkatesh [35] have discovered that when the noise is indeed impulsive, such a result will still hold for normal CS recovery but is rather inefficient. Thus, they propose a modification to the CS formulation, known as robust CS, to appropriately address the characteristics of the underlying additive noise. This is achieved by considering the robust loss function instead of the quadratic cost function in (2)

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x} \in \mathbb{R}^N} \left\{ g(\mathbf{x}) + \lambda\|\mathbf{x}\|_1 \right\}. \tag{3}$$

Here, $g(\mathbf{x}) = \sum_{i=1}^{M} \rho(y_i - (\mathbf{\Phi}\mathbf{x})_i)$ and $\rho(r)$ is the Huber's penalty function (soft limiter) given as follows

$$\rho(r) = \begin{cases} \frac{r^2}{2} & |r| \leq k\nu^2 \\ -\frac{k^2\nu^4}{2} + k\nu^2|r| & |r| > k\nu^2, \end{cases} \tag{4}$$

and its derivative is given by

$$\psi(r) = \rho'(r) = \begin{cases} r & |r| \leq k\nu^2 \\ k\nu^2 \mathrm{sgn}(r) & |r| > k\nu^2. \end{cases} \tag{5}$$

The parameter k of the Huber's penalty function is determined by the fraction of the outliers whilst the scale parameter ν is often estimated from some statistic of the median, such as the median of the absolute deviation (MAD). For detail, see [25].

As $\rho(r)$ is quadratic or linear depending on the actual value of r, solving (3) directly is hard. Pham and Venkatesh suggested that instead of solving (3), a better alternative is to solve a series of the normal CS problems. The idea is to replace $g(\mathbf{x})$ with an approximate quadratic function at every outer iteration with the general form

$$\begin{aligned} l^k(\mathbf{x}) &= (1/2)(\mathbf{v}^k - \mathbf{\Phi}\mathbf{x})^T\mathbf{W}(\mathbf{v}^k - \mathbf{\Phi}\mathbf{x}) + C &(6) \\ &= (1/2)\|\mathbf{W}^{1/2}\mathbf{v}^k - \mathbf{W}^{1/2}\mathbf{\Phi}\mathbf{x}\|_2^2 + C, &(7) \end{aligned}$$

where

$$C = g(\hat{\mathbf{x}}^k) - (1/2)\psi(\mathbf{y} - \mathbf{\Phi}\hat{\mathbf{x}}^k)^T\mathbf{W}^{-1}\psi(\mathbf{y} - \mathbf{\Phi}\hat{\mathbf{x}}^k), \tag{8}$$

$$\mathbf{v}^k = \mathbf{W}^{-1}\psi(\mathbf{y} - \mathbf{\Phi}\hat{\mathbf{x}}^k) + \mathbf{\Phi}\hat{\mathbf{x}}^k. \tag{9}$$

Pham and Venkatesh detailed two options for $\mathbf{W}$, which are commonly used in the robust statistics literature

- Modified residuals (MR): $\mathbf{W} = \mu\mathbf{I}$
- Iteratively reweighted: $w_{ii} = \psi(r_i^k)/r_i^k$, $w_{ij=0}, i \neq j$.

When using $l^k(\mathbf{x})$ as shown in (7) for $g(\mathbf{x})$ in (3), the resultant problem is essentially a normal CS problem and thus considered solved.

Whilst the above strategy will work, it is inefficient because each outer iteration involves a full CS problem and it is known that the CS problem needs to be solved iteratively as well. The double loops are therefore the main computational deficiency of the above strategy. To address this limitation, we consider bypassing the inner CS step and thus there will be only one loop for the overall algorithm. There are two powerful optimization frameworks that are suitable for this purpose which we describe next.

III. PROPOSED ALGORITHMS

A. FISTA Algorithm

Fast iterative shrinkage thresholding (FISTA) is an optimization approach that effectively decouples the variables from the smooth loss function in the compressed sensing objective. This approach was proposed by [4], which also shares the same philosophy as an unpublished work of [34]. Technically, FISTA is a variant of majorization minimization (MM) algorithms [26] and has a special choice for the quadratic majorization as well updates that involve historical points.

Consider minimizing a convex optimization of the form $\arg\min_{\mathbf{x}} f(\mathbf{x})$ where

$$f(\mathbf{x}) = g(\mathbf{x}) + R(\mathbf{x}). \tag{10}$$

Here, $g(\mathbf{x})$ is a smooth loss function but the variables in this loss function are coupled. The core idea of FISTA is to consider a quadratic majorization of $g(\mathbf{x})$, denotes as

$h(\mathbf{x})$, such that it effectively decouples the variables. If such decoupling is possible, the approximate problem is then easier to solve even when the regularization term $R(\mathbf{x})$ is possibly non-smooth (such as $\|\mathbf{x}\|_1$), because it can be decomposed into a number of univariate optimization problems whose solution is analytical.

The first trick of FISTA is to decouple the variables by considering the majorization at iteration k and approximation point $\mathbf{z}^k$

$$h(\mathbf{x}; \mathbf{z}^k) = g(\mathbf{z}^k) + \nabla g(\mathbf{z}^k)^T (\mathbf{x} - \mathbf{z}^k) + \frac{L}{2}\|\mathbf{x} - \mathbf{z}^k\|_2^2. \quad (11)$$

Here, $\mathbf{z}^k$ is used as the approximation point rather than $\mathbf{x}^k$ as it involves historical updates of $\mathbf{x}^k$ by a careful choice, which is subsequently show in (17). Also, L is the Lipchitz constant of the gradient of the loss function $g(\mathbf{x})$ to ensure that $h(\mathbf{x})$ is a proper majorization of $g(\mathbf{x})$. Thus, at iteration k, FISTA finds $\mathbf{x}^k$ via

$$\mathbf{x}^k = \arg\min_{\mathbf{x}} \left\{ \frac{L}{2}\|\mathbf{x} - \mathbf{v}^k\|_2^2 + R(\mathbf{x}) \right\} \quad (12)$$

where $\mathbf{v}^k = \mathbf{z}^k - (1/L)\nabla g(\mathbf{z}^k)$. For the quadratic loss function $g(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi x}\|_2^2$, it can be shown that $L = 2\lambda_{\max}(\mathbf{\Phi}^T\mathbf{\Phi})$, and $\mathbf{v} = \mathbf{z}^k - (1/L)(\mathbf{\Phi}^T\mathbf{\Phi z}^k - \mathbf{y})$. For the ℓ_1-norm regularization as in the case of CS, this results in

$$\mathbf{x}^k = \arg\min_{\mathbf{x}} \left\{ \frac{L}{2}\|\mathbf{x} - \mathbf{v}\|_2^2 + \lambda\|\mathbf{x}\|_1 \right\}. \quad (13)$$

This problem can be solved element-wise and its solution is

$$\mathbf{x}^k = \mathsf{S}_{\lambda/L}(\mathbf{v}), \quad (14)$$

where the soft-thresholding shrinkage operator is defined as

$$\mathsf{S}_\tau(\mathbf{x}) = \{\mathbf{t} : t_i = \text{sign}(x_i)\max(|x_i| - \tau, 0)\}. \quad (15)$$

The second trick of FISTA is to use a clever update of the approximation point to speed up convergence

$$t^{k+1} = \frac{1 + \sqrt{1 + 4(t^k)^2}}{2} \quad (16)$$

$$\mathbf{z}^{k+1} = \mathbf{x}^k + \left(\frac{t^k - 1}{t^{k+1}} \right)(\mathbf{x}^k - \mathbf{x}^{k-1}). \quad (17)$$

The original FISTA framework can be readily used for robust CS case if $\mathbf{v}^k$ and the Lipchitz constant can be computed for the robust loss function. In case of $\mathbf{v}^k$, it can be easily seen that

$$\mathbf{v}^k = \mathbf{z}^k - \frac{1}{L}\mathbf{\Phi}^T\psi(\mathbf{\Phi z}^k - \mathbf{y}). \quad (18)$$

It remains to compute the Lipchitz constant. To do so, we rely on the following result:

Lemma 1: Let $f(x)$ be a smooth convex function on $\mathcal{X}$ and suppose that the domain $\mathcal{X}$ is divided into two regions $\mathcal{X}_1$ and $\mathcal{X}_2$ such that $\mathcal{X}_1 \cup \mathcal{X}_2 = \mathcal{X}$, and that $g(x) = h(x)$ for $x \in \mathcal{X}_1 \cap \mathcal{X}_2$. Denote as L_g and L_h the Lipchitz constants of g and h respectively on the domains $\mathcal{X}_1$ and $\mathcal{X}_2$. Then the Lipchitz constant of f is bounded by

$$L_f \leq \{L_g + L_h\}.$$

The proof of this Lemma is detailed in the Appendix. The result implies that for mixed function like the robust CS cost functions being considered, we just take the sum of Lipchitz constants over each continuous and bounded domain. The Lipchitz constant for the quadratic part is as before, i.e. $L_1 = 2\lambda_{\max}(\mathbf{\Phi}^T\mathbf{\Phi})$, whilst for the linear part we can split into negative and positive domain. In both cases, the Lipchitz constant is zero due to the fact that ψ is a constant. Thus, the Lipchitz constant for the robust CS cost function is still $2\lambda_{\max}(\mathbf{\Phi}^T\mathbf{\Phi})$.

