

# Group-Sparse Signal Denoising: Non-Convex Regularization, Convex Optimization

Po-Yu Chen and Ivan W. Selesnick

**Abstract**—Convex optimization with sparsity-promoting convex regularization is a standard approach for estimating sparse signals in noise. In order to promote sparsity more strongly than convex regularization, it is also standard practice to employ non-convex optimization. In this paper, we take a third approach. We utilize a non-convex regularization term chosen such that the total cost function (consisting of data consistency and regularization terms) is convex. Therefore, sparsity is more strongly promoted than in the standard convex formulation, but without sacrificing the attractive aspects of convex optimization (unique minimum, robust algorithms, etc.). We use this idea to improve the recently developed ‘overlapping group shrinkage’ (OGS) algorithm for the denoising of group-sparse signals. The algorithm is applied to the problem of speech enhancement with favorable results in terms of both SNR and perceptual quality.

**Index Terms**—group sparse model; convex optimization; non-convex optimization; sparse optimization; translation-invariant denoising; denoising; speech enhancement

## I. INTRODUCTION

In this work, we address the problem of estimating a vector,  $\mathbf{x}$ , from an observation,  $\mathbf{y}$ ,

$$y(i) = x(i) + w(i), \quad i \in \mathbb{Z}_N = \{0, \dots, N-1\}, \quad (1)$$

where  $w$  is additive white Gaussian noise (AWGN). We assume that  $\mathbf{x}$  is a group-sparse vector. By group-sparse, we mean that large magnitude values of  $\mathbf{x}$  tend not to be isolated. Rather, large magnitude values tend to form clusters (groups). Furthermore, we do not assume that the group locations are known, nor that the group boundaries are known. In fact, we do not assume that the groups have well defined boundaries. An example of such a vector (in 2D) is the spectrogram of a speech waveform. The spectrogram of a speech waveform exhibits areas and ridges of large magnitude, but not isolated large values. The method proposed in this work will be demonstrated on the problem of speech filtering.

Convex and non-convex optimization are both common practice for the estimation of sparse vectors from noisy data [1]. In both cases one often seeks the solution  $\mathbf{x}^* \in \mathbb{R}^N$  to the problem

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \left\{ F(\mathbf{x}) = \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda R(\mathbf{x}) \right\}$$

where  $R(\mathbf{x}) : \mathbb{R}^N \rightarrow \mathbb{R}$  is the regularization (or penalty) term and  $\lambda > 0$ . Convex formulations are advantageous in

that a wealth of convex optimization theory can be leveraged and robust algorithms with guaranteed convergence are available [8]. On the other hand, non-convex approaches are advantageous in that they usually yield sparser solutions for a given residual energy. However, non-convex formulations are generally more difficult to solve (due to suboptimal local minima, initialization issues, etc.). Also, solutions produced by non-convex formulations are generally discontinuous functions of input data (e.g., the discontinuity of the hard-threshold function).

Generally, convex approaches are based on sparsity-promoting convex penalty functions (e.g., the  $\ell_1$  norm), while non-convex approaches are based on non-convex penalty functions (e.g., the  $\ell_p$  pseudo-norm with  $p < 1$  [39], re-weighted  $\ell_2/\ell_1$  [10], [63]). Other non-convex algorithms seek sparse solutions directly (e.g., OMP [40], iterative hard thresholding [6], [25], [34], [51], and greedy  $\ell_1$  [37]).

In this work, we take a different approach, proposed by Blake and Zimmerman [5] and by Nikolova [43]. Namely, the use of a non-convex non-smooth penalty function chosen such that the total cost function  $F$  (consisting of data consistency and regularization terms) is strictly convex. This is possible “by balancing the positive second derivatives in the [data consistency term] against the negative second derivatives in the [penalty] terms” [5, page 132]. This idea has been further extended by Nikolova et al. [44], [46]–[48].

The contribution of this work relates to (1) the formulation of the group-sparse denoising problem as a convex optimization problem albeit defined in terms of a non-convex penalty function, and (2) the derivation of a computationally efficient iterative algorithm that monotonically reduces the cost function value. We utilize non-convex penalty functions (in fact, concave on the positive real line) with parametric forms; and we identify an interval for the parameter that ensures the strict convexity of the total cost function,  $F$ . As the total cost function is strictly convex, the minimizer is unique and can be obtained reliably using convex optimization techniques. The algorithm we present is derived according to the principle of majorization-minimization (MM) [24]. The proposed approach:

- 1) does not underestimate large amplitude components of sparse solutions to the extent that convex penalties do,
- 2) is translation invariant (due to groups in the proposed method being fully overlapping),
- 3) is computationally efficient ( $O(N)$  per iteration) with monotonically decreasing cost function, and
- 4) requires no algorithmic parameters (step-size, Lagrange, etc.).

We demonstrate below that the proposed approach substan-

Copyright (c) 2014 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

The authors are with the Department of Electrical and Computer Engineering, Polytechnic School of Engineering, New York University, 6 Metrotech Center, Brooklyn, NY 11201. Email: poyupaulchen@gmail.com, selesi@poly.edu, phone: 718 260-3416.

This research was supported by the NSF under Grant No. CCF-1018020.

tially improves upon our earlier work that considered only convex regularization [13]. The fact that convex penalties yield estimates that are biased toward zero (i.e., that underestimate large amplitude components) is discussed, for example, in [10], [11], [22], [26], [27]; see also Proposition 1 of [58].

### A. Related Work

The estimation and reconstruction of signals with group sparsity properties has been addressed by numerous authors. We make a distinction between two cases: non-overlapping groups [12], [21], [35], [36], [66] and overlapping groups [1]–[3], [14], [19], [23], [32], [33], [41], [50], [65]. The non-overlapping case is the easier case: when the groups are non-overlapping, there is a decoupling of variables, which simplifies the optimization problem. When the groups are overlapping, the variables are coupled. In this case, it is common to define auxiliary variables (e.g., through the variable splitting technique) and apply methods such as the alternating direction method of multipliers (ADMM) [7]. This approach increases the number of variables (proportional to the group size) and hence increases memory usage and data indexing. In previous work we describe the ‘overlapping group shrinkage’ (OGS) algorithm [13] for the overlapping-group case that does not use auxiliary variables. The OGS algorithm exhibits favorable asymptotic convergence in comparison with algorithms that use auxiliary variables [13, Fig. 5]. It was applied to total variation denoising in [59]. In comparison with previous work on convex optimization for overlapping group sparsity, including [13], [59], the approach we propose here promotes sparsity more strongly. In this paper, we extend the OGS algorithm to the case of non-convex regularization, yet the approach remains within the convex optimization framework.

As noted above, the balancing of the data consistency term and the penalty term, so as to formulate a convex problem with a non-convex penalty term, was described in Refs. [5], [43] and extended in [44], [46]–[48]. This approach was used to initialize a scheme named ‘graduated non-convexity’ (GNC) in [5]. The goal of GNC is to minimize a non-convex function  $F$  by minimizing a sequence of functions  $F_k$ ,  $k \geq 1$ . The first one is a convex approximation of  $F$ , and the subsequent ones are non-convex and progressively similar to  $F$ . In order that the initial approximation of  $F$  be convex, the penalty function must satisfy an eigenvalue condition [5]. A looser condition, which promotes sparsity more strongly, can be expressed as a semidefinite program (SDP), but this incurs a higher computational cost [58]. In the method described here, we use the same balancing idea as in GNC; however, our goal is to minimize a convex function, not a non-convex one as in GNC. In particular, we use the balancing idea to construct a convex function that maximally promotes sparsity, and we seek to subsequently solve this convex problem. We note that here our primary goal is to capture group sparsity behavior, which is not considered in the GNC work. We also note that the computationally demanding SDP arising in Ref. [58] does not arise in the current work. The algorithm developed here is computationally simple.

## II. PRELIMINARIES

### A. Notation

We will work with finite-length discrete signals which we denote in lower case bold. The  $N$ -point signal  $\mathbf{x}$  is written as

$$\mathbf{x} = [x(0), \dots, x(N-1)] \in \mathbb{R}^N.$$

We use the notation

$$\mathbf{x}_{i,K} = [x(i), \dots, x(i+K-1)] \in \mathbb{R}^K \quad (2)$$

to denote the  $i$ -th group of size  $K$ . We consistently use  $K$  (a positive integer) to denote the group size. At the boundaries (i.e., for  $i < 0$  and  $i > N - K$ ), some indices of  $\mathbf{x}_{i,K}$  fall outside  $\mathbb{Z}_N$ , where  $\mathbb{Z}_N$  is defined in (1). We take these values as zero; i.e., for  $i \notin \mathbb{Z}_N$ , we take  $x(i) = 0$ . The  $\ell_2$  and  $\ell_1$  norms are defined as usual:

$$\|\mathbf{x}\|_1 := \sum_i |x(i)|, \quad \|\mathbf{x}\|_2 := \left[ \sum_i |x(i)|^2 \right]^{1/2}. \quad (3)$$

We denote the non-negative real line as  $\mathbb{R}_+ := \{x \in \mathbb{R} : x \geq 0\}$  and the positive real line as  $\mathbb{R}_+^* := \{x \in \mathbb{R} : x > 0\}$ . Given a function  $f : \mathbb{R} \rightarrow \mathbb{R}$ , the left-sided and right-sided derivatives of  $f$  at  $x$  are denoted  $f'(x^-)$  and  $f'(x^+)$ , respectively. The notation  $A \setminus B$  denotes set difference; i.e.,  $A \setminus B = \{a \in A : a \notin B\}$ .

### B. Penalty Functions

We will make the following assumptions on the penalty function,  $\phi : \mathbb{R} \rightarrow \mathbb{R}$ .

- 1)  $\phi$  is continuous on  $\mathbb{R}$
- 2)  $\phi$  is twice differentiable on  $\mathbb{R} \setminus \{0\}$
- 3)  $\phi(-x) = \phi(x)$  (symmetric)
- 4)  $\phi'(x) > 0$ ,  $\forall x > 0$  (increasing on  $\mathbb{R}_+^*$ )
- 5)  $\phi'(0^+) = 1$  (unit slope at zero)
- 6)  $\phi''(x) \leq 0$ ,  $\forall x > 0$  (concave on  $\mathbb{R}_+^*$ )
- 7)  $\phi''(0^+) \leq \phi''(x)$ ,  $\forall x > 0$
- 8)  $\phi''(0^+)$  is finite.

We will utilize penalty functions parameterized by a scalar parameter,  $a > 0$ . We use the notation  $\phi(x; a)$  to denote the parameterized form.