B. ADMM Algorithm

Alternating direction method of multipliers (ADMM) is a simple but powerful framework in optimization, which is suited for today's large-scale problems arising in machine learning and signal processing. The method was in fact developed a long ago before advanced computing power was available, and re-discovered many times under different perspectives. Recently, [5] has unified the framework in a simple and concise explanation. In either the CS or robust CS problem, the main technical challenge is that the variables are coupled through $\mathbf{\Phi}$ in either the quadratic or robust loss function. This makes it rather difficult when the extra constraint with non-smooth ℓ_1 norm is introduced. In principle, the problem is easier to tackle if the variables can be decoupled, so that the problem can be solved element-wise or group-wise. Using a clever trick, known as operator splitting [16], the ADMM framework suggests to separate the regularization term from the smooth term by introducing an additional variable $\mathbf{z}$, which is tied to the original variable via an affine constraint:

$$\min_{\mathbf{x},\mathbf{z}} \quad g(\mathbf{x}) + \|\mathbf{z}\|_1 \quad \text{s.t} \quad \mathbf{x} - \mathbf{z} = 0. \quad (19)$$

Here, $g(\mathbf{x})$ is the robust CS loss function. For this type of regularized objective function, ADMM considers the following augmented Lagrangian

$$\mathcal{L}(\mathbf{x}, \mathbf{z}, \mathbf{y}) = g(\mathbf{x}) + \lambda\|\mathbf{z}\|_1 + \mathbf{w}^T(\mathbf{x} - \mathbf{z}) + \frac{\eta}{2}\|\mathbf{x} - \mathbf{z}\|_2^2. \quad (20)$$

Here, η is the parameter associated with the augmentation $\frac{\eta}{2}\|\mathbf{x} - \mathbf{z}\|_2^2$, and this is to improve the numerical stability of the algorithm. The strategy for minimizing this augmented Lagrangian is iterative updating of the primal and dual variables. With a further normalization on the dual variable $\mathbf{u} = (1/\eta)\mathbf{w}$, it is shown [5] that as far as the primal and dual variables $\mathbf{x}$ and $\mathbf{z}$ are concerned

$$\mathcal{L}(\mathbf{x}, \mathbf{z}; \mathbf{u}) = g(\mathbf{x}) + \lambda\|\mathbf{z}\|_1 + \frac{\eta}{2}\|\mathbf{x} - \mathbf{z} + \mathbf{u}\|_2^2 + \text{const.} \quad (21)$$

where the constant is independent of $\mathbf{x}$ and $\mathbf{z}$ (actually const $= -\eta\|\mathbf{u}\|_2^2/2$). Note of the semi-colon, which treats $\mathbf{u}$ as a parameter rather than a variable when solving for other variables. Thus, the optimality point of the Lagrangian can be found by iteratively updating the variables as follows:

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \left\{ g(\mathbf{x}) + \frac{\eta}{2}\|\mathbf{x} - \mathbf{z}^k + \mathbf{u}^k\|_2^2 \right\} \quad (22)$$

$$\mathbf{z}^{k+1} = \arg\min_{\mathbf{z}} \left\{ \lambda\|\mathbf{z}\|_1 + \frac{\eta}{2}\|\mathbf{x}^{k+1} - \mathbf{z} + \mathbf{u}^k\|_2^2 \right\} \quad (23)$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \mathbf{x}^{k+1} - \mathbf{z}^{k+1}. \quad (24)$$

We note that the update steps for $\mathbf{u}$ and $\mathbf{z}$ are straightforward. In particular, for $\mathbf{z}$ it is known that it is a soft-thresholding shrinkage operation

$$\mathbf{z}^{k+1} = \mathsf{S}_{\lambda/\eta}(\mathbf{x}^{k+1} + \mathbf{u}^k). \qquad (25)$$

Due to the nature of $g(\mathbf{x})$, there is no exact solution for (22), and finding it always necessitates iterative algorithms. This will increase computational burden to the overall algorithm in a similar way as the previous robust CS algorithms introduced in [35]. To alleviate the computational problem, we propose to follow a novel framework, known as generalized ADMM and developed by Eckstein and Bertsekas [16]. In generalized ADMM, the update steps can be solved *approximately* as long as the differences between the exact and approximate solutions generate a summable sequence. When such a condition is satisfied, the generalized ADMM theory has proved that the algorithm will converge to the solution [16, Theorem 8].

To utilize the generalized ADMM theory, once again we adapt an MM algorithm to solve (22), which is in the same spirit as the original robust CS [35]. In essence, this replaces $g(\mathbf{x})$ with a suitable quadratic majorization as discussed previously. The major difference is that we only perform the minimization of the majorization *once*, as opposed to iteratively as in [35]. Specifically, we propose to modify the update step for $\mathbf{x}$ in (22) by using the quadratic approximation of $g(\mathbf{x})$ at iteration k as $l^k(\mathbf{x})$ (shown in (7))

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}}(1/2)\|\mathbf{W}^{1/2}\mathbf{v}^k - \mathbf{W}^{1/2}\boldsymbol{\Phi}\mathbf{x}\|_2^2$$
$$+\frac{\eta}{2}\|\mathbf{x} - \mathbf{z}^k + \mathbf{u}^k\|_2^2. \qquad (26)$$

It can be easily recognized that the solution of this problem is exact

$$\mathbf{x}^{k+1} = (\boldsymbol{\Phi}^T\mathbf{W}\boldsymbol{\Phi}^T + \eta\mathbf{I})^{-1}(\boldsymbol{\Phi}^T\mathbf{W}\mathbf{v}^k + \eta(\mathbf{z}^k - \mathbf{u}^k)). \quad (27)$$

We note that the quadratic approximation of FISTA can also be used. However, the choice above leads to a better approximation and hence will converge to the true solution faster. It is also easily seen that for the MR choice of the quadratic approximation where $\mathbf{W} = \mu\mathbf{I}$, the matrix under inversion in (28) is fixed

$$\mathbf{x}^{k+1} = (\mu\boldsymbol{\Phi}^T\boldsymbol{\Phi}^T + \eta\mathbf{I})^{-1}(\mu\boldsymbol{\Phi}^T\mathbf{v}^k + \eta(\mathbf{z}^k - \mathbf{u}^k)). \quad (28)$$

Hence, the inversion $(\mu\boldsymbol{\Phi}^T\boldsymbol{\Phi}^T + \eta\mathbf{I})^{-1}$ can be computed once and cached so that the update step in subsequent iterations can be fast.

The generalized ADMM for the specific case being considered can be stated as follows:

Theorem 1: Consider an ADMM algorithm that solves the convex problem (3) via the updates (28), (23), (24). Denote as $\mathbf{x}_*^{k+1}$ the exact solution of (22), and as $\mathbf{x}^{k+1}$ the approximate of (22) via (28). If the sequence $\{\mu_{k+1} : \mu_{k+1} = \|\mathbf{x}^{k+1} - \mathbf{x}_*^{k+1}\|_2\}$ is summable, i.e., $\sum_{k=1}^{\infty}\mu_k \leq \infty$, then the above updates will generate a sequence $\{\mathbf{x}^{k+1}\}$ that converge to the true solution of (3).

Next, we discuss the convergence aspect of the proposed generalized ADMM algorithm. When the update steps are solved exactly, the existing ADMM theory [5] states that the

penalty parameter η affects both the primal residual (defined as $\mathbf{s}^{k+1} = \eta(\mathbf{z}^{k+1} - \mathbf{z}^k)$), and the primal residual (defined as $\mathbf{r}^{k+1} = \mathbf{x}^{k+1} - \mathbf{z}^{k+1}$) in an opposite manner: a large η tends to generate a small primal residual and a large dual residual and vice versa. Thus, selecting the optimal penalty parameter is typically a trade-off between primal and residual residuals with an ADMM algorithm, and $\eta = 1$ generally works for most cases. However, more emphasis should be made to the primal residual in the case of the proposed generalized ADMM algorithm because the update step of the primal variable $\mathbf{x}$ is not solved exactly. This will ensure that the approximation error in the primal variable is promptly compensated by the dual update, at the small sacrifice in convergence rate due to the residual error being slightly larger. Intensive numerical studies suggest that a value for η of between 2 and 5 for η works rather well in many cases. We shall examine this in more detail in the experimental section, where we use $\eta = 2$.

IV. BEYOND ROBUST CS

The FISTA and ADMM algorithms for robust CS presented tackle the optimization from slightly different angles. Whilst FISTA solves the problem by replacing the robust cost function with a simpler quadratic approximation that decouples the variables, the ADMM decouples the ℓ_1 regularization norm via operator splitting. Whilst FISTA has only one approximation, ADMM involves operator splitting *and* quadratic approximation at the step that updates $\mathbf{x}$. Thus it appears that FISTA may have a convergence advantageous due to being simpler and having less tuning requirements. However, numerical experience indicates that for a given tolerance, the ADMM algorithm is actually faster than FISTA in terms of both number of iterations or computational time to reach a given tolerance. This will be illustrated further in the experiments.

The advantage of ADMM is better realized when one needs to extend robust CS in similar ways as many extensions on the basic CS have been made in the literature. This is difficult, if not impossible, with the FISTA scheme. Next, we discuss several possible extensions that can be simply achieved with the proposed ADMM algorithm. We shall omit the arguments of the Lagrangian, which are obvious in the context, for notational simplicity and space reduction.

A. Additional Affine Constraints

In some cases, one would like to impose additional affine constraints on the optimization problem $\mathbf{c}^T\mathbf{x} = 1$. This could be of prior knowledge on the power modeling and this could potentially improve stabilization of the CS solution. Thus, the Lagrangian (20) could be altered as follows

$$\mathcal{L} = g(\mathbf{x}) + \lambda\|\mathbf{z}\|_1 + \mathbf{w}_1^T(\mathbf{x} - \mathbf{z}) + \frac{\eta_1}{2}\|\mathbf{x} - \mathbf{z}\|_2^2$$
$$+w_2(\mathbf{c}^T\mathbf{x} - 1) + \frac{\eta_2}{2}\|\mathbf{c}^T\mathbf{x} - 1\|_2^2. \qquad (29)$$

Here, $\mathbf{w}_1$ and w_2 are the dual variables for the equality constraints. Again, by scaling the dual variables $\mathbf{u}_1 = \mathbf{w}_1/\eta_1$ and $u_2 = w_2/\eta_2$ we obtain

$$\mathcal{L} = g(\mathbf{x}) + \lambda\|\mathbf{z}\|_1 + \frac{\eta_1}{2}\|\mathbf{x} - \mathbf{z} + \mathbf{u}_1\|_2^2$$
$$+\frac{\eta_2}{2}\|\mathbf{c}^T\mathbf{x} - 1 + u_2\|_2^2 + \text{const.} \qquad (30)$$

Thus, the ADMM update step for $\mathbf{x}$ is the solution of

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \left\{ g(\mathbf{x}) + \frac{\eta_1}{2}\|\mathbf{x} - \mathbf{z}^k + \mathbf{u}_1^k\|_2^2 \right.$$
$$\left. + \frac{\eta_2}{2}\|\mathbf{c}^T\mathbf{x} - 1 + u_2^k\|_2^2 \right\}. \quad (31)$$

Once again, if this step is to be solved approximately using a quadratic majorization with $\mathbf{W} = \mu\mathbf{I}$ as discussed previously then it can be shown that

$$\mathbf{x}^{k+1} = \mathbf{H}(\mu\mathbf{\Phi}^T\mathbf{v}^k + \eta_1(\mathbf{z}^k - \mathbf{u}^k) + \eta_2(1 - u_2^k)\mathbf{c}), \quad (32)$$

where $\mathbf{H} = (\mu\mathbf{\Phi}^T\mathbf{W}\mathbf{\Phi} + \eta_1\mathbf{I} + \eta_2\mathbf{c}\mathbf{c}^T)^{-1}$.