Examples of parameterized penalty functions satisfying the assumptions above are the logarithmic penalty,

$$\phi_{\log}(x; a) = \frac{1}{a} \log(1 + a|x|), \quad (4)$$

the arctangent penalty [58],

$$\phi_{\text{atan}}(x; a) = \frac{2}{a\sqrt{3}} \left( \tan^{-1} \left( \frac{1 + 2a|x|}{\sqrt{3}} \right) - \frac{\pi}{6} \right), \quad (5)$$

and the first order rational function [28]

$$\phi_{\text{rat}}(x; a) = \frac{|x|}{1 + a|x|/2}. \quad (6)$$

The rational penalty is defined for  $a \geq 0$ . The log and atan penalties are defined for  $a > 0$ . Note that as  $a \rightarrow 0$ , the three penalty functions approach the absolute value function. They are illustrated in Fig. 1. The absolute value function,  $\phi_{\text{abs}}(x) = |x|$ , also satisfies the listed assumptions.

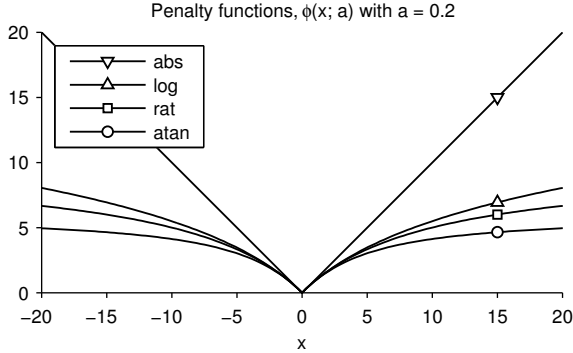


Fig. 1. Several sparsity promoting penalty functions satisfying the assumptions in Sec. II-B.

For later use, we record the value of the right-sided second derivative of the three non-convex penalty functions:

$$\phi''_{\log}(0^+; a) = \phi''_{\text{atan}}(0^+; a) = \phi''_{\text{rat}}(0^+; a) = -a. \quad (7)$$

Note that the  $\ell_p$  pseudo-norm ( $0 < p < 1$ ), i.e.,  $\phi(x) = |x|^p$ , does not satisfy the above assumptions. It does not have unit slope at zero nor can it be normalized or scaled to do so.

### C. Threshold Functions

Proximity operators are a fundamental tool in efficient sparse signal processing [16], [17]. In the scalar case, a proximity operator, which is defined in terms of a convex penalty,  $\phi$ , is a thresholding function when (and only when)  $\phi$  is non-smooth at zero. In this work, we utilize non-convex penalty functions; however, we can still define a threshold function similar to the definition of a proximity operator [56]. The following proposition is closely related to Lemma 3.1 in [42] and Theorem 3.3 in [45], both of which analyze the behavior of  $\theta$  for non-smooth, not necessarily convex,  $\phi$ .

**Proposition 1.** Define  $\theta : \mathbb{R} \rightarrow \mathbb{R}$  by

$$\theta(y) = \arg \min_{x \in \mathbb{R}} \left\{ G(x) = \frac{1}{2} |y - x|^2 + \lambda \phi(x) \right\} \quad (8)$$

where  $G : \mathbb{R} \rightarrow \mathbb{R}$ ,  $\lambda > 0$ , and  $\phi$  satisfies the assumptions in Sec. II-B. Suppose also that  $G$  is strictly convex. If  $|y| \leq \lambda$ , then the unique minimizer of  $G$  is zero. That is,  $\theta$  is a *threshold function* and  $\lambda$  is the *threshold value*. Also,  $\theta$  is an odd function.

*Proof.* This is a special case of Proposition 3 wherein  $\theta$  is a multivariate threshold function,  $\theta : \mathbb{R}^K \rightarrow \mathbb{R}^K$ .  $\square$

Figure 2 illustrates threshold functions corresponding to several penalty functions. We use  $\lambda = 4$  and  $a = 0.2$ . The threshold function corresponding to the absolute value penalty function is called the *soft* threshold function [20]. Notice that, except for the soft threshold function, the threshold functions approach the identity function asymptotically. The atan threshold function approaches identity the fastest.

The fact that the soft threshold function reduces large values by a constant amount is considered its deficiency. In the estimation of sparse signals in AWGN, this behavior results in

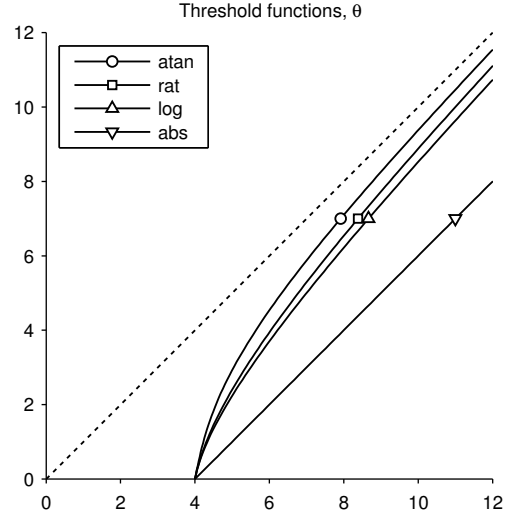


Fig. 2. Threshold functions derived from the four penalty functions given in Sec. II-B; three of which are non-convex.

a systematic underestimation (bias) of large magnitude signal values [22]. Hence, threshold functions that are asymptotically unbiased are often preferred to the soft threshold function, and the penalty functions from which they are derived promote sparsity more strongly than the  $\ell_1$  norm [10], [11], [26], [27]. The atan penalty function is derived specifically for its favorable behavior in this regard [58].

Proposition 1 and Figs. 1 and 2 illustrate the fact that threshold functions derived from suitable non-convex penalties converge asymptotically to the identity function (hence, they do not underestimate large amplitudes), and furthermore, that the total cost function can be, at the same time, convex.

As shown in Ref. [58], if  $\phi$  satisfies the assumptions in Sec. II-B, then the right-sided derivative of  $\theta$  at the threshold  $\lambda$  is given by

$$\theta'(\lambda^+) = \frac{1}{1 + \lambda \phi''(0^+)}. \quad (9)$$

Hence, with parameters  $\lambda = 4$  and  $a = 0.2$ , we use (7) to find that  $\theta'(\lambda^+) = 5$  for  $\phi_{\log}$ ,  $\phi_{\text{atan}}$ , and  $\phi_{\text{rat}}$ . That is, each of the threshold functions in Fig. 2 have the same right-sided derivative at  $\lambda$ , but they asymptotically approach identity at different rates.

### III. OGS WITH NON-CONVEX REGULARIZATION

For denoising group-sparse signals in AWGN, we propose to minimize the cost function,  $F : \mathbb{R}^N \rightarrow \mathbb{R}$ ,

$$F(\mathbf{x}) = \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \sum_{i \in \mathcal{Z}} \phi(\|\mathbf{x}_{i,K}\|_2; a) \quad (10)$$

where  $\phi$  is a (non-convex) sparsity promoting penalty function satisfying the assumptions in Sec. II-B, and  $\lambda > 0$ . The notation  $\mathbf{x}_{i,K}$ , defined in (2), denotes a  $K$ -point group starting at index  $i$ . The group size,  $K$  (a positive integer), should be selected based roughly on the size of the groups (clusters) arising in the data. This constitutes one's ‘prior knowledge’

regarding the data to be denoised and may need to be set through some trial-and-error. We note that  $K$  does not impose any strict constraint on the size of groups, not does it define the boundaries of groups.

We note that the minimization of  $F$  is not so straightforward. First, for  $K > 1$ , the variables  $x(i)$  are coupled due to the overlapping group structure of the regularization term. That is to say, the cost function  $F(\mathbf{x})$  can not be written as  $F(\mathbf{x}) = f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2)$  where  $(\mathbf{x}_1, \mathbf{x}_2)$  is a partition of  $\mathbf{x}$ . Consequently, each component  $x^*(i)$  of the minimizer  $\mathbf{x}^*$  depends in general on every data sample  $y(k)$  (albeit the influence diminishes with distance  $|i - k|$ ). Secondly,  $F$  is not differentiable. In particular,  $F$  is generally not differentiable at the minimizer,  $\mathbf{x}^*$ , due to the sparsity of  $\mathbf{x}^*$  induced by the regularizer. (The penalty function,  $\phi$ , is non-differentiable at zero).

In order to leverage convex optimization principles and avoid non-convex optimization issues (local minima, sensitivity to noise, etc.), we seek to restrict  $a$  so that  $F$  is strictly convex. In the following, we address the questions:

- 1) For what values of  $a$  is  $F$  strictly convex?
- 2) When  $F$  is strictly convex, how can the unique minimizer,  $\mathbf{x}^*$ , be efficiently computed?

First, we make a few remarks. If  $K = 1$ , then  $F$  in (10) simplifies to

$$F(\mathbf{x}) = \sum_i \left[ \frac{1}{2} |y(i) - x(i)|^2 + \lambda \phi(x(i); a) \right], \quad (11)$$

the components  $x(i)$  are not coupled, and the minimization of  $F$  amounts to component-wise non-linear thresholding; i.e.,  $x^*(i) = \theta(y(i); a)$ . In this case, the cost function  $F$  does not promote any group structure.

If  $\phi$  is the absolute value function, i.e.,  $\phi(x) = |x|$ , then the cost function  $F$  in (10) is the same cost function considered in our earlier work [13], which considers only convex regularization.

If  $K = 1$  and  $\phi$  is the absolute value function, then the minimizer of  $F$  is given by point-wise soft thresholding of  $\mathbf{y}$ .

The current work addresses the case  $K > 1$  and  $\phi$  a non-convex regularizer, so as to promote group sparsity more strongly in comparison to convex regularization. The enhanced sparsity will be illustrated in Example 1 in Sect. IV-A.

#### A. Group Thresholding

In order to determine the convexity of  $F$ , we first consider a simpler cost function,  $H$ , which consists of a single group. What values of  $a$  ensure that  $H$  is strictly convex?

**Proposition 2.** Consider the function  $H : \mathbb{R}^K \rightarrow \mathbb{R}$ ,

$$H(\mathbf{x}) = \frac{1}{2} \|\bar{\mathbf{y}} - \mathbf{x}\|_2^2 + \lambda \phi(\|\mathbf{x}\|_2; a) \quad (12)$$

where  $\bar{\mathbf{y}} \in \mathbb{R}^K$ ,  $\lambda > 0$ , and  $\phi(\cdot, a) : \mathbb{R} \rightarrow \mathbb{R}$  satisfies the assumptions in Sec. II-B. Then  $H$  is strictly convex if

$$\phi''(0^+; a) > -\frac{1}{\lambda}. \quad (13)$$

*Proof.* Define  $B : \mathbb{R} \rightarrow \mathbb{R}$ ,

$$B(v) = \frac{1}{2} |v|^2 + \lambda \phi(v; a). \quad (14)$$

According to the assumptions on  $\phi$  in Sec. II-B,  $B$  is continuous on  $\mathbb{R}$ , twice differentiable on  $\mathbb{R} \setminus \{0\}$ , and symmetric. Hence, by Lemma A in the appendix, the strict convexity of  $B$  is ensured if its second derivative is positive on  $\mathbb{R} \setminus \{0\}$  and  $B'(0^-) < B'(0^+)$ .

Note that  $B'(0^+) = \lambda \phi'(0^+; a) = \lambda > 0$ ; and by symmetry  $B'(0^-) = \lambda \phi'(0^-; a) = -\lambda < 0$ . Hence  $B'(0^-) < B'(0^+)$ .

To ensure  $B''$  is positive on  $\mathbb{R} \setminus \{0\}$ , we have the condition

$$B''(v) = 1 + \lambda \phi''(v; a) > 0, \quad \text{for } v > 0 \quad (15)$$

or

$$\phi''(v; a) > -\frac{1}{\lambda}, \quad v > 0. \quad (16)$$

Due to assumption 7 in Sec. II-B, we have (13).

Furthermore,  $B(v)$  is increasing on  $\mathbb{R} \geq 0$ . Based on the convexity of  $\|\mathbf{x}\|_2$  and Proposition 2.1.7 of Ref. [30, page 89],  $B(\|\mathbf{x}\|_2)$  is strictly convex.

Expanding  $H(\mathbf{x})$  in (12) shows it is a linear combination of  $B(\|\mathbf{x}\|_2)$  (strictly convex),  $\bar{\mathbf{y}}^T \mathbf{x}$  (convex) and  $\|\bar{\mathbf{y}}\|_2^2$  (constant). Hence,  $H(\mathbf{x})$  is strictly convex.  $\square$

The condition (13) can be used to determine values of  $a$  that ensure strict convexity of  $H$ . For the log, atan, and rational penalty functions ( $\phi_{\log}$ ,  $\phi_{\text{atan}}$ ,  $\phi_{\text{rat}}$ ), we use (7) to obtain the following intervals for  $a$  ensuring strict convexity of  $H$ .

**Corollary 1.** Suppose  $\phi$  is one of the non-convex penalty functions given in Sec. II-B ( $\phi_{\log}$ ,  $\phi_{\text{atan}}$ ,  $\phi_{\text{rat}}$ ). Then  $H$  is strictly convex if and only if

$$0 < a < \frac{1}{\lambda}. \quad (17)$$

Based on a strictly convex function  $H$ , we may define a multivariate threshold/shrinkage function  $\theta : \mathbb{R}^K \rightarrow \mathbb{R}^K$  as in the scalar case (8). It is informative to note the threshold of the multivariate thresholding function.

**Proposition 3.** Define  $\theta : \mathbb{R}^K \rightarrow \mathbb{R}^K$  by

$$\theta(\mathbf{y}) = \arg \min_{\mathbf{x} \in \mathbb{R}^K} \left\{ H(\mathbf{x}) = \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \phi(\|\mathbf{x}\|_2) \right\} \quad (18)$$

where  $\lambda > 0$ , and  $\phi$  satisfies the assumptions in Sec. II-B. Suppose also that  $H$  is strictly convex. If  $\|\mathbf{y}\|_2 < \lambda$ , then the unique minimizer of  $H$  is the zero vector. That is,  $\theta$  is a multivariate threshold function with threshold  $\lambda$ .

*Proof.* The subgradient of the convex function  $H$  is given by  $\partial H(\mathbf{x}) = \mathbf{x} - \mathbf{y} + \lambda \partial \phi(\|\mathbf{x}\|_2)$ . Since  $\phi'(0^+) = 1$  by assumption 5 in Sec. II-B, we have  $\partial \phi(\|\mathbf{0}\|_2) = \partial \{\|\mathbf{0}\|_2\}$ , which is equal to  $\{\mathbf{v} \in \mathbb{R}^K, \|\mathbf{v}\|_2 \leq 1\}$ .

This leads to

$$\partial H(\mathbf{0}) = \{\lambda \mathbf{v} - \mathbf{y} : \|\mathbf{v}\|_2 \leq 1\}. \quad (19)$$

Since  $\mathbf{x}^*$  is a minimizer of  $H$  iff  $\mathbf{0} \in \partial H(\mathbf{x}^*)$  (see [30, Theorem 2.2.1, page 177]), we deduce the following.

- Suppose  $\|\mathbf{y}\|_2 \leq \lambda$ . We can choose  $\mathbf{v} = \mathbf{y}/\lambda$  satisfying  $\|\mathbf{v}\|_2 \leq 1$  such that  $\lambda \mathbf{v} - \mathbf{y} = \mathbf{0}$ . We have  $\mathbf{0} \in \partial H(\mathbf{0})$ , which implies that  $\mathbf{0}$  is the minimizer of  $H$ .
- Suppose  $\|\mathbf{y}\|_2 > \lambda$ . There is no  $\mathbf{v}$  satisfying  $\|\mathbf{v}\|_2 \leq 1$  such that  $\lambda \mathbf{v} - \mathbf{y} = \mathbf{0}$ . Hence,  $\mathbf{0}$  is not the minimizer of  $H$ .

From the arguments above, we conclude that  $\lambda$  defines the threshold of  $\theta$ .  $\square$

When  $\phi$  is the absolute value function, the induced multivariate threshold function  $\theta$  can be expressed in closed form [60]. (Essentially, it performs soft-thresholding on the 2-norm.) A generalization to the case where the data consistency term in (18) is of the form  $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$  has also been addressed [52]. We note that neither [52] nor [60] consider either non-convex regularization or overlapping group sparsity.

If the penalty function,  $\phi$ , is strictly concave on the positive real line (log, atan, etc.), then the induced multivariate threshold function results in less bias of large magnitude components; i.e.,  $\theta(\mathbf{y})$  approaches the identity function for large  $\mathbf{y}$ . An exploration along these lines is given in [57]; however, in that work, the non-convexity was quite mild and not adjustable. (The non-convex regularization in [57] is based on the multivariate Laplace probability density function, which does not have a shape parameter, analogous to  $a$  in the current work.) Furthermore, overlapping group sparsity is not considered in [57].

### B. Overlapping Group Thresholding

Using the results above, we can find a condition on  $a$  to ensure  $F$  in (10) is strictly convex. The result permits the use of non-convex regularization to strongly promote group sparsity while preserving strict convexity of the total cost function,  $F$ .

**Theorem 1.** Consider  $F : \mathbb{R}^N \rightarrow \mathbb{R}$ , defined as

$$F(\mathbf{x}) = \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \sum_i \phi(\|\mathbf{x}_{i,K}\|_2; a) \quad (20)$$

where  $\mathbf{y} \in \mathbb{R}^N$ ,  $K \in \mathbb{Z}_+$ ,  $\lambda > 0$ , and  $\phi(\cdot, a) : \mathbb{R} \rightarrow \mathbb{R}$  satisfies the assumptions in Sec. II-B. Then  $F$  is strictly convex if

$$\phi''(0^+; a) > -\frac{1}{K\lambda}. \quad (21)$$

*Proof.* Write  $F$  as

$$F(\mathbf{x}) = \sum_i F_i(\mathbf{x}_{i,K}) \quad (22)$$

where  $F_i : \mathbb{R}^K \rightarrow \mathbb{R}$  is defined as

$$F_i(\mathbf{v}) = \frac{1}{2K} \|\mathbf{y}_{i,K} - \mathbf{v}\|_2^2 + \lambda \phi(\|\mathbf{v}\|_2; a) \quad (23)$$

for  $i \in \mathbb{Z}$ . Suppose (21) is satisfied. Then by Prop. 2 the functions  $F_i$  are strictly convex. Since  $F$  is a sum of strictly convex functions,  $F$  is strictly convex.  $\square$

**Corollary 2.** Suppose  $\phi$  is one of the penalty functions given in Sec. II-B ( $\phi_{\log}$ ,  $\phi_{\text{atan}}$ ,  $\phi_{\text{rat}}$ ). Then  $F$  is strictly convex if

$$0 < a < \frac{1}{K\lambda}. \quad (24)$$

We give some practical comments on using (24) to set the parameters  $\{K, \lambda, a\}$ . We suggest that  $K$  be chosen first, based on the structural properties of the signal to be denoised. We suggest that  $a$  then be set to a fixed fraction of its maximal value; i.e., fix  $\beta \in [0, 1]$  and set  $a = \beta/(K\lambda)$ . So, we consider  $a$  as a function of  $\lambda$ . We then set  $\lambda$  according to the noise variance. In Sec. III-E, we describe two approaches for the selection of  $\lambda$ . In our numerical experiments on speech enhancement, we have found that setting  $a$  to its maximal value of  $1/(K\lambda)$  generally yields the best results; i.e.,  $\beta = 1$ . Hence, in the examples in Sec. IV, we set  $a$  to its maximal value.

Equation (24) may suggest the proposed method becomes ineffective for large  $K$ . It can be noted from (24) that for large  $K$ ,  $a\lambda$  should be small ( $< 1/K$ ). If  $\lambda$  is set so as to achieve a desired degree of noise suppression, then (24) implies  $a$  should be small. A small  $a$ , in turn, limits the non-convexity of the regularizer. Hence, it appears the benefit of the proposed non-convex regularization method is diminished for large  $K$ . However, two considerations offset this reasoning. First, for larger  $K$ , a smaller value of  $\lambda$  is needed so as to achieve a fixed level of noise suppression (this can be seen, for example, in Table III). Secondly, for larger  $K$ , there is greater overlap between adjacent groups because the groups are fully-overlapping; so, regularization may be more sensitive to  $a$ .

### C. Minimization Algorithm

To derive an algorithm minimizing the strictly convex function  $F$  in (10), we use the majorization-minimization (MM) procedure [24] as in [13]. The MM procedure replaces a single minimization problem by a sequence of (simpler) ones. Specifically, MM is based on the iteration

$$\mathbf{x}^{(k+1)} = \arg \min_{\mathbf{x}} Q(\mathbf{x}, \mathbf{x}^{(k)}) \quad (25)$$

where the function,  $Q : \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}$ , is a majorizer (upper bound) of  $F$  and  $k$  is the iteration index. For  $Q$  to be a majorizer of  $F$  it should satisfy

$$Q(\mathbf{x}, \mathbf{v}) \geq F(\mathbf{x}), \quad \forall \mathbf{x} \in \mathbb{R}^N \quad (26)$$

$$Q(\mathbf{v}, \mathbf{v}) = F(\mathbf{v}). \quad (27)$$

The MM procedure monotonically reduces the cost function at each iteration; i.e.,  $F(\mathbf{x}^{(k)})$  is a decreasing sequence.