It can be shown that the updates step for $\mathbf{z}$ remains the same as (25) except that $\mathbf{u}$ and η are replaced with $\mathbf{u}_1$ and η_1 respectively. Finally, the updates of the dual variables are

$$\mathbf{u}_1^{k+1} = \mathbf{u}_1^k + \mathbf{x}^{k+1} - \mathbf{z}^{k+1}, \quad (33)$$
$$u_2^{k+1} = u_2^k + \mathbf{c}^T\mathbf{x}^{k+1} - 1. \quad (34)$$

Just like the basic ADMM algorithm, convergence is determined when both the primal and dual residuals are sufficiently small. Whilst the dual residual is as before, i.e., $\mathbf{s} = \eta_1(\mathbf{z}^{k+1} - \mathbf{z}^k)$, there are effectively two residual vectors $\mathbf{r}_1^k = \mathbf{x}^k - \mathbf{z}^k$ and $r_2^k = \mathbf{c}^T\mathbf{x}^k - 1$. Depending on the desired accuracy requirement of a particular application, the stopping criterion can be determined accordingly (see [5, p.19]).

B. Mixed-Norm Regularization

In certain situations, one may wish to impose ℓ_2 regularization on the solution of the recovery. Such a motivation may arise from the fact that the absolute sparse model may not be realistic, and thus it is more desirable to consider

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x} \in \mathbb{R}^N} \left\{ g(\mathbf{x}) + \lambda\|\mathbf{x}\|_1 + \beta\|\mathbf{x}\|_2^2 \right\}. \quad (35)$$

In the case of quadratic loss function, i.e., $g(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2^2$, this is known as elastic-net [43]. Thus, the proposed formulation could be interpreted as a robust version of elastic-net. The robust CS formulation is treated a special case when $\beta = 0$.

For the original elastic-net, it is easily recognized that a simple algebra can convert it to a Lasso (or CS) form, and thus it can be solved with many efficient ℓ_1-regularization algorithms. For the proposed robust elastic-net, it is not possible because of the loss function $g(\mathbf{x})$ being not quadratic. However, it is trivial to show that it is possible to modify the FISTA and generalized ADMM algorithms discussed in the previous section to cater for this additional regularization term. Indeed, this regularization term only affects the update step of $\mathbf{x}$. In both FISTA and generalized ADMM, the majorization is a quadratic function and thus absorbing this extra quadratic term is straightforward. For example, in the case of the FISTA algorithm, we need to solve (c.f. (13))

$$\min_{\mathbf{x}} (1/2)\|\mathbf{v} - \mathbf{x}\|_2^2 + (\lambda/L)\|\mathbf{x}\|_1 + (\beta/L)\|\mathbf{x}\|_2^2, \quad (36)$$

which is equivalent to

$$\min_{\mathbf{x}} \frac{1}{2}\left\|\frac{\mathbf{v}}{1 + (\beta/L)} - \mathbf{x}\right\|_2^2 + \frac{\lambda}{L + \beta}\|\mathbf{x}\|_1, \quad (37)$$

which is of the same form and this induces the soft-thresholding shrinkage operation. Likewise, in the case of the generalized ADMM algorithm, we need to to solve (c.f (26))

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} (1/2)\|\mathbf{W}^{1/2}\mathbf{v}^k - \mathbf{W}^{1/2}\mathbf{\Phi}\mathbf{x}\|_2^2$$
$$+ \frac{\eta}{2}\|\mathbf{x} - \mathbf{z}^k + \mathbf{u}^k\|_2^2 + \|\beta\|_2^2, \quad (38)$$

which has a slight modification compared with (28)

$$\mathbf{x}^{k+1} = (\mu\mathbf{\Phi}^T\mathbf{\Phi}^T + (\eta + \beta)\mathbf{I})^{-1}(\mu\mathbf{\Phi}^T\mathbf{v}^k + \eta(\mathbf{z}^k - \mathbf{u}^k)). \quad (39)$$

Thus, extension to mixed-norm regularization is straightforward of the proposed ADMM algorithm.

C. ℓ_1 Loss Function

In the original robust CS paper [35], the Huber loss is selected. This is suitable for impulsive noise being modeled as a contaminated mixture [25]. However, the robust CS framework is not necessarily restricted to the Huber loss function and indeed many loss functions in the robust statistics can be used to cater for different noise types. One particular interest is the ℓ_1-norm loss function, which is optimal when the impulsive noise is modeled as a Cauchy distribution [25]. In this case, it is desirable to solve

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x} \in \mathbb{R}^N} \left\{ \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_1 + \lambda\|\mathbf{x}\|_1 \right\}. \quad (40)$$

It has come to our attention that a specific case where $\lambda = 1$ is proposed in [39] where a hybrid algorithm using FISTA and alternative Lagrangian multiplier (ALM) algorithm is derived. In what follows, we propose an ADMM algorithm inspired by the recent framework [5], [41].

We note that the FISTA algorithm is not easily derived, because the loss function is not differentiable. To overcome the difficulty associated with two parts of the objective function that are both non-differentiable, we propose to apply the operator splitting mechanism of the ADMM framework twice. Specifically, we introduce two additional variables $\mathbf{v}$ and $\mathbf{z}$ and rewrite the formulation as

$$\arg\min_{\mathbf{x}, \mathbf{v}, \mathbf{z}} \quad \|\mathbf{v}\|_1 + \lambda\|\mathbf{z}\|_1$$
$$\text{s.t.} \quad \mathbf{\Phi}\mathbf{x} - \mathbf{v} - \mathbf{y} = 0, \quad \mathbf{x} - \mathbf{z} = 0. \quad (41)$$

Thus, the augmented Lagrangian is

$$\mathcal{L} = \|\mathbf{v}\|_1 + \lambda\|\mathbf{z}\|_1 + \mathbf{w}_1^T(\mathbf{\Phi}\mathbf{x} - \mathbf{v} - \mathbf{y}) + \frac{\eta_1}{2}\|\mathbf{\Phi}\mathbf{x} - \mathbf{v} - \mathbf{y}\|_2^2$$
$$+ \mathbf{w}_2^T(\mathbf{x} - \mathbf{z}) + \frac{\eta_2}{2}\|\mathbf{x} - \mathbf{z}\|_2^2. \quad (42)$$

With the scaled dual variables $\mathbf{u}_1 = \mathbf{w}_1/\eta_1$ and $\mathbf{u}_2 = \mathbf{w}_2/\eta_2$

$$\mathcal{L} = \|\mathbf{v}\|_1 + \lambda\|\mathbf{z}\|_1 + \frac{\eta_1}{2}\|\mathbf{\Phi}\mathbf{x} - \mathbf{v} - \mathbf{y} + \mathbf{u}_1\|_2^2$$
$$+ \frac{\eta_2}{2}\|\mathbf{x} - \mathbf{z} + \mathbf{u}_2\|_2^2 + \text{const}. \quad (43)$$

With this form, the updates for the variables are easily computed under the ADMM principle. For $\mathbf{x}$, the update solves

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \frac{\eta_1}{2}\|\mathbf{\Phi}\mathbf{x} - \mathbf{v}^k - \mathbf{y} + \mathbf{u}_1^k\|_2^2$$
$$+ \frac{\eta_2}{2}\|\mathbf{x} - \mathbf{z}^k + \mathbf{u}_2^k\|_2^2 \quad (44)$$

which yields the exact solution

$$
\begin{aligned}
\mathbf{x}^{k+1} = & (\eta_1 \mathbf{\Phi}^T \mathbf{\Phi} + \eta_2 \mathbf{I})^{-1}(\eta_1 \mathbf{\Phi}^T(\mathbf{v}^k + \mathbf{y} - \mathbf{u}_1^k) \\
& + \eta_2(\mathbf{z}^k - \mathbf{u}_2^k)).
\end{aligned} \quad (45)
$$

For both $\mathbf{v}$ and $\mathbf{z}$, the update steps are simple soft-thresholding operations. For $\mathbf{v}$, the update step solves

$$
\mathbf{v}^{k+1} = \arg\min_{\mathbf{v}} \|\mathbf{v}\|_1 + \frac{\eta_1}{2}\|\mathbf{t}^k - \mathbf{v}\|_2^2, \quad (46)
$$

where $\mathbf{t}^k = \mathbf{\Phi}\mathbf{x}^{k+1} - \mathbf{y} + \mathbf{u}_1^k$. Likewise, for $\mathbf{z}$ the update step solves

$$
\mathbf{z}^{k+1} = \arg\min_{\mathbf{z}} \lambda\|\mathbf{z}\|_1 + \frac{\eta_2}{2}\|\mathbf{x}^{k+1} + \mathbf{u}_2^k - \mathbf{z}\|_2^2. \quad (47)
$$

They both have a similar form as (23), and thus from (25) we deduce (c.f. (15))

$$
\mathbf{v}^{k+1} = \mathsf{S}_{1/\eta_1}(\mathbf{t}^k), \quad (48)
$$

$$
\mathbf{z}^{k+1} = \mathsf{S}_{\lambda/\eta_2}(\mathbf{x}^{k+1} + \mathbf{u}_2^k), \quad (49)
$$

as the updates for $\mathbf{v}$ and $\mathbf{z}$. Finally, the dual updates are

$$
\mathbf{u}_1^{k+1} = \mathbf{u}_1^k + \mathbf{\Phi}\mathbf{x}^{k+1} - \mathbf{v}^{k+1} - \mathbf{y} \quad (50)
$$

$$
\mathbf{u}_2^{k+1} = \mathbf{u}_2^k + \mathbf{x}^{k+1} - \mathbf{z}^{k+1}. \quad (51)
$$

The stopping criterion is when the residual vectors are sufficiently small, including $\mathbf{s}_1^k = \eta_1(\mathbf{v}^{k+1} - \mathbf{v}^k)$, $\mathbf{s}_2^k = \eta_2(\mathbf{z}^{k+1} - \mathbf{z}^k)$, $\mathbf{r}_1^k = \mathbf{x}^k - \mathbf{z}^k$, and $\mathbf{r}_2^k = \mathbf{\Phi}\mathbf{x}^k - \mathbf{y} - \mathbf{v}^k$.

D. Multi-Task Setting

The recent literature on CS also reveals that the basic sparsity recovery scheme can be improved if one exploits further domain knowledge. Such an exploitation could be base on the constraint of the sparsity models. Extensions, such as model-based CS [3] and group sparsity [24], are key examples of the exploitation that can effectively reduce the CS requirements for a comparable recovery error when compared with conventional CS. Here, we focus on a slight variation where there are multiple CS tasks are to be performed: there are multiple CS measurements $\mathbf{y}_i, i = 1, \dots, L$, each follows the model $\mathbf{y}_i = \mathbf{\Phi}\mathbf{x}_i + \mathbf{n}_i$.

In the image processing context, this could arise in, for example, compressed sensing of multiple video images. In these circumstances, there many be similarities between images. For example, moving images likely consist of relatively same large background and small moving objects. Thus, the sparse representation of these original images may have similar sparse coefficients representing the common background part (see Fig. 9 for an illustration of a sequence of random bars images used later in the experiment). For that reason, it follows from the existing results on advanced CS [24] that exploiting the shared structure between tasks is likely to improve CS recovery compared to the case where the tasks are performed independently.

Denote as $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_L]$ the collection of sparse vectors to be recovered from the tasks, and $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_L]$ the collection of CS measurements. Extending the single-task robust CS, the multi-task robust CS can be formulated as follows

$$
\hat{\mathbf{x}} = \arg\min_{\mathbf{X} \in \mathbb{R}^{N \times L}} \left\{ g(\mathbf{X}) + \lambda\|\mathbf{X}^T\|_{\ell_2/\ell_1} \right\}. \quad (52)
$$

Here, $g(\mathbf{X}) = \sum_{i=1}^{L} \rho(\mathbf{y}_i - \mathbf{\Phi}\mathbf{x}_i)$ and $\|\mathbf{A}\|_{\ell_2/\ell_1} = \sum_i \|\mathbf{a}_i\|_2$ where $\mathbf{a}_i$'s denote the columns of $\mathbf{A}$. Clearly the loss term is the same, whilst for the regularization terms, we seek sparsity along the columns of $\mathbf{X}$ but denseness along the rows of $\mathbf{X}$. This clearly reflects the prior assumption that sparse coefficients of the common parts are likely to be similar, hence the corresponding rows of $\mathbf{X}$ should be dense, whilst it is sparse column-wise to respect the single-task CS's assumption. When $g(\mathbf{X})$ is a quadratic loss function, this is a special matrix formulation of group Lasso in the statistics literature [1], [30].

We now show that it is possible to extend both the FISTA and generalized ADMM algorithms to cater for this formulation. Before doing so, we present a generalization of the soft-thresholding shrinkage operation as follows:

Lemma 2: The optimization problem

$$
\arg\min_{\mathbf{z}} \left\{ \lambda\|\mathbf{z}\|_2 + \frac{\eta}{2}\|\mathbf{v} - \mathbf{z}\|_2^2 \right\}. \quad (53)
$$

has the solution $\mathbf{z} = \mathbf{v}\max(\|\mathbf{v}\|_2 - \lambda/\eta, 0)/\|\mathbf{v}\|_2$

This result can be proved by simple geometrical arguments. Indeed, denote $\mathbf{z}^*$ as the solution of (53), then we consider all points $\mathbf{z}$ such that $\|\mathbf{v} - \mathbf{z}\|_2 = \|\mathbf{v} - \mathbf{z}^*\|_2 = R$. It turns out that these points are lying on the ball with center at $\mathbf{v}$ and radius R. Among these points, only the point that satisfies $\mathbf{z} = \alpha\mathbf{v}$, i.e., intersection of the ball and the vector $\mathbf{v}$, will have minimum ℓ_2 norm, which minimizes the second term in (53). Substituting this into (53) yields the form of the soft-thresholding shrinkage problem, for which the result is obtained after simple manipulations.

1) FISTA algorithm.: Generalizing (13) for the multi-task settings, denote as $\mathbf{V}^k = [\mathbf{v}_1^k, \dots, \mathbf{v}_L^k]$, where $\mathbf{v}_i^k = \mathbf{z}_i^k - \frac{1}{L}\mathbf{\Phi}^T\psi(\mathbf{\Phi}\mathbf{z}_i^k - \mathbf{y}_i)$. Then, the update step for $\mathbf{X}$ solves

$$
\arg\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{V} - \mathbf{X}\|_F^2 + \lambda\|\mathbf{X}^T\|_{\ell_2/\ell_1}. \quad (54)
$$

This problem can be written row-wise in the form of (53) and thus the solution is exact. Meanwhile, the update step for $\mathbf{Z}$ is also similar

$$
\mathbf{Z}^{k+1} = \mathbf{X}^k + \frac{t_k - 1}{t_{k+1}}(\mathbf{X}^k - \mathbf{X}^{k-1}), \quad (55)
$$

with $t^{k+1} = (1 + \sqrt{1 + 4(t^k)^2})/2$.

2) ADMM algorithm.: We rewrite the Lagrangian for the current setting as follows

$$
\mathcal{L}(\mathbf{X}, \mathbf{Z}; \mathbf{U}) = g(\mathbf{X}) + \lambda\|\mathbf{Z}^T\|_{\ell_2/\ell_1} + \frac{\eta}{2}\|\mathbf{X} - \mathbf{Z} + \mathbf{U}\|_F^2 \\
+ \text{const.} \quad (56)
$$

Thus, the ADMM update steps are

$$
\mathbf{X}^{k+1} = \arg\min_{\mathbf{X}} \left\{ g(\mathbf{X}) + \frac{\eta}{2}\|\mathbf{X} - \mathbf{Z}^k + \mathbf{U}^k\|_F^2 \right\} \quad (57)
$$

$$
\mathbf{Z}^{k+1} = \arg\min_{\mathbf{z}} \left\{ \lambda\|\mathbf{Z}^T\|_{\ell_2/\ell_1} + \frac{\eta}{2}\|\mathbf{X}^{k+1} + \mathbf{U}^k \\ - \mathbf{Z}\|_F^2 \right\} \quad (58)
$$

$$
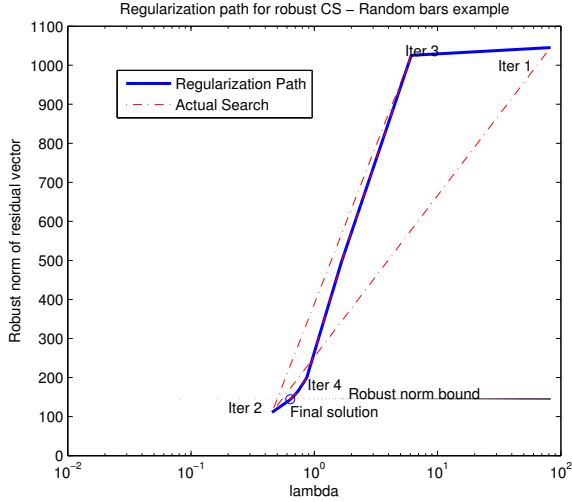\mathbf{U}^{k+1} = \mathbf{U}^k + \mathbf{X}^{k+1} - \mathbf{Z}^{k+1}. \quad (59)
$$

Fig. 1. Regularization path of robust CS for random bars example

Like FISTA, the update step of $\mathbf{Z}$ can easily be decomposed row-wise, each has the form of (53), and thus the solution for each row of $\mathbf{Z}^{k+1}$ can be obtained immediately. For the update step of $\mathbf{X}$, again we resort to the generalized ADMM principle. That is, we approximate $g(\mathbf{X})$ with a quadratic loss at $\mathbf{X}^k$

$$
\begin{aligned}
h(\mathbf{X}; \mathbf{X}^k) &= g(\mathbf{X}^k) + \nabla_{\mathbf{X}} g(\mathbf{X}^k)^T (\mathbf{X} - \mathbf{X}^k) \\
&\quad + \frac{\mu}{2} \|\mathbf{X} - \mathbf{X}^k\|_F^2. \quad (60)
\end{aligned}
$$

Thus, the generalized ADMM algorithm finds the update via

$$
\begin{aligned}
\mathbf{X}^{k+1} = \arg\min_{\mathbf{X}} \Big\{ &\nabla_{\mathbf{X}} g(\mathbf{X}^k)^T (\mathbf{X} - \mathbf{X}^k) + \frac{\mu}{2} \|\mathbf{X} - \mathbf{X}^k\|_F^2 \\
&+ \frac{\eta}{2} \|\mathbf{X} - \mathbf{Z}^k + \mathbf{U}^k\|_F^2 \Big\}, \quad (61)
\end{aligned}
$$

which yields the following solution

$$
\mathbf{X}^{k+1} = (\mu \mathbf{\Phi}^T \mathbf{\Phi} + \eta \mathbf{I})^{-1} (\mu \mathbf{\Phi}^T \mathbf{V}^k + \eta (\mathbf{Z}^k - \mathbf{U}^k)), \quad (62)
$$

where $\mathbf{V}^k = [\mathbf{v}_1^k, \ldots, \mathbf{v}_L^k]$, $\mathbf{v}_k = \frac{1}{\mu} \psi(\mathbf{y}_i - \mathbf{\Phi} \mathbf{x}_i^k) + \mathbf{\Phi} \mathbf{x}_i^k$.

The stopping criterion is when all primal and dual residual matrices are small, they include

$$
\begin{aligned}
\mathbf{S}^{k+1} &= \eta (\mathbf{Z}^{k+1} - \mathbf{Z}^k) \quad (63) \\
\mathbf{R}^{k+1} &= \mathbf{X}^{k+1} - \mathbf{Z}^{k+1}. \quad (64)
\end{aligned}
$$

Like the single-task case, one should set η sufficiently large to obtain a smooth decrease of the objective function.