We do not know a proof that  $\mathbf{x}^{(k)}$  converges to the minimizer of  $F$  when  $F$  is non-differentiable, as is the case here. However, it is suggested in [24] (Sec. III) that this class of algorithms converges with ‘probability one.’ Similar remarks are made in [24] (Sec. V-D). We also note that  $F$  in (10) can be approximated by a differentiable function by substituting  $(\|\mathbf{x}_{i,K}\|_2^2 + \epsilon)^{1/2}$  for  $\|\mathbf{x}_{i,K}\|_2$  and that in this case, the MM algorithm is ensured to converge. However, we do not make this differentiable approximation here because we observe reliable convergence in practice without it.

To specify a majorizer of the cost function  $F$  in (10), we first specify a majorizer of the penalty function,  $\phi$ . To simplify notation, we suppress the dependence of  $\phi$  on  $a$ .

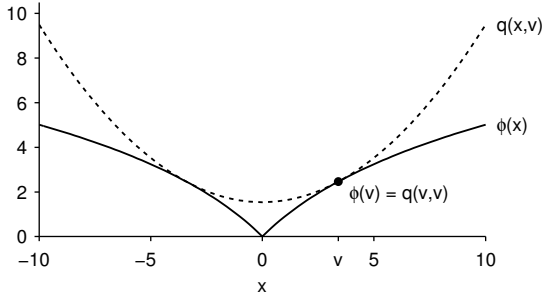


Fig. 3. Majorization of non-convex  $\phi(x)$  by  $q(x, v)$ .

**Lemma 1.** Assume  $\phi : \mathbb{R} \rightarrow \mathbb{R}$  satisfies the assumptions in Sec. II-B. Then  $q : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ , defined by

$$q(x, v) = \frac{1}{2v} \phi'(v) x^2 + \phi(v) - \frac{v}{2} \phi'(v), \quad (28)$$

is a majorizer of  $\phi$  except for  $v = 0$ , i.e.,

$$q(x, v) \geq \phi(x), \quad \forall x \in \mathbb{R}, \quad \forall v \in \mathbb{R} \setminus \{0\} \quad (29)$$

$$q(v, v) = \phi(v), \quad \forall v \in \mathbb{R} \setminus \{0\} \quad (30)$$

The majorization of  $\phi(x)$  by  $q(x, v)$  is illustrated in Fig. 3.

*Proof.* By direct substitution, one may verify (30). We now show (29). Let  $v > 0$  and  $x \geq 0$ . Using Taylor's Theorem [55, Theorem 5.15], we have

$$\phi(x) = \phi(v) + \phi'(v)(x - v) + \frac{\phi''(v_0)}{2}(x - v)^2 \quad (31)$$

for some  $v_0$  between  $x$  and  $v$ . By the assumptions on  $\phi$ , we have  $\phi''(v_0) < 0$ . Hence from (31),

$$\phi(x) \leq \phi(v) + \phi'(v)(x - v). \quad (32)$$

Note that  $(x - v)^2 \geq 0$  implies

$$x \leq \frac{1}{2v} x^2 + \frac{v}{2}. \quad (33)$$

Using (33) for  $x$  on the left-hand side of (32) gives

$$\phi(x) \leq \phi(v) + \phi'(v) \left( \frac{1}{2v} x^2 + \frac{v}{2} - v \right). \quad (34)$$

Recognizing that the right-hand side of (34) is  $q(x, v)$ , we obtain  $\phi(x) \leq q(x, v)$  for all  $x \geq 0$ ,  $v > 0$ . By symmetry of  $q$  and  $\phi$ , we obtain (29).  $\square$

Since  $q$  is a majorizer of  $\phi$ , the function

$$Q(\mathbf{x}, \mathbf{v}) = \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \sum_i q(\|\mathbf{x}_{i,K}\|_2, \|\mathbf{v}_{i,K}\|_2) \quad (35)$$

is a majorizer of  $F$ . Using (28), the function  $Q$  is given by

$$Q(\mathbf{x}, \mathbf{v}) = \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 + \frac{\lambda}{2} \sum_i \frac{\phi'(\|\mathbf{v}_{i,K}\|_2)}{\|\mathbf{v}_{i,K}\|_2} \|\mathbf{x}_{i,K}\|_2^2 + C$$

where  $C$  does not depend on  $\mathbf{x}$ . After algebraic manipulations,  $Q$  can be expressed as

$$Q(\mathbf{x}, \mathbf{v}) = \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 + \frac{\lambda}{2} \sum_i r(i; \mathbf{v}) x^2(i) + C \quad (36)$$

TABLE I  
OVERLAPPING GROUP SHRINKAGE (OGS) WITH PENALTY  $\phi$ .

input:  $\mathbf{y} \in \mathbb{R}^N$ ,  $\lambda > 0$ ,  $K$ ,  $\phi$

$\mathbf{x} = \mathbf{y}$  (initialization)

$S = \{i \in \mathbb{Z}_N : y(i) \neq 0\}$

repeat

$$a(i) = \left[ \sum_{k=0}^{K-1} |x(i+k)|^2 \right]^{1/2}, \quad i \in S$$

$$b(i) = \frac{\phi'(a(i))}{a(i)}, \quad i \in S$$

$$r(i) = \sum_{j=0}^{K-1} b(i-j), \quad i \in S$$

$$x(i) = \frac{y(i)}{1 + \lambda r(i)}, \quad i \in S$$

$$S = \{i \in \mathbb{Z}_N : |x(i)| > \epsilon\} \quad (*)$$

until convergence

return:  $\mathbf{x}$

(\*) For finite precision implementations.

where  $r : \mathbb{Z} \times \mathbb{R}^K \rightarrow \mathbb{R}$  is defined as

$$r(i; \mathbf{v}) = \sum_{j=0}^{K-1} \frac{\phi'(\|\mathbf{v}_{i-j,K}\|_2)}{\|\mathbf{v}_{i-j,K}\|_2}. \quad (37)$$

Note that the components  $x(i)$  in (36) are uncoupled. Furthermore,  $Q$  is quadratic in  $x(i)$ . Hence, the minimizer of  $Q$  with respect to  $\mathbf{x}$  is easily obtained. The quantities  $r(i, \mathbf{v})$  in (37) are readily computed;  $r$  is essentially a double  $K$ -point convolution, with a nonlinearity between the two convolutions.

Using (36) in the MM iteration (25), we obtain

$$x^{(k+1)}(i) = \frac{y(i)}{1 + \lambda r(i; \mathbf{x}^{(k)})}, \quad i \in \mathbb{Z}_N, \quad (38)$$

where  $r$  is given by (37). This constitutes the OGS algorithm. The algorithm is summarized in Table I. We denote the output of the OGS algorithm as  $\mathbf{y} = \text{ogs}(\mathbf{x}; \lambda, K, \phi)$ .

In case the denominator in (37) is zero, we assign positive infinity to  $r(i, \mathbf{v})$ . Since  $r(i; \mathbf{x}^{(k)})$  arises in the denominator of (38), a value of  $+\infty$  for this variable, in the event it occurs, leads to a value of zero for  $x^{(k+1)}(i)$ .

Note that  $q$  in (28) is undefined if  $v = 0$ . This singularity issue often arises when a quadratic function is used to majorize a non-smooth function [24], [49]. This issue may manifest itself in the OGS algorithm whenever a  $K$ -point group of  $\mathbf{x}$  is equal to the  $K$ -point zero vector; i.e., if  $\mathbf{x}_{i,K}^{(k)} = \mathbf{0} \in \mathbb{R}^K$  for some index  $i$  and iteration  $k$ . In the event of such an occurrence, the OGS algorithm would encounter a 'divide-by-zero' error. However, such an occurrence is guaranteed not to occur with suitable initialization, as described in [13]. For example, it is sufficient to initialize all  $x(i)$  to non-zero values, i.e.,  $x^{(0)}(i) \neq 0$  for all  $i \in \mathbb{Z}_N$ . With such an initialization, it is readily observed that  $r(i; \mathbf{x}^{(k)})$  in the denominator of (38) is

strictly positive and finite and that  $x^{(k)}(i) \neq 0$  for all  $i \in \mathbb{Z}_N$  and all iterations  $k$ . When some components of the solution  $\mathbf{x}^*$  are zero (as expected, due to sparse regularization), those values  $x^{(k)}(i)$  approach zero in limit; i.e.,  $x^{(k)}(i) \rightarrow 0$  as  $k \rightarrow \infty$ .

We propose initializing  $\mathbf{x}$  to  $\mathbf{y}$ ; i.e.,  $\mathbf{x}^{(0)} = \mathbf{y}$ , and we exclude from the iteration (38) those  $i$  for which  $y(i) = 0$ . The set  $S \subset \mathbb{Z}_N$  in Table I serves to exclude these components from the iterative update. In this case,  $x^{(k)}(i) = 0$  for all iterations  $k$ , which is justified by part 1) of Lemma B in the appendix. As a consequence of Lemma B, initializing  $x^{(0)}(i)$  to zero for  $i \notin S$  is optimal. Therefore, the algorithm excludes these values from the update procedure because they are already optimal.

With the initialization  $\mathbf{x}^{(0)} = \mathbf{y}$ , it is readily observed, as above, that  $r(i; \mathbf{x}^{(k)})$  in the denominator of (38) is strictly positive and finite and that  $x^{(k)}(i) \neq 0$  for all  $i \in S$  and all iterations  $k$ . Assuming infinite precision, it is sufficient to define  $S$  prior to the loop only; the last line in Table I, indicated by (\*), can be omitted. It is guaranteed that a division by zero will never occur, as discussed above.

The OGS algorithm proceeds by iteratively reducing  $x(i)$ ,  $i \in S$ , toward their optimal values (including zero). The attenuation is multiplicative, so the value never equals zero, even though it may converge to zero. But if a value reaches ‘machine epsilon’ then a divide-by-zero error may subsequently occur in the implementation. Hence, to avoid possible divide-by-zero errors due to finite precision arithmetic, the OGS algorithm updates  $S$  at the end of the loop in Table I. The small number,  $\epsilon$ , may be set to ‘machine epsilon’, which for single precision floating point is about  $10^{-16}$ .

We do not prove the convergence of the OGS algorithm to the minimizer of  $F$  due to the complication of the singularity issue. However, due to its derivation based on the majorization-minimization principle, OGS is guaranteed to decrease the cost function at each iteration. Moreover, in practice, we have observed through extensive numerical investigation, that the algorithm has the same rapid convergence behavior as convex regularized OGS [13].

Note that in the OGS algorithm, summarized in Table I, the penalty function appears in only one place: the computation of  $b(i)$ . It can therefore be observed that the role of the penalty is encapsulated by the function  $\phi'(u)/u$ . Table II lists this function for the penalty functions given in Sec. II-B. The function  $\phi'(u)/u$  have very similar functional forms. The similarity of these functions reveal the close relationship among the listed penalty functions.

#### D. The Multidimensional Case

The results and algorithm described in the preceding sections can be extended to the multidimensional case straightforwardly. In the numerical experiments below, we use a two-dimensional version of the algorithm in order to denoise the time-frequency spectrogram of a noisy speech waveform.

Suppose  $\mathbf{x}$  is a 2D array of size  $N_1 \times N_2$ ; i.e.,

$$\mathbf{x} = \{x(i_1, i_2), 0 \leq i_1 \leq N_1 - 1, 0 \leq i_2 \leq N_2 - 1\}.$$

TABLE II  
SPARSE PENALTIES AND CORRESPONDING NONLINEARITIES

penalty	$\phi(u)$	$\phi'(u)/u$
abs	$ u $	$\frac{1}{ u }$
log	$\frac{1}{a} \log(1 + a u )$	$\frac{1}{ u (1 + a u )}$
atan	$\frac{2}{a\sqrt{3}} \left( \tan^{-1} \left( \frac{1 + 2a u }{\sqrt{3}} \right) - \frac{\pi}{6} \right)$	$\frac{1}{ u (1 + a u  + a^2 u ^2)}$
rational	$\frac{ u }{1 + a u /2}$	$\frac{1}{ u (1 + a u )^2}$

The array can be expressed using multi-indices as

$$\mathbf{x} = \{x(i), i \in \mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2}\}.$$

Let  $K = (K_1, K_2)$  denote the size of a 2D group. Then a sub-group of size  $K$  can be expressed as

$$\mathbf{x}_{i,K} = \{x(i + j), j \in \mathbb{Z}_{K_1} \times \mathbb{Z}_{K_2}\}.$$

In the two-dimensional case, the function  $F$  in (10) is

$$F(\mathbf{x}) = \sum_{i \in \mathbb{Z}^2} \frac{1}{2} |y(i) - x(i)|^2 + \lambda \phi(\|\mathbf{x}_{i,K}\|_2; a), \quad (39)$$

and conditions (21) and (24) become

$$\phi''(0^+; a) > -\frac{1}{K_1 K_2 \lambda} \quad (40)$$

and

$$0 < a < \frac{1}{K_1 K_2 \lambda} \quad (41)$$

respectively. The algorithm in Table I is essentially the same for the two-dimensional case. The summations become double summations, etc. Extensions to higher dimensional signals are similarly straightforward.

#### E. Regularization Parameter Selection

**Noise level suppression.** The regularization parameter,  $\lambda$ , can be selected using existing generic techniques such as the L-curve method. However, in [13] we described an approach to set  $\lambda$  based directly on the standard deviation,  $\sigma$ , of the AWGN, which we assume is known. This approach seeks to preserve one of the concepts of scalar thresholding (e.g., hard or soft thresholding), namely the processing of signal values based on relative magnitude. Consider the problem of estimating a sparse signal in AWGN. If many of the non-zero values of the sparse signal exceed the noise floor, then a suitable threshold value,  $T$ , should exceed the noise floor. But  $T$  should not be too large, or else the non-zero values of the sparse signal will be annihilated. Hence, it is reasonable to use the value  $T = 3\sigma$ . This threshold will set most of noise (about 99.7%) to zero. (If the sparse signal has non-zero values less than  $T$  in magnitude, then those values will be lost.)

The simplicity of the ‘three-sigma’ rule can not be leveraged so easily in the proposed OGS algorithm. However, we can still implement the concept of setting  $\lambda$  so as to reduce the

TABLE III  
OGS REGULARIZATION PARAMETER WITH PENALTY  $\phi(\cdot) = \phi_{\text{atan}}(\cdot, 1/(K_1 K_2 \lambda))$  AND 25 ITERATIONS

$K$	$\lambda, \alpha(\lambda, K, \phi)$					
$1 \times 1$	4.25, $1.00 \cdot 10^{-2}$	4.59, $4.33 \cdot 10^{-3}$	4.93, $1.51 \cdot 10^{-3}$	5.27, $4.05 \cdot 10^{-4}$	5.61, $1.00 \cdot 10^{-4}$	
$1 \times 2$	2.14, $1.00 \cdot 10^{-2}$	2.31, $4.35 \cdot 10^{-3}$	2.48, $1.49 \cdot 10^{-3}$	2.64, $3.99 \cdot 10^{-4}$	2.81, $1.00 \cdot 10^{-4}$	
$1 \times 3$	1.45, $1.00 \cdot 10^{-2}$	1.56, $4.52 \cdot 10^{-3}$	1.68, $1.56 \cdot 10^{-3}$	1.79, $4.06 \cdot 10^{-4}$	1.91, $1.00 \cdot 10^{-4}$	
$1 \times 4$	1.11, $1.00 \cdot 10^{-2}$	1.20, $4.47 \cdot 10^{-3}$	1.29, $1.58 \cdot 10^{-3}$	1.38, $4.11 \cdot 10^{-4}$	1.47, $1.00 \cdot 10^{-4}$	
$1 \times 5$	0.91, $1.00 \cdot 10^{-2}$	0.98, $4.37 \cdot 10^{-3}$	1.05, $1.55 \cdot 10^{-3}$	1.13, $4.07 \cdot 10^{-4}$	1.20, $1.00 \cdot 10^{-4}$	
$2 \times 2$	1.08, $1.00 \cdot 10^{-2}$	1.16, $4.37 \cdot 10^{-3}$	1.24, $1.47 \cdot 10^{-3}$	1.33, $3.95 \cdot 10^{-4}$	1.41, $1.00 \cdot 10^{-4}$	
$2 \times 3$	0.73, $1.00 \cdot 10^{-2}$	0.79, $4.41 \cdot 10^{-3}$	0.85, $1.49 \cdot 10^{-3}$	0.90, $3.96 \cdot 10^{-4}$	0.96, $1.00 \cdot 10^{-4}$	
$2 \times 4$	0.56, $1.00 \cdot 10^{-2}$	0.61, $4.18 \cdot 10^{-3}$	0.65, $1.44 \cdot 10^{-3}$	0.70, $3.91 \cdot 10^{-4}$	0.74, $1.00 \cdot 10^{-4}$	
$2 \times 5$	0.47, $1.00 \cdot 10^{-2}$	0.50, $3.89 \cdot 10^{-3}$	0.54, $1.33 \cdot 10^{-3}$	0.58, $3.74 \cdot 10^{-4}$	0.61, $1.00 \cdot 10^{-4}$	
$3 \times 3$	0.50, $1.00 \cdot 10^{-2}$	0.54, $4.11 \cdot 10^{-3}$	0.58, $1.38 \cdot 10^{-3}$	0.62, $3.81 \cdot 10^{-4}$	0.66, $1.00 \cdot 10^{-4}$	
$3 \times 4$	0.40, $1.00 \cdot 10^{-2}$	0.43, $3.57 \cdot 10^{-3}$	0.46, $1.19 \cdot 10^{-3}$	0.49, $3.51 \cdot 10^{-4}$	0.51, $1.00 \cdot 10^{-4}$	
$3 \times 5$	0.34, $1.00 \cdot 10^{-2}$	0.36, $3.26 \cdot 10^{-3}$	0.39, $1.04 \cdot 10^{-3}$	0.41, $3.23 \cdot 10^{-4}$	0.43, $1.00 \cdot 10^{-4}$	
$4 \times 4$	0.33, $1.00 \cdot 10^{-2}$	0.35, $3.24 \cdot 10^{-3}$	0.37, $1.02 \cdot 10^{-3}$	0.39, $3.16 \cdot 10^{-4}$	0.41, $1.00 \cdot 10^{-4}$	
$4 \times 5$	0.29, $1.00 \cdot 10^{-2}$	0.30, $3.09 \cdot 10^{-3}$	0.32, $9.61 \cdot 10^{-4}$	0.33, $3.04 \cdot 10^{-4}$	0.35, $1.00 \cdot 10^{-4}$	
$5 \times 5$	0.25, $1.00 \cdot 10^{-2}$	0.26, $3.05 \cdot 10^{-3}$	0.28, $9.44 \cdot 10^{-4}$	0.29, $3.01 \cdot 10^{-4}$	0.30, $1.00 \cdot 10^{-4}$	
$2 \times 8$	0.33, $1.00 \cdot 10^{-2}$	0.35, $3.33 \cdot 10^{-3}$	0.37, $1.05 \cdot 10^{-3}$	0.39, $3.22 \cdot 10^{-4}$	0.41, $1.00 \cdot 10^{-4}$	

noise down to a specified fraction of its original power. For this purpose, the effect of the OGS algorithm on pure zero-mean Gaussian noise,  $x(i) = \mathcal{N}(0, \sigma^2)$ , can be measured through computation. In particular, the standard deviation of the OGS output as a function of  $(\lambda, K, \phi)$  can be found empirically and recorded. For example, Table III records the value

$$\alpha(\lambda, K, \phi) = \frac{1}{\sigma} \text{std}\{\text{ogs}(\mathbf{x}; \lambda, K, \phi)\}, \quad x(i) = \mathcal{N}(0, \sigma^2)$$

for several  $\lambda$  and group sizes  $K$ . For this table we used the atan penalty function with  $a$  set to its maximum value of  $1/(K\lambda)$ ; i.e.,  $\phi(\cdot) = \phi_{\text{atan}}(\cdot, 1/(K\lambda))$ . The value  $\alpha$  also depends on the number of iterations of the OGS algorithm. In computing Table III we have used a fixed number of 25 iterations.

We clarify how to use Table III to set the regularization parameter: Suppose in one-dimensional signal denoising, one seeks to set  $\lambda$  so that the OGS algorithm reduces  $\sigma$  down to  $10^{-4}\sigma$ . If one uses a group size of  $K = 5$ , the atan penalty function with  $a = 1/(5\lambda)$ , and 25 iterations, then according to Table III, one should use  $\lambda = 1.2\sigma$  (see the last column of the fifth row of the table). For each group size  $K$ , the table records a discrete set of  $(\lambda, \alpha)$  pairs for  $10^{-4} < \alpha < 10^{-2}$ . Linear interpolation on a  $\alpha$ -logarithmic scale can be used to estimate  $\lambda$  for other  $\alpha$ . For example, if one seeks to set  $\lambda$  so that the OGS algorithm reduces  $\sigma$  down to  $10^{-3}\sigma$ , then according to the interpolation illustrated in Fig. 4, one should use  $\lambda = 1.07\sigma$ .