E. Discussion

Further extensions. We have presented some fundamental extensions of the CS formulation. Under the ADMM frameworks, it appears that it is possible to consider extensions based on the combination of the basics extensions presented. For example, the ℓ_1 loss could be used with affine constraint or in multi-task setting, etc. Such extensions will be worthwhile investigation for future work.

Regularization Path. In practice, the optimal value of the regularization λ is not known in advance, and thus one needs

to select a proper value to do robust CS recovery. Such a problem is known in statistics as model selection. Typically, one needs to compute the recovery along the regularization path, and select the one which meets the ℓ_1 norm constraint. This is discussed in detail in [35]. Essentially, some estimates of the noise statistics must be obtained in order to construct the bound on the residual ε. It is well-known that there exist a $\lambda_{\max} = \|\mathbf{\Phi}^T \mathbf{y}\|_\infty$ above which the solution is zero. For decreasing values of λ, the residual $\mathbf{r} = \mathbf{y} - \mathbf{\Phi} \hat{\mathbf{x}}$ will become smaller whilst the recovery becomes denser. The optimal λ is the maximum value of λ such that the bound constraint on the residual vector is met. In CS recovery, this happens when $\|\mathbf{r}\|_2^2 \leq \varepsilon$, whilst in robust CS recovery, Pham and Venkatesh [35] have suggested $\rho(\mathbf{r}) \leq \varepsilon$, which is a generalization of the CS selection criteria for the robust case. In our implementation, we combine a coarse grid search and a fine bi-section search to find this optimal λ (see Fig. 1 for an illustration).

Cholesky Factorization. As can be seen, most update step of $\mathbf{x}$ in different ADMM variants involves the computation of the form $\mathbf{x}^{k+1} = (\mu \mathbf{\Phi}^T \mathbf{\Phi} + \mathbf{Q})^{-1} \mathbf{q}$ where $\mathbf{Q}$ is an positive definite matrix. The matrix under inversion has a size of $N \times N$ and it is large in image processing application. Thus, it is inefficient to compute the inversion directly to obtain the update. A much more efficient approach is to use Cholesky decomposition to achieve the goal. It is known from linear algebra that if $\mathbf{H}$ is a positive definite matrix then it admits the factorization $\mathbf{H} = \mathbf{L}\mathbf{L}^T$ and thus $\mathbf{H}^{-1}\mathbf{q}$ can be efficiently computed by solving $\mathbf{L}\mathbf{x}_1 = \mathbf{q}$ first, then $\mathbf{L}^T\mathbf{x} = \mathbf{x}_1$, which can be written as $\mathbf{x} = \mathbf{L}^T \backslash (\mathbf{L} \backslash \mathbf{q})$. For compressed sensing applications where $\mathbf{\Phi}$ is a fat matrix, further exploitation can be made by reducing the dimension of the matrix for Cholesky factorization. Indeed, according to the matrix inversion lemma

$$
(\mu \mathbf{\Phi}^T \mathbf{\Phi} + \mathbf{Q})^{-1} = \mathbf{Q}^{-1} - \mathbf{Q}^{-1} \mathbf{\Phi}^T \mathbf{P}^{-1} (\mu \mathbf{\Phi} \mathbf{Q}^{-1}),
$$

where $\mathbf{P} = \mathbf{I} + \mu \mathbf{\Phi} \mathbf{Q}^{-1} \mathbf{\Phi}^T$. Suppose that the Cholesky factorization of $\mathbf{P}$ is $\mathbf{P} = \mathbf{L}\mathbf{L}^T$ then

$$
(\mu \mathbf{\Phi}^T \mathbf{\Phi} + \mathbf{Q})^{-1} \mathbf{q} = \mathbf{Q}^{-1}(\mathbf{q} - \mathbf{\Phi}^T (\mathbf{L}^T \backslash (\mathbf{L} \backslash (\mu \mathbf{\Phi} \mathbf{Q}^{-1} \mathbf{q})))).
$$

We can avoid the direct inversion of $\mathbf{Q}$ by exploiting the fact that if $\mathbf{Q} = \rho_1 \mathbf{I} + \rho_2 \mathbf{c}\mathbf{c}^T$ then the matrix inversion lemma once again gives

$$
\mathbf{Q}^{-1} = \rho_1^{-1} \mathbf{I} - \gamma \mathbf{c}\mathbf{c}^T,
$$

where $\gamma = \rho_1^{-2}(\rho_2^{-1} + \rho_1^{-1} \mathbf{c}^T \mathbf{c})$. Finally, we note that this Cholesky factorization is independent of the regularization parameter λ and thus it can be cached for the whole regularization path to reduce computation.

V. EXPERIMENTS

A. Numerical Properties

1) Algorithmically Efficient Robust CS Algorithms: First, we examine the convergence property of the FISTA and ADMM algorithms and compare them with the previously proposed method in [35], which we refer to as nested robust CS algorithm due to the nature of the double loops inside that algorithm. As the nested robust CS algorithm [35] is dependent

on the particular CS solver being used for the inner loop, we select the ADMM implementation as the CS solver as it provides the best computational accuracy and speed. Note that Pham and Venkatesh [35] used the `ll_ls` algorithm originally, which is known for high-accuracy but computationally expensive. However, numerical experience shows that the inner steps do not required to be solved with high accuracy. Thus, the ADMM implementation as a CS solver for the nested robust CS algorithm is better overall. In this case, it can be seen that the computational complexity per iteration (regardless of inner or outer) in all compared algorithms are approximately the same: they all involve the computation of the majorization point and the soft-thresholding shrinkage operation.

To compare the algorithms, we examine two aspects: the error versus the iterations and the computational time taken to achieve a particular tolerance. Whilst the former indicates how fast an algorithm converges, the latter provides a much valuable insight for practical purpose. To do so, we let all algorithms run for sufficiently large number of iterations and measure the error (with respect to the true value of the robust CS solution) as iterations go on, and the computational time taken when the error reaches certain thresholds. For the ADMM-based CS solver used in the inner loop of the nested robust CS algorithm, we select the termination with relative tolerance of 10^{-2} and absolute tolerance of 10^{-4} (see [5, p.19]). This allows a reasonable convergence within the inner loops. We also choose the modified residual approach for nested robust CS as it is simpler without loosing convergence advantage. All algorithms are implemented in Matlab, and roughly optimized.

We revisit the random bars example in [35] and the results of this study is shown in Fig. 2. In this example, the signal to noise ratio is 20dB and the impulsive noise is modeled as a two-component Gaussian mixture model where the there is 10% contamination whose variance is $\kappa = 100$ times that of the main component. Here, the left subplot shows the reduction of the error versus the iterations, whilst the right plot shows the time taken to achieve the relative accuracy from initialized zeros (as indicated by 1E0) to as small as 10^{-10} of the initial error (as indicated by 1E-10). We note the error profile of the nested robust CS algorithm ranges considerably due to the fact that we measure with respect to the global solution of the outer loop and that within each CS inner loop the algorithm still converges normally.

Clearly the error profile plot indicates that the ADMM algorithm offers the best convergence speed per iteration, followed by the FISTA algorithm. For example, to achieve an accuracy of 10^{-5} of the initial error, it only takes the ADMM algorithm less than 100 iterations, whilst the FISTA algorithm needs to spend more than 20 times, and the nested algorithm would need 200 times the number of iterations. In terms of the actual time taken to achieve a particular tolerance, the right subplot further indicates the advantage of ADMM and FISTA algorithms over the nested one. In practice, one would be interested in the tolerance of between 10^{-2} to 10^{-6}, over which the ADMM and FISTA algorithms are observed to be 100 and 10 times faster than the nested algorithm respectively.

2) Affine Robust CS Algorithm: Next, we examine how much improvement can be made to robust CS if the power is known. The affine robust CS formulation is slightly different to the robust CS formulation in that additional constraint $\mathbf{c}^T\mathbf{x} = 1$ is imposed, and here we select $\mathbf{c} = 1/(\mathbf{1}^T\mathbf{x}) = 1/\sum_i x_i$ and assume that $\sum_i x_i$ is known.

We study the convergence behavior of the affine ADMM robust CS algorithm to solve this formulation by revisiting the random bars example. In this case, we select $\rho_1 = \rho_2 = 1$ and let the algorithm run over sufficient number of iterations. The results are shown in Fig. 3. Again, the left subplot shows the absolute error against the iterations whilst the right subplots indicates computational time taken to reach a particular accuracy. Compared with those of the ADMM robust CS algorithm, it can be clearly seen that the affine ADMM robust CS algorithm takes more time to reach. This is as expected because there are only minor changes to the update steps of the primal and dual variables.

3) Robust CS with ℓ_1 loss: Next, we demonstrate the robust CS algorithm with ℓ_1 loss function rather than the Huber's loss function used in [35]. This is useful in situations with very impulsive corruption, where the noise is best modeled by a Cauchy distribution. To do so, we revisit the random bars example, but we use Cauchy noise instead. For the ℓ_1 ADMM robust CS algorithm, the model selection criteria is the ℓ_1 norm of the residual, rather than the Huber's loss function to reflect the new formulation. Other than that, all other experimental settings remain the same.

We examine the convergence behavior of the ADMM robust CS algorithm with ℓ_1 loss. Fig. 4 shows the typical convergence behavior of the algorithm in terms of accuracy versus iterations (left) and computational time taken to reach certain accuracy (right). It is observed that the convergence is slower with modest accuracy as compared with the formulation using Huber's loss function. This is as expected from ADMM optimization theory due to an increasing number of variables to solve the ℓ_1 loss formulation. Nevertheless, modest accuracy might be sufficient for many practical situations.

B. Image Recovery in Impulsive Noise

Whilst the primary interest is on improving robust CS, it is also relevant to examine the compressed image recovery qualitatively in the presence of impulsive noise. We compare image recovery of the following method: original compressed sensing formulation (CS), robust CS using the nested algorithm [35], robust CS using the ADMM algorithm, affine robust CS, ℓ_1 robust CS. In addition, we also compare against two related formulations: one is Wagner *et al* [39] and the other is Lorentzian BP [9], which have recently been brought to our attention. The former is a special case of the ℓ_1 robust CS formulation and is solved via a hybrid algorithm that combines both alternative Lagrangian multiplier (ALM) and FISTA. The latter uses a special Lorentzian loss function. For Lorentizan BP, it is noted that the authors did not explicitly indicate how to solve the formulation, and thus we solve it using the FISTA framework.