To set  $\lambda$  by this approach for other penalty functions, other values of  $a$ , and for complex data, it is necessary to compute additional tables. We have precomputed a set of such tables to be available as supplementary material. Using precomputed tables and interpolation, a suitable value for  $\lambda$  can be found very quickly. These tables assume the noise is AWGN; for other noise models, other tables need to be precomputed. This approach is also effective for two-dimensional denoising (e.g., spectrogram denoising).

**Monte-Carlo SURE.** Another approach to select the regularization parameter,  $\lambda$ , is based on minimizing the mean square

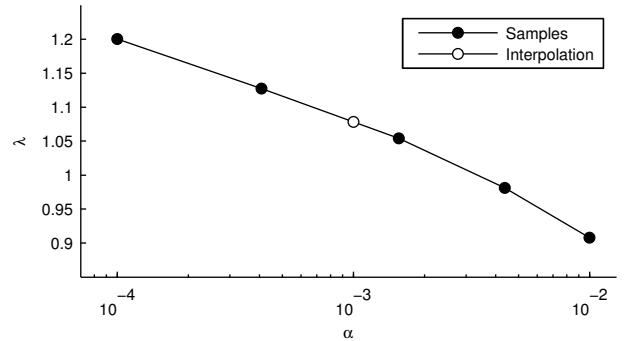


Fig. 4. Solid dots indicate the values from Table III for the group size  $K = 5$ . The circle indicates the interpolated value at  $\alpha = 10^{-3}$ .

error (MSE). For the problem of denoising a signal in AWGN, the MSE is unknown in practice, due to the noise-free signal being unknown. But, the MSE can be estimated using Stein's unbiased risk estimator (SURE) [62]. To estimate the MSE, SURE requires only the observation  $\mathbf{y}$ , noise variance  $\sigma^2$ , and divergence of the estimator. However, the computation of the divergence is intractable for many estimators, including OGS. To overcome this issue, it is proposed in Ref. [53] that Monte-Carlo methods be used. We have applied this approach, i.e., 'Monte-Carlo SURE' (MC-SURE), to estimate the MSE for complex-valued speech spectrogram denoising using OGS. Since the spectrogram is complex, we calculate the MS-SURE MSE by averaging real and imaginary divergences as in [9]. Figure 5 illustrates both the MSE, as calculated by MC-SURE, and the true MSE, as functions of  $\lambda$ . The estimated MSE is quite accurate, and the MSE-optimal value of  $\lambda$  is about 0.33. However, a disadvantage of MC-SURE is its high computational complexity. It requires two OGS optimizations for each  $\lambda$  to emulate the divergence.

It is noted in Ref. [53] that for non-smooth estimators, the MSE, as calculated by MC-SURE, tends to deviate randomly from the true MSE (see Fig. 4 in [53]). For OGS, the MSE



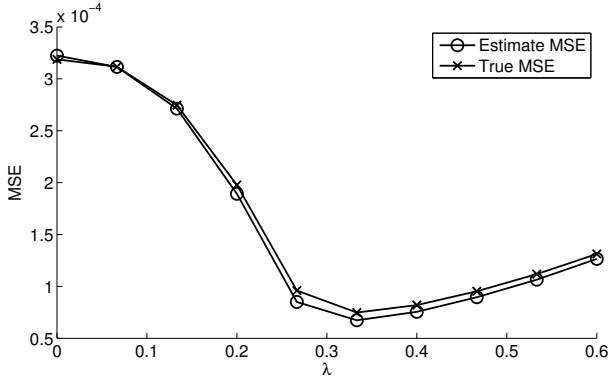


Fig. 5. True MSE and MSE calculated using Monte-Carlo SURE.

TABLE IV  
EXAMPLE 1. OUTPUT SNR

Param.	Estimator				
	Hard thr.	Soft thr.	OGS[abs]	OGS[log]	OGS[atan]
max SNR	13.84	12.17	12.30	14.52	15.37
$10^{-2}\sigma$	6.74	3.86	8.01	12.07	13.92
$10^{-3}\sigma$	5.05	2.17	6.23	9.69	11.54

SNR is in dB;  $\sigma$  is the noise standard deviation.

calculated by MC-SURE closely follows the true MSE, as illustrated in Fig 5, consistent with OGS being a continuous, bounded estimator.

## IV. EXPERIMENTAL RESULTS

### A. Example 1: One-dimensional Signal Denoising

This example compares the proposed non-convex regularized OGS algorithm with the earlier (convex regularized) version of OGS and with scalar thresholding. The SNRs are summarized in Table IV.

Figure 6a shows a synthetic group-sparse signal (same as in [13]). The noisy signal, shown in Fig. 6b, was obtained by adding white Gaussian noise (AWGN) with SNR of 10 dB. For each of soft and hard thresholding, we used the threshold,  $T$ , that maximizes the SNR. The SNR values are summarized in the top row of Table IV.

The result obtained using the prior version of OGS [13] is shown in Fig 6c. This is equivalent to setting  $\phi$  to the absolute value function; i.e.  $\phi(x) = |x|$ . So, we denote this as OGS[abs]. The result using the proposed non-convex regularized OGS is shown in Fig. 6d. We use the arctangent penalty function with  $a$  set to the maximum value of  $1/(K\lambda)$  that preserves convexity of  $F$ ; i.e., we use  $\phi(\cdot) = \phi_{\text{atan}}(\cdot, 1/(K\lambda))$ . We denote this as OGS[atan]. We also used the logarithmic penalty (not shown in the figure). For each version of OGS, we used a group size of  $K = 5$ , and we set  $\lambda$  to maximize the SNR.

Comparing soft thresholding and OGS[abs] (both of which are based on convex regularization), it can be observed that OGS[abs] gives a higher SNR, but only marginally. Both methods leave residual noise, as can be observed for OGS[abs] in Fig. 6c. On the other hand, comparing OGS[atan] and

OGS[abs], it can be observed that OGS[atan] (based on non-convex regularization) is substantially superior: it has a substantially higher SNR and almost no residual noise is visible in the denoised signal. Comparing OGS[log] and OGS[atan] with hard thresholding (see Table IV), it can be observed the new non-convex regularized OGS algorithm also yields higher SNR than hard thresholding. This example demonstrates the effectiveness of non-convex regularization for promoting group sparsity.

To more clearly compare the result of OGS[abs] and OGS[atan], these two results are shown together in Fig. 7. In Fig. 7a, the output value,  $x(i)$ , is shown versus the input value,  $y(i)$ , for  $i \in \mathbb{Z}_N$ . Compared to OGS[abs], the OGS[atan] algorithm better preserves the amplitude of the non-zero values of the original signal, while better thresholding small values. Figure 7b shows the denoising error for the two OGS methods. It can be observed that the denoised signal produced by OGS[atan] has much less error than OGS[abs]. (For OGS[atan], the error is essentially zero for 50% of the signal values.)

As a second experiment, we selected  $T$  and  $\lambda$  for each method, so as to reduce the noise standard deviation,  $\sigma$ , down to  $0.01\sigma$ , as described in Sec. III-E. The resulting SNRs, given in the second row of Table IV, are much lower. (This method does not maximize SNR, but it does ensure residual noise is reduced to the specified level.) The low SNR in these cases is due to the attenuation (bias) of large magnitude values. However, it can be observed that OGS, especially with non-convex regularization, significantly outperforms scalar thresholding.

### B. Example 2: Speech Denoising

This example evaluates the use of the proposed OGS algorithm for the problem of speech enhancement (denoising). We compare the OGS algorithm with several other algorithms. For the evaluation, we use female and male speakers, multiple sentences, two noise levels, and two sampling rates.

Let  $\mathbf{s} = \{s(n), n \in \mathbb{Z}_N\}$  denote the noisy speech waveform and  $\mathbf{y} = \{y(i), i \in \mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2}\} = \text{STFT}\{\mathbf{s}\}$  denote the complex-valued short-time Fourier transform of  $\mathbf{s}$ . For speech enhancement, we apply the two-dimensional form of the OGS algorithm to  $\mathbf{y}$  and then compute the inverse STFT; i.e.,

$$\mathbf{x} = \text{STFT}^{-1}\{\text{ogs}(\text{STFT}\{\mathbf{s}\}; \lambda, K, \phi)\}$$

with  $K = (K_1, K_2)$  where  $K_1$  and  $K_2$  are the spectral and temporal widths of the two-dimensional group. We implement the STFT with 50% frame overlap and a frame duration of 32 milliseconds (e.g., 512 samples at sampling rate 16 kHz).

Throughout this example, we use the non-convex arctangent penalty function with  $a$  set to its maximum value of  $a = 1/(K_1 K_2 \lambda)$ . In all cases, we use a fixed number of 25 iterations within the OGS algorithm.

Each sentence in the evaluation is spoken by both a male and a female speaker. There are 15 sentences sampled at 8 kHz, and 30 sentences sampled at 16 kHz. The 8 kHz and 16 kHz signals were obtained from Ref. [38] and a Carnegie

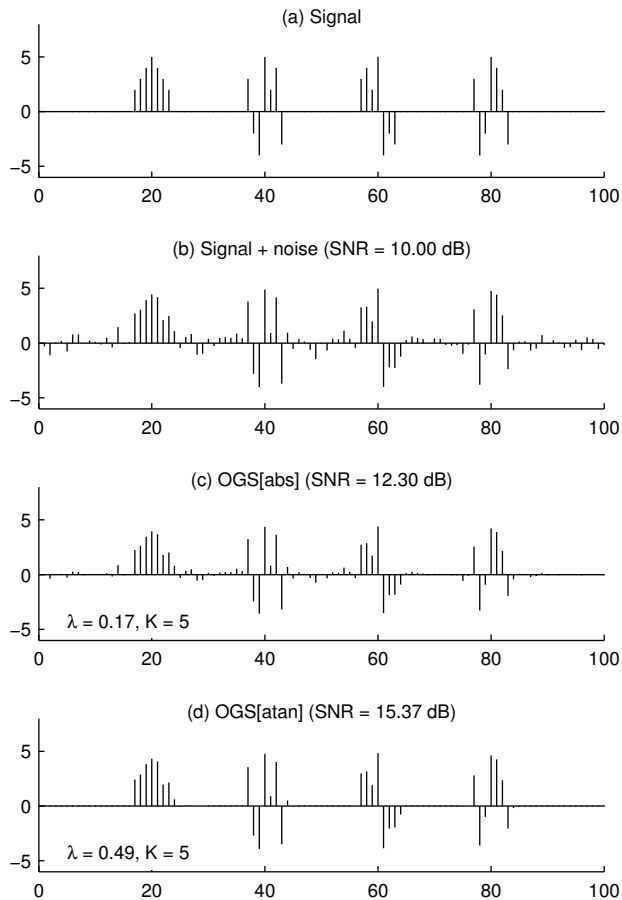


Fig. 6. Example 1: Group-sparse signal denoising.

Mellon University (CMU) website, respectively.<sup>1</sup> To simulate noisy speech, we added white Gaussian noise.

The time-frequency spectrogram of a noisy speech signal (arctic\_a0001) with an SNR of 10 dB is illustrated in Fig. 8a. Figure 8b illustrates the result of OGS[atan] using group size  $K = (8, 2)$ ; i.e., eight spectral samples by two temporal samples. It can be observed that noise is effectively suppressed while details are preserved.

Figure 9 compares the proposed OGS[atan] algorithm with the prior version of OGS [13], i.e., OGS[abs]. The figure shows a single frame of the denoised spectrograms, corresponding to  $t = 0.79$  seconds. The prior and proposed OGS algorithms are illustrated in parts (a) and (b) respectively. In both (a) and (b), the noise-free spectrogram, to be recovered, is indicated in gray. (The noisy spectrogram is not illustrated). Comparing (a) and (b), it can be observed that above 2 kHz, OGS[atan] estimates the noise-free spectrum more accurately than OGS[abs].

In terms of run-time, for a signal of length  $N = 51761$  (i.e., 3.2 seconds at sampling rate of 16 kHz), algorithms OGS[abs] and OGS[atan] ran in 0.18 and 0.22 seconds, respectively.

<sup>1</sup>The CMU files were downloaded from [http://www.speech.cs.cmu.edu/cmu\\_arctic/cmu\\_us\\_bdl\\_arctic/wav](http://www.speech.cs.cmu.edu/cmu_arctic/cmu_us_bdl_arctic/wav) and [http://www.speech.cs.cmu.edu/cmu\\_arctic/cmu\\_us\\_clb\\_arctic/wav](http://www.speech.cs.cmu.edu/cmu_arctic/cmu_us_clb_arctic/wav). This evaluation used files arctic\_a0001 - arctic\_a0030.

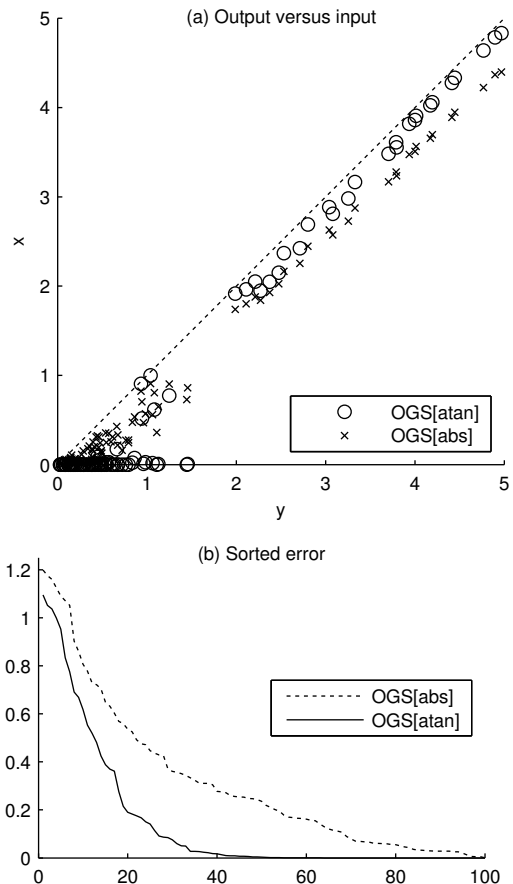


Fig. 7. Example 1. Comparison of OGS[abs] and OGS[atan] in Fig. 6.

Timings were performed on a 2013 MacBook Pro (2.5 GHz Intel Core i5) running Matlab R2011a.

**Regularization parameter.** We have found empirically, that setting  $\lambda$  to maximize SNR yields speech with noticeable undesirable perceptual artifacts (‘musical noise’). This known phenomenon is due to residual noise in the STFT domain. Therefore, we instead set the regularization parameter,  $\lambda$ , using the noise suppression approach described in Sec. III-E. In particular, we set  $\lambda$  so as to reduce the noise standard deviation  $\sigma$  down to  $(3 \times 10^{-4})\sigma$ . We have selected this value so as to optimize the perceptual quality of the denoised speech according to informal listening tests. In particular, this value is effective at suppressing the ‘musical noise’ artifact. We also note that this approach leads to greater regularization (higher  $\lambda$ ) than SNR-optimization of  $\lambda$ .

**Group size.** The perceptual quality of speech denoised using OGS depends on the specified group size. As we apply OGS to a time-frequency spectrogram, the size of the group with respect to both the temporal and spectral dimensions must be specified. We let  $K_1$  and  $K_2$  denote the number of spectral and temporal samples, respectively.

One approach to select the pair of parameters,  $(K_1, K_2)$ , is to maximize the SNR for a set of denoising experiments. We have performed OGS denoising for each of 30 noisy speech signals using all pairs  $(K_1, K_2)$  such that  $1 \leq K_1 \leq 10$  and

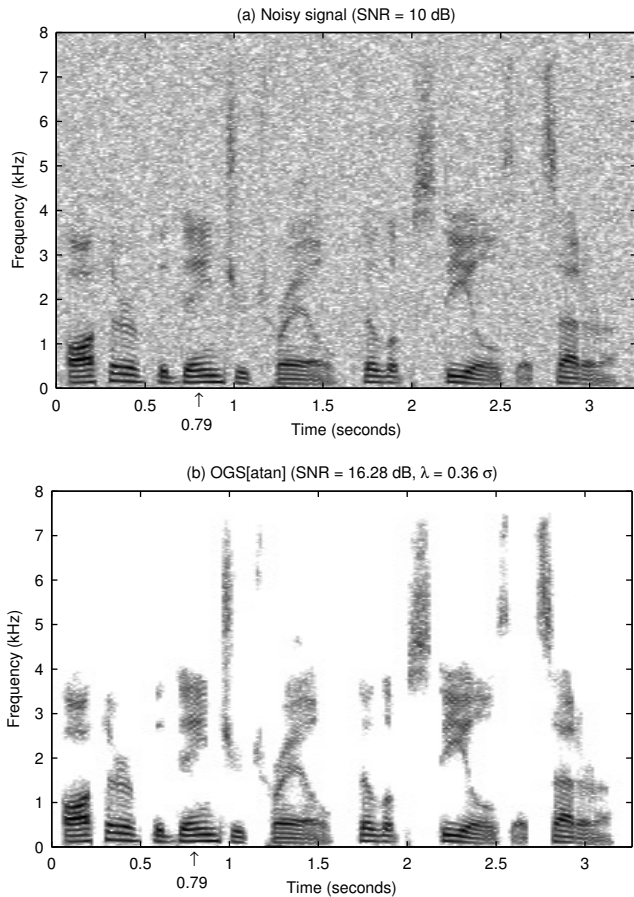


Fig. 8. Spectrograms before and after denoising (male speaker). (a) Noisy signal. (b) OGS[atan] with group size  $K = (8, 2)$ . Gray scale represents decibels.

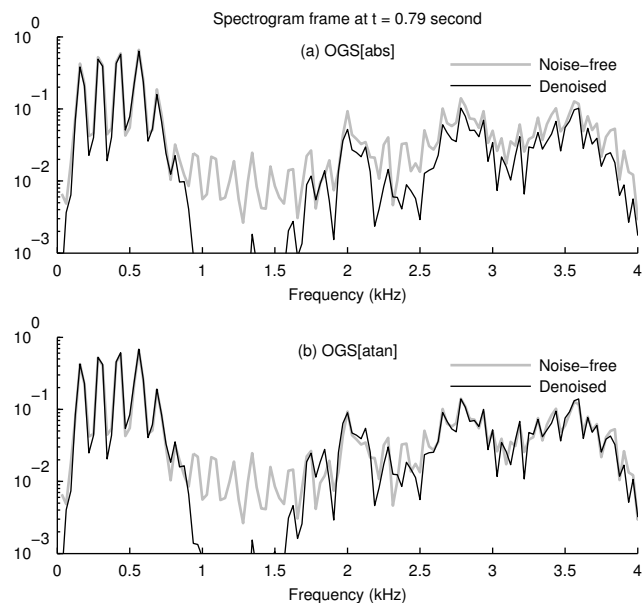


Fig. 9. Frequency spectrum of denoised spectrograms at  $t = 0.79$  seconds. (a) OGS[abs]. (b) OGS[atan]. The group size is  $K = (8, 2)$  in both cases. The noise-free spectrum is in gray.

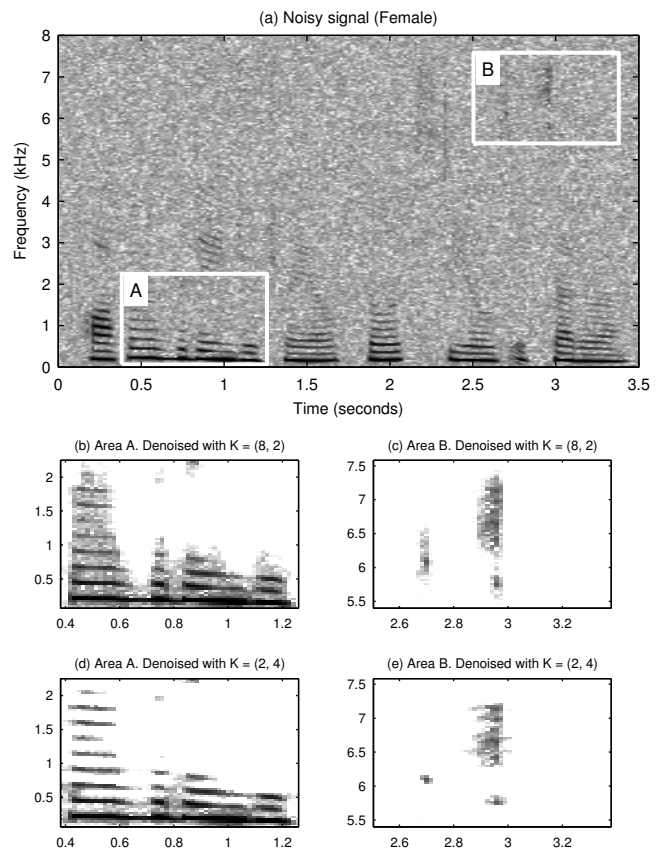


Fig. 10. Denoised spectrograms; female speaker. (a) Noisy spectrogram with SNR = 10 dB. (b, c) Areas A and B, denoised with group size  $(8, 2)$ . (d, e) Areas A and B, denoised with group size  $(2, 4)$ .