Again, we revisit the random bars example as in the previous experiments with the same settings. Here, we do
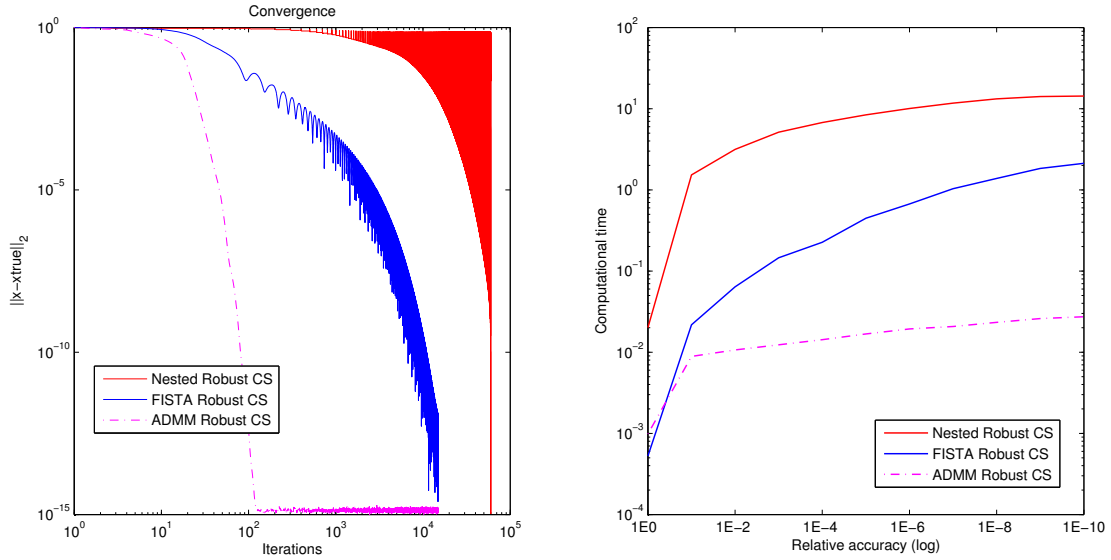
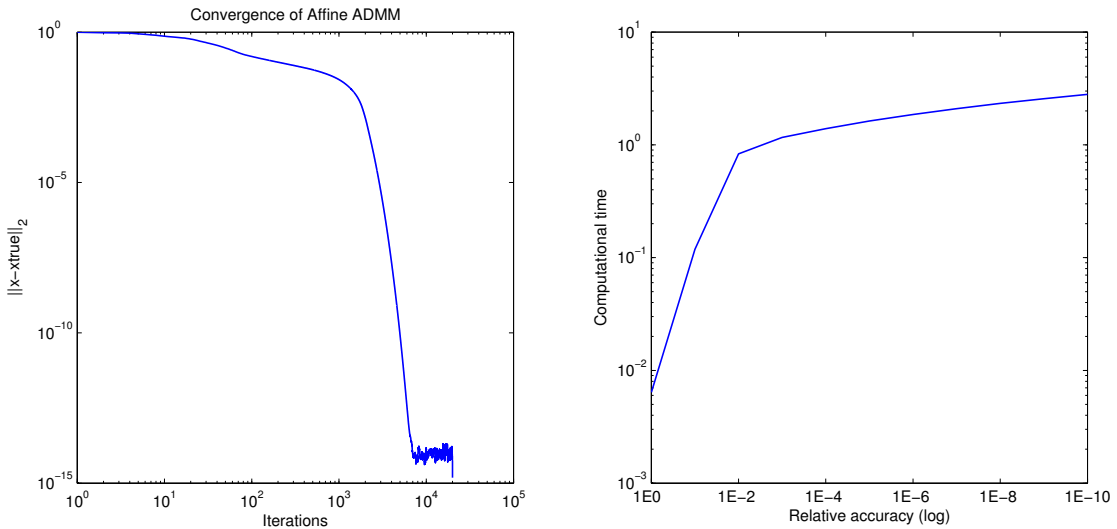Fig. 2. Convergence property of the compared robust CS algorithms

Fig. 3. Convergence behavior of affine robust CS

extra modeling of impulsive noise using both the Gaussian mixture ε-contaminated model, and the Cauchy distribution. The recovery results for Gaussian mixture and Cauchy noise are shown in Figs. 5 and 6 respectively.

We make the following observations from the results

- The recovery of robust CS implemented using the efficient ADMM scheme is similar to that of the nested algorithm in [35]. This is as expected because they both solve the same formulation. In practice, we observe that the ADMM implementation is at least twice faster than the nested algorithm for similar accuracy, taking into acount all overheads. So the ADMM implementation is prefered for robust CS.

- The affine formulation does not provide significant benefit for the cases being considered.

- The original CS formulation fails in the case of Cauchy noise, which is evidenced by its very noisey recovery.

- The best performer in both cases is the robust CS with ℓ_1 loss. This is interesting as it performs well also in Gaussian mixture noise and suggests that further study to help understand this will be needed. The variation proposed in Wagner *et al.* [39] is also close. This suggests that it is the choice of the loss that is most important and the regularization parameter provides small additional benefit.

- Suprisingly, the Lorentizan BP performs quite poorly. It is only slightly better than the original CS formulation. In addition, we found in practice that Lorentzian BP converges rather slowly, even when the FISTA algorithm is used, and that very careful initialization is required.
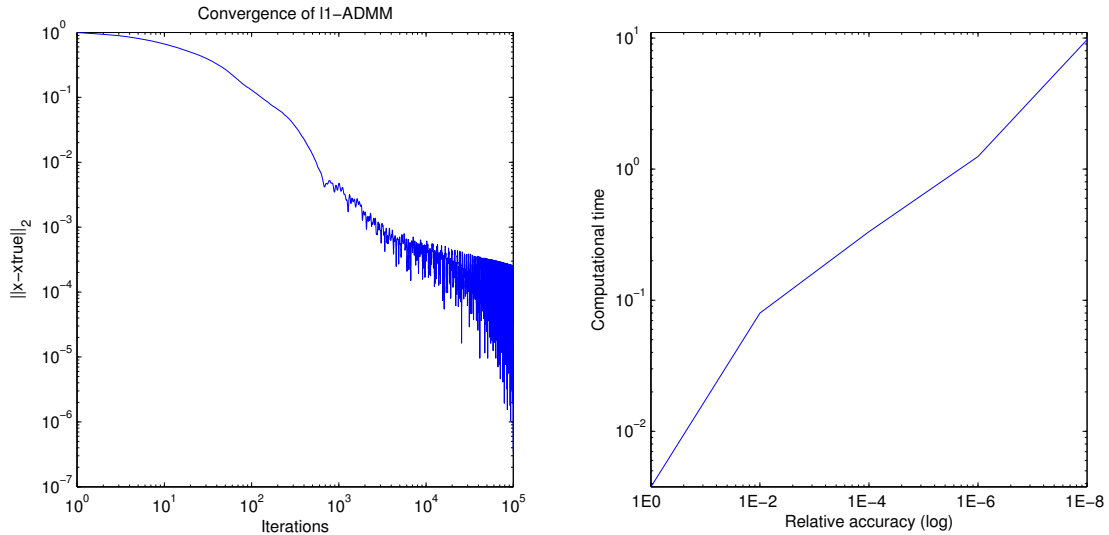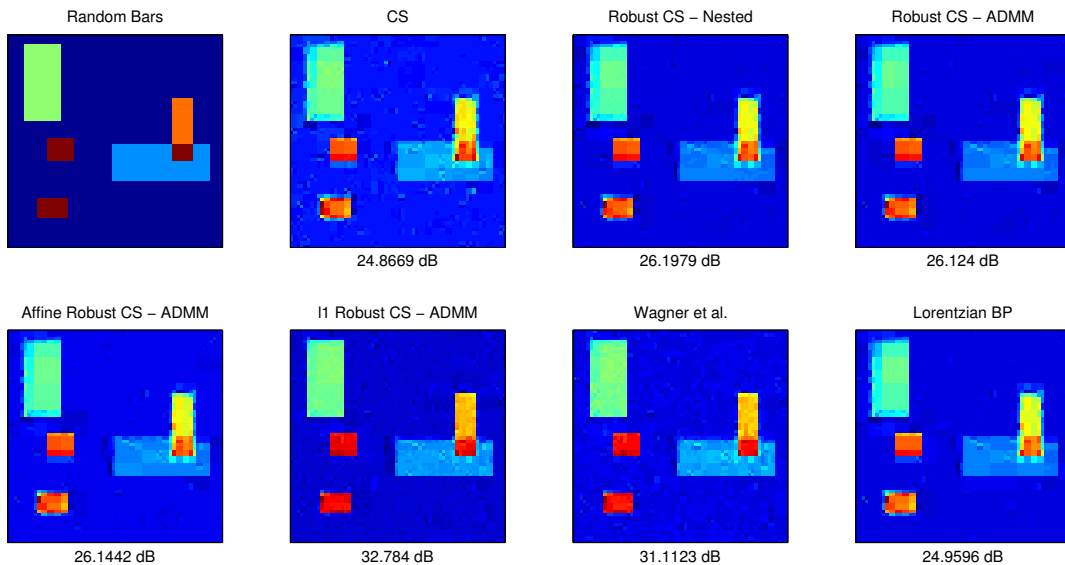
Fig. 4. Convergence of the ℓ_1-loss ADMM robust CS algorithm

Fig. 5. Image recovery in impulsive Gaussian mixture noise

C. CS Sampling Factor

It is well known from CS theory that in order for good recovery, the CS sampling factor M/N, which is the ratio between the dimension of the compressed and the original data, must be sufficiently large. The minimum number of measurements M is often shown in CS theoretical work to be dependent on N by a factor whose order is governed by the sparsity of the underlying signal. Here, we aim to study the dependence of recovery on the CS sampling factor M/N for all methods. With the same settings as above, the random bars example is revisited. But now we vary the sampling factor M/N from 0.05 to 0.4. The recovery PSNRs for both the Gaussian mixture and Cauchy noise case are graphically shown in Figs. 7 and 8 respectively. Here, we observe that the quality of the recovery degrades somewhat linearly as the CS sampling factor become smaller for the region smaller than 0.3. In both cases, the robust CS with ℓ_1 loss demonstrates the overall best recovery performance, followed by the method by Wagner *et al.*. Both CS and Lorentzian BP struggle considerably in Cauchy noise, whilst other robust method exhibit similar dependency pattern over the range. Unless a pre-defined threshold is available, the experimental results suggest that there is no sharp turning (break-down) point for the CS sampling factor. Rather, the larger the value the better PSNR a given robust CS recovery algorithm can achieve.
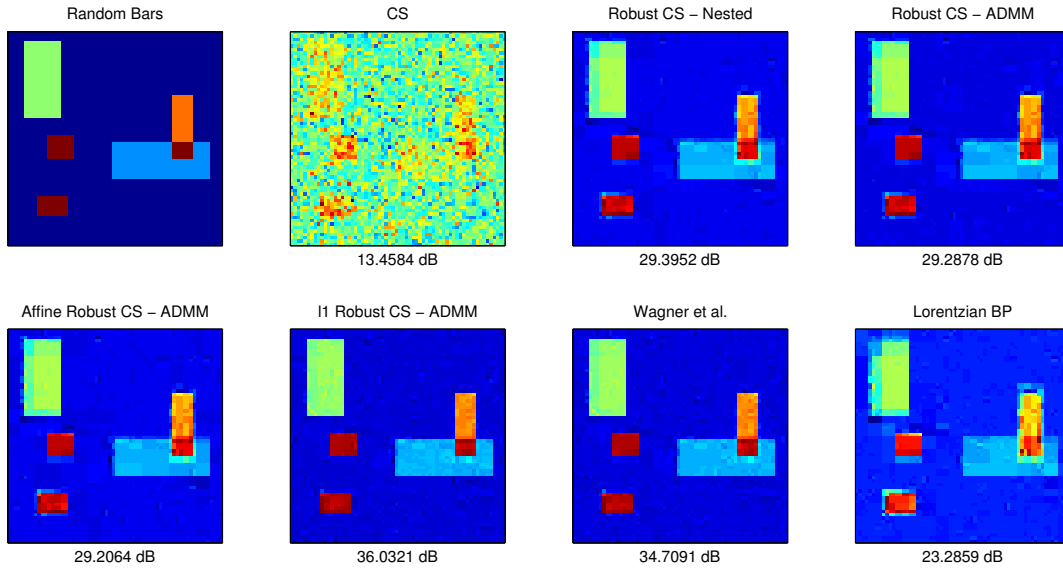
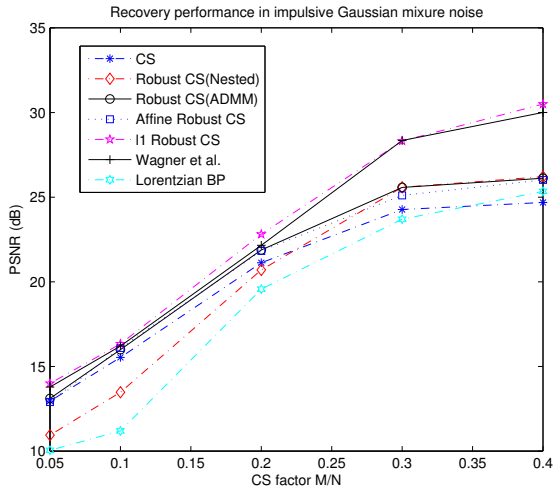Fig. 6. Image recovery in impulsive Cauchy noise

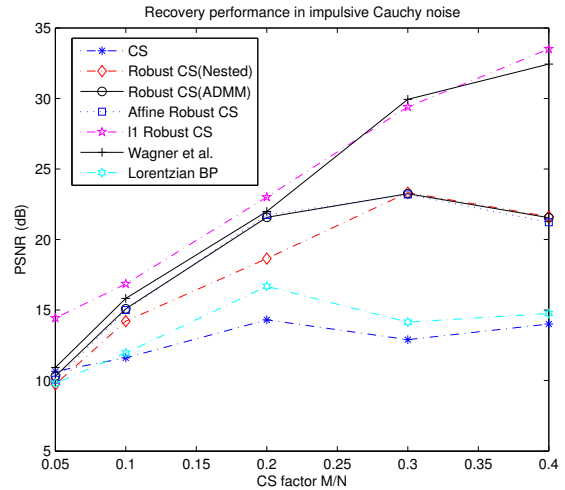Fig. 7. Recovery dependency on CS sampling factor - impulsive Gaussian mixture noise

Fig. 8. Recovery dependency on CS sampling factor - impulsive Cauchy noise

D. Recovery of A Sequence of Compressed Images

Finally, we demonstrate the usefulness of the multi-task robust CS formulation when a sequence of 10 compressed images corrupted by impulsive noise needs to be recovered. Whilst each image in the sequence can be recovered separately, the multi-task robust CS formulation suggests that exploiting the shared structure between the tasks may provide better recovery. To do so, we consider a sequence of random bars frames shown in the top row of Fig. 10. Here, there are common static random bars and a moving block across the frames. Obviously, the wavelet coefficients for common static bars are shared between the CS tasks. Only the coefficients corresponding to the moving block distinguish between tasks. This is clearly illustrated in Fig. 9 which shows an image plot of Haar wavelet coefficients of all 10 random bars image in a sequence: the horizontal lines correspond to common coefficients.

The settings for the recovery are the same as previous experiment. For robust CS, we select the ADMM algorithm, and similarly for multi-task robust CS we also select the corresponding multi-task ADMM algorithm. The first 4 recovered images are shown in Fig. 10: the second row shows CS recovery, the third row shows robust CS recovery, and finally the last row shows multi-task robust CS recovery. The actual PSNRs for every frame are shown on Fig. 11. Here, we observe clearly that, on average, the multi-task robust CS formulation does provide significant improvement over the robust CS formulation, both of which outperform CS recovery considerably.
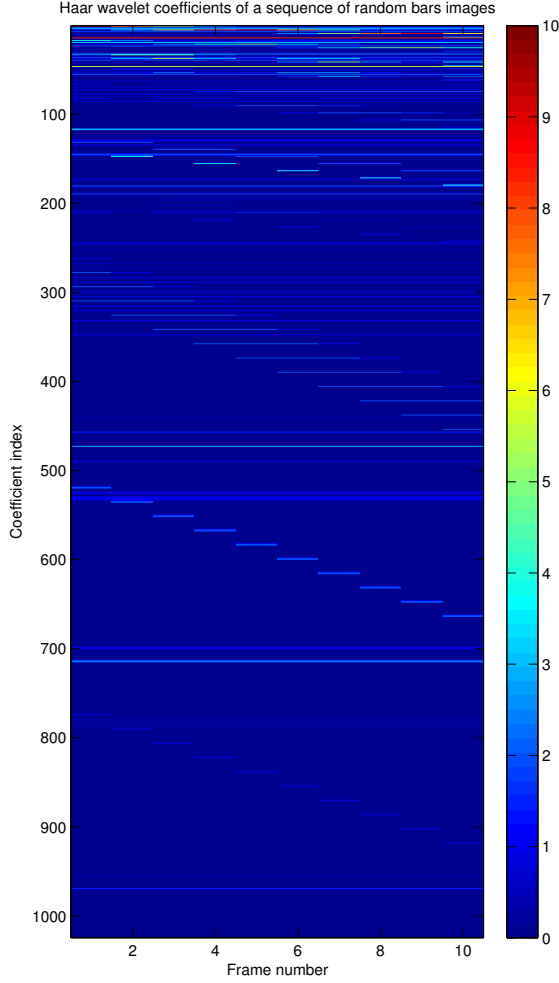
Fig. 9. Haar wavelet coefficients - Multi random bars

Fig. 11. PSNR Comparison of CS, robust CS, and multi-task (MT) robust CS

VI. CONCLUSION

We have presented more computationally efficient and extendable approaches to the recently proposed robust CS algorithm. We have also extended robust CS formulation in a number of ways, including affine constraints, ℓ_1-loss function, and multi-task formulation. For improving computational efficiency of robust CS, we found that the (generalized) ADMM robust CS algorithm is the best, then followed by the FISTA robust CS algorithm. We also found that imposing affine constraint can provide improvement, though slightly. The striking result is that ℓ_1 loss formulation for robust CS seems to offer considerable gain over the Huber's loss formulation, despite the fact that its convergence seems slower. Finally, in the case where one needs to robustly recover a sequence of compressed images, the multi-task formulation is proved to provide additional advantages in terms of both PSNR output and computational speed.
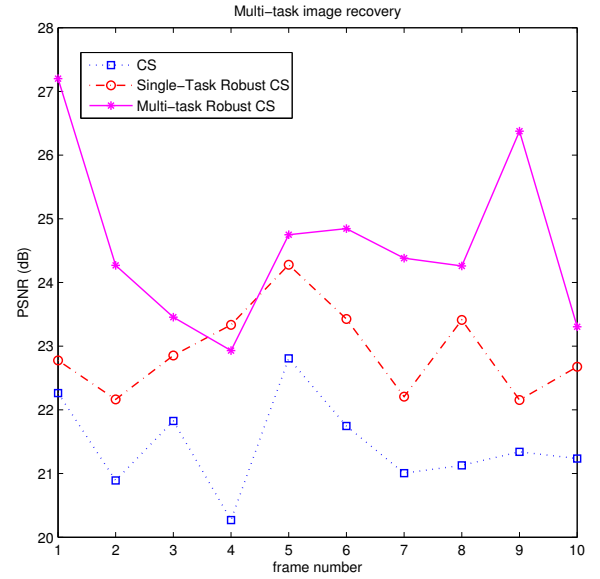
APPENDICES

Proof of Lemma 1

We start from the definition of the Lipchitz constant as a term such as

$$\sup_{x_1, x_2 \in \mathcal{X}} |f(x_1) - f(x_2)| \le L_f |x_1 - x_2|. \tag{65}$$

As there are two possible scenarios $x_1, x_2 \in \mathcal{X}_1$, $x_1, x_2 \in \mathcal{X}_2$, and $x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2$ and from the definition of L_g and L_h, we immediately have

$$L_f \le \max\{L_g, L_h, L_{12}\}, \tag{66}$$

where L_{12} is defined as the minimum constant such that

$$\sup_{x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2} |g(x_1) - h(x_2)| \le L_{12} |x_1 - x_2|. \tag{67}$$

Let $\mathcal{X}_3 = \mathcal{X}_1 \cap \mathcal{X}_2$. For arbitrary $x_1 \in \mathcal{X}_1$ and $x_2 \in \mathcal{X}_2$ we construct $x_3 \in \mathcal{X}_3$ such that it is a convex combination of x_1 and x_2, so that $|x_2 - x_3| \le |x_1 - x_2|$ and $|x_1 - x_3| \le |x_1 - x_2|$. Then using triangle inequalities and definitions of L_g and L_h, we have

$$
\begin{aligned}
\sup |g(x_1) - h(x_2)| &= \sup |g(x_1) - g(x_3) + g(x_3) + h(x_2)| \\
&= \sup |g(x_1) - g(x_3) + h(x_3) - h(x_2)| \\
&\le \sup |g(x_1) - g(x_3)| + |h(x_3) - h(x_2)| \\
&\le \sup |g(x_1) - g(x_3)| + \sup |h(x_2) - h(x_3)| \\
&\le L_g |x_1 - x_3| + L_h |x_2 - x_3| \\
&\le L_g |x_1 - x_2| + L_h |x_2 - x_1| \\
&\le (L_g + L_h)|x_1 - x_2|. \tag{68}
\end{aligned}
$$

The proof immediate follows from (66) and (68).

References

[1] F. Bach, "Consistency of the group lasso and multiple kernel learning," *The Journal of Machine Learning Research*, vol. 9, pp. 1179–1225, 2008.

[2] L. Bar, A. Brook, N. Sochen, and N. Kiryati, "Deblurring of color images corrupted by impulsive noise," *IEEE Transactions on Image Processing*, vol. 16, no. 4, pp. 1101–1111, 2007.

[3] R. Baraniuk, V. Cevher, M. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1982–2001, 2010.

[4] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Foundations and Trends in Machine Learning*. Now Publisher, 2011, vol. 3, no. 1, ch. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, pp. 1–122.

[6] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[7] E. Candes and T. Tao, "Near-optimal signal recovery from random projections: universal encoding strategies," *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[8] E. Candes and Y. Plan, "A probabilistic and ripless theory of compressed sensing," *IEEE Transactions on Information Theory*, vol. 57, no. 11, pp. 7235–7254, 2011.

[9] R. E. Carrillo, K. E. Barner, and T. C. Aysal, "Robust sampling and reconstruction methods for sparse signals in the presence of impulsive noise," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 392–408, 2010.

[10] R. Chan, C.-W. Ho, and M. Nikolova, "Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1479–1485, 2005.

[11] W. Chan, M. Moravec, R. Baraniuk, and D. Mittleman, "Terahertz imaging with compressed sensing and phase retrieval," *Optics letters*, vol. 33, no. 9, pp. 974–976, 2008.

[12] P. Civicioglu, "Using uncorrupted neighborhoods of the pixels for impulsive noise suppression with ANFIS," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 759–773, 2007.

[13] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Inform. Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.

[14] D. Donoho, "Compressed Sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[15] D. Donoho and G. Reeves, "The sensitivity of compressed sensing performance to relaxation of sparsity," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2012, pp. 2211–2215.

[16] J. Eckstein and D. Bertsekas, "On the Douglas - Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, no. 1, pp. 293–318, 1992.

[17] Y. Eldar, "Compressed sensing of analog signals in shift-invariant spaces," *IEEE Transactions on Signal Processing*, vol. 57, no. 8, pp. 2986–2997, 2009.

[18] Z. Guo, T. Wittman, and S. Osher, "L1 unmixing and its application to hyperspectral image enhancement," in *Proceedings SPIE Conference on Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XV*, vol. 7334, 2009, pp. 73 341M–73 341M.

[19] E. Hale, W. Yin, and Y. Zhang, "A fixed-point continuation method for ℓ_1 regularized minimization with applications to compressed sensing," *Preprint*, 2007.

[20] T. Hashimoto, "Bounds on a probability for the heavy tailed distribution and the probability of deficient decoding in seequential decoding," *IEEE Transactions on Information Theory*, vol. 51, no. 3, pp. 990–1002, 2005.

[21] J. Haupt and R. Nowak, "Compressive sampling vs. conventional imaging," in *Proceedings of the International Conference on Image Processing (ICIP)*, 2006.

[22] ——, "Signal reconstruction from noisy random projections," *IEEE Transactions on Information Theory*, vol. 52, no. 9, pp. 4036–4048, 2006.

[23] M. Herman and T. Strohmer, "High-resolution radar via compressed sensing," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2275–2284, 2009.

[24] J. Huang and T. Zhang, "The benefit of group sparsity," *The Annals of Statistics*, vol. 38, no. 4, pp. 1978–2004, 2010.

[25] P. J. Huber, *Robust Statistics*. New York: Wiley, 1981.

[26] D. Hunter and K. Lange, "A tutorial on MM algorithms," *The American Statistician*, vol. 58, no. 1, pp. 30–37, 2004.

[27] S. Ji, Y. Xue, and L. Carin, "Bayesian Compressive Sensing," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2346–2356, 2008.

[28] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "A method for large-scale ℓ_1-regularized least squares," *IEEE Journal on Selected Topics in Signal Processing*, vol. 4, no. 1, pp. 606–617, 2007.

[29] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magnetic resonance in medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.

[30] X. Lv, G. Bi, and C. Wan, "The group Lasso for stable recovery of block-sparse signal representations," *IEEE Trans. Sig. Process.*, no. 99, 2011.

[31] S. G. Mallat and Z. Zhang, "Matching pursuit with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, Dec 1993.

[32] M. Mishali and Y. Eldar, "Blind multiband signal reconstruction: Compressed sensing for analog signals," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 993–1009, 2009.

[33] D. Needell and J. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[34] Y. Nesterov, *Gradient methods for minimizing composite objective function*. CORE, 2007.

[35] D. Pham and S. Venkatesh, "Improved image recovery from compressed data contaminated with impulsive noise," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 397–405, 2012.

[36] J. Provost and F. Lesage, "The application of compressed sensing for photo-acoustic tomography," *IEEE Transactions on Medical Imaging*, vol. 28, no. 4, pp. 585–594, 2009.

[37] N. Rao, R. Nowak, S. Wright, and N. Kingsbury, "Convex approaches to model wavelet sparsity patterns," in *Proceedings of the IEEE International Conference on Image Processing*. IEEE, 2011, pp. 1917–1920.

[38] N. Vaswani, "Kalman filtered compressed sensing," in *Proceedings of the IEEE International Conference on Image Processing*. IEEE, 2008, pp. 893–896.

[39] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, H. Mobahi, and Y. Ma, "Toward a practical face recognition system: Robust alignment and illumination by sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 372–386, 2012.

[40] P. Windyga, "Fast impulsive noise removal," *IEEE Transactions on Image Processing*, vol. 10, pp. 173–179, 2001.

[41] J. Yang and Y. Zhang, "Alternating direction algorithms for \ell_1-problems in compressive sensing," *SIAM journal on scientific computing*, vol. 33, no. 1, pp. 250–278, 2011.

[42] M. Zibulevsky and M. Elad, "L1-l2 optimization in signal and image processing," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 76–88, 2010.

[43] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.
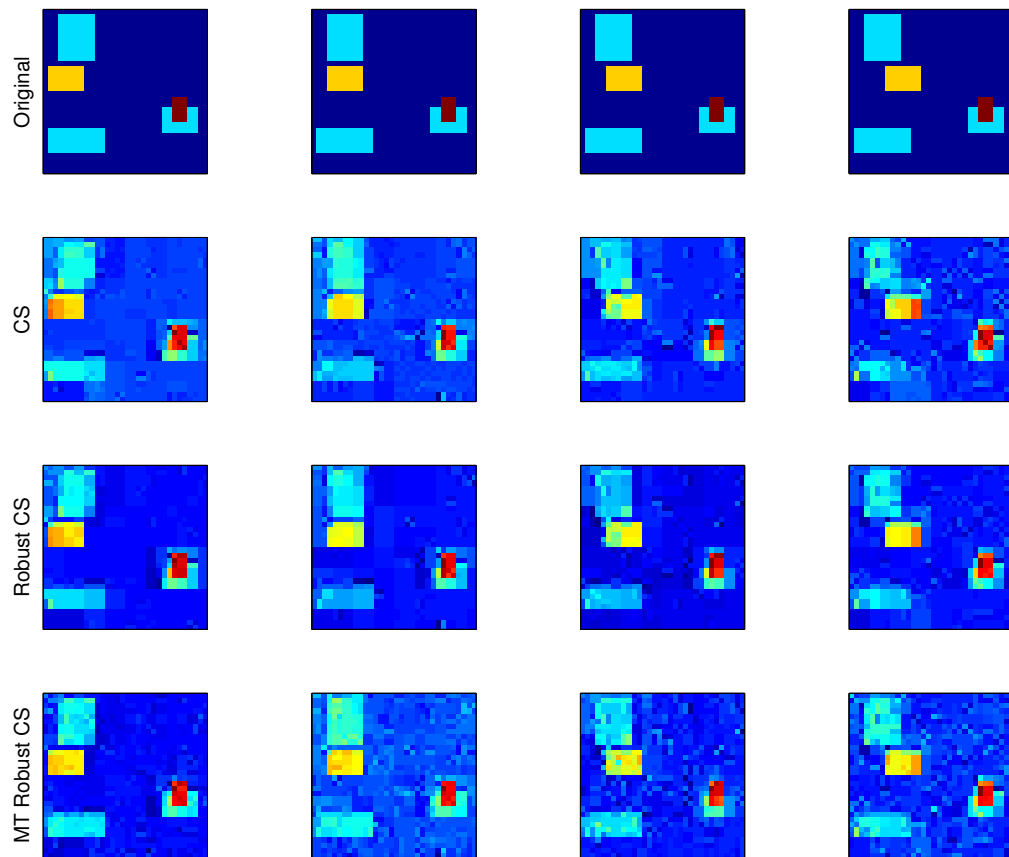
Fig. 10. Recovery of sequence of compressed images