$1 \leq K_2 \leq 4$ . In this experiment, we have used speech sampled at 16 kHz, an SNR of 10 dB, and we have selected  $\lambda$  in each case according to the preceding note [suppression of noise down to  $(3 \times 10^{-4})\sigma$ ]. We found that for the male speaker, a group size of  $(8, 2)$  maximized the SNR most frequently. This conforms with our informal listening tests with different group sizes. The denoised spectrum in Figure 8b was obtained using this group size of  $(8, 2)$ .

For the female speaker, the experiment reveals that a group size of  $(2, 4)$  maximizes the SNR most frequently. However, we found that this group size results in poor perceptual quality. To investigate the effect of group size, the denoised spectrograms using groups of size  $(8, 2)$  and  $(2, 4)$  are illustrated in Fig. 10. Fig. 10a shows the noisy spectrogram (file *arctic\_a0001*). We highlight two areas of the spectrogram. The low-frequency area, denoted ‘A’, exhibits a high level of temporal correlation. On the other hand, the high-frequency area, denoted ‘B’, exhibits a high level of spectral correlation. Figs. 10(b,c) show areas A and B of the spectrogram obtained using group size  $(8, 2)$ . Figs. 10(d,e) show areas A and B of the spectrogram obtained using group size  $(2, 4)$ .

It can be observed in area A that group size  $(2, 4)$  suppresses the inter-formant noise more completely than group size  $(8, 2)$ . Conversely, in area B, group size  $(8, 2)$  recovers the original spectrogram more accurately than group size  $(2, 4)$ . Since area

A is representative of more of the spectrogram than area B, the SNR-optimal group size for the whole spectrogram is (2, 4). However, due to the distortion of high frequencies, as in area B, group size (2, 4) yields the perceptually inferior result. Moreover, the lower inter-formant noise suppression of group size (8, 2) appears to have a negligible adverse impact on perceptual quality. Therefore, even though group size (2, 4) yields a higher SNR for the female speaker, we use group size (8, 2) in the evaluation of OGS due to its superior perceptual quality. This points to the potential value of allowing groups in OGS to be sized adaptively, as in Ref. [64]. However, we do not explore such an extension of OGS in this work.

We conducted equivalent evaluations at the sampling rate of 8 kHz in order to determine an appropriate group size for this case. We found that group sizes of  $K = (7, 2)$  and  $K = (3, 3)$  were optimal in terms of SNR, for the male and female speaker, respectively. As above, we selected the group size  $K = (7, 2)$  for both genders for its better perceptual quality.

**Algorithm comparisons.** In Table V we compare the OGS[atan] algorithm with several other speech enhancement algorithms. The table summarizes the output SNR for two sampling rates, male and female speakers, and two input SNR (noise) levels. Each SNR value is averaged over 30 or 15 sentences, depending on the sampling rate. It can be observed that the proposed algorithm, OGS[atan], achieves the highest SNR in each case. (We also note that in all cases, OGS is used not with SNR-optimized  $\lambda$ , but with the larger  $\lambda$ , set according to the noise suppression method. The SNR of OGS could be further increased, but at the cost of perceptual quality.)

The algorithms used in the comparison are: spectral subtraction (SS) [4], the log-MMSE algorithm (LMA) [15], the subspace algorithm (SUB) [31], block thresholding (BT) [64], and persistent shrinkage (PS) [61]. For SS, LMA, and SUB, we used the MATLAB software provided in Ref. [38]. For the BT<sup>2</sup> and PS<sup>3</sup> algorithms, we used the software provided by the authors on their web pages.

Furthermore, we additionally evaluated each method with empirical Wiener post-processing (EWP) [29]. The EWP technique is based on mean square error minimization and its effectiveness has been well demonstrated [13], [18], [64]. In Table V, SNR values obtained using EWP are shown in parenthesis for each algorithm and scenario.

The proposed algorithm, OGS[atan], achieves the highest SNR for both noise levels and genders. For example, for the male speaker with an input SNR of 10 dB, OGS[atan] attains the highest output SNR of 16.58 dB. BT achieves the second highest, 15.61 dB. In terms of perceptual quality, SS and LMA have clearly audible artifacts; BT and PS have slight audible artifacts; OGS[atan], OGS[abs] and SUB have the least audible artifacts. However, SUB has a high computational complexity due to eigenvalue factorization. Compared to OGS[abs] and SUB, OGS[atan] better preserves the perceptual quality of high frequencies. Similar results can be observed for different noise levels and the female speaker.

Empirical Wiener post-processing (EWP) improves the SNR

TABLE V  
AVERAGE SNR FOR SIX SPEECH ENHANCEMENT ALGORITHMS.

(a) $f_s = 16$ kHz (average of 30 samples)				
Method	Male / Input SNR (dB)		Female / Input SNR (dB)	
	5	10	5	10
SS	9.44 (10.96)	13.63 (14.99)	13.36 (14.59)	16.86 (17.93)
LMA	10.24 (11.64)	13.30 (15.25)	13.30 (15.16)	15.71 (18.13)
SUB	11.28 (12.31)	13.94 (16.11)	13.39 (15.31)	15.05 (18.48)
BT	12.00 (12.49)	15.61 (16.10)	15.09 (15.69)	18.18 (18.78)
PS	10.75 (12.00)	14.17 (15.73)	12.67 (14.71)	16.39 (18.13)
OGS[abs]	10.48 (12.36)	13.92 (16.00)	12.91 (15.53)	16.24 (18.60)
OGS[atan]	12.93 (12.98)	16.58 (16.58)	15.37 (15.83)	18.68 (19.02)

(b) $f_s = 8$ kHz (average of 15 samples)				
Method	Male / Input SNR (dB)		Female / Input SNR (dB)	
	5	10	5	10
SS	10.73 (11.75)	14.57 (15.54)	10.45 (11.59)	14.38 (15.47)
LMA	10.66 (12.00)	13.75 (15.61)	9.34 (11.05)	12.51 (14.85)
SUB	10.83 (12.29)	14.03 (16.06)	9.57 (11.53)	13.25 (15.55)
BT	11.80 (12.48)	15.45 (16.10)	11.54 (12.40)	15.12 (16.00)
PS	10.45 (12.20)	13.64 (15.75)	9.11 (11.20)	13.52 (15.47)
OGS[abs]	9.96 (12.25)	13.42 (15.87)	9.34 (11.91)	12.81 (15.70)
OGS[atan]	12.80 (12.97)	16.41 (16.53)	12.10 (12.62)	15.84 (16.31)

for all methods at all noise levels, but least for OGS[atan]. EWP is effective for increasing SNR because it effectively rescales large-amplitude STFT coefficients that are unnecessarily attenuated by these algorithms (the results of which are biased toward zero). The fact that EWP yields the least improvement for OGS[atan] demonstrates that this algorithm inherently induces less bias than the other algorithms.

According to informal listening tests (conducted at input SNR of 10 dB and sampling rate of  $f_s = 16$  kHz), the effect of EWP on audible artifacts depends on the algorithm. Although EWP improves the SNR of SS and LMA, denoising artifacts are still clearly perceptible. EWP improves the perceptual quality of BT and PS slightly. EWP also improves perceptual quality of OGS[abs] and SUB, which already had good perceptual quality. The effect of EWP on OGS[atan] is almost imperceptible; its good perceptual quality is maintained.

Figure 11 illustrates the individual SNRs of the 30 sentences denoised using each of the utilized algorithms (male, input SNR of 10 dB,  $f_s$  of 16 kHz). It can be observed that EWP improves each algorithm, except OGS[atan]. However, as shown in Fig. 11b, OGS[atan] outperforms the other algorithms in terms of SNR irrespective of EWP.

## V. REMARKS

Several aspects of the non-convex regularized OGS algorithm are sufficiently similar to those of the convex regularized OGS algorithm [13] that we refer the reader to Ref. [13]. In particular, remarks in Ref. [13] regarding the convergence behavior, implementation issues, computational complexity, and relationship of OGS to FOCUS [54], apply also to the version of OGS presented here.

The proximal framework has proven effective for convex optimization problems arising in sparse signal estimation and reconstruction [16], [17]. The proposed non-convex regularized OGS algorithm resembles a proximity operator; however,

<sup>2</sup><http://www.cmap.polytechnique.fr/~yu/research/ABT/samples.html>

<sup>3</sup><http://homepage.univie.ac.at/monika.doerfler/StrucAudio.html>

## APPENDIX

Proposition 2 uses the following result.

**Lemma A.** Suppose  $G : \mathbb{R} \rightarrow \mathbb{R}$  is continuous, and the second derivative of  $G$  exists satisfying  $G''(x) > 0$  on  $\mathbb{R} \setminus \{0\}$ . If  $G'(0^-) < G'(0^+)$ , then  $G$  is strictly convex on  $\mathbb{R}$ .

*Proof.* From Theorem 6.4 of [30], if a function  $f$  is continuous on an open interval  $I$  and possesses a strictly increasing right-derivative, or a strictly increasing left-derivative, on  $I$ , then  $f$  is strictly convex on  $I$ .

Hence, it is sufficient to prove the right derivative of  $G$  is strictly increasing on  $\mathbb{R}$ . Let  $x_1 < x_2$ . Case 1: If  $x_2 < 0$  or  $x_1 \geq 0$ , then  $G'(x_1^+) < G'(x_2^+)$  because  $G''(x)$  is positive on  $\mathbb{R} \setminus \{0\}$ . Case 2: If  $x_1 < 0$  and  $x_2 \geq 0$ , then  $G'(x_1^+) < G(0^-) < G(0^+) \leq G(x_2^+)$ . Hence  $G'(x^+)$  is strictly increasing and  $G$  is strictly convex.  $\square$

The following result is similar to Lemma 1 of Ref. [65] which considered only convex regularizers. We use this lemma to justify the initialization of the set  $S$  in Sec. III-C.

**Lemma B.** Let  $\phi(\cdot, a) : \mathbb{R} \rightarrow \mathbb{R}$  satisfy the assumptions in Sec. II-B and define  $F$  as in (10). Suppose  $F$  is strictly convex and that  $\mathbf{x}^*$  is the minimizer of  $F$ .

- 1) If  $y(i) = 0$  for some  $i$ , then  $x^*(i) = 0$ .
- 2) If  $y(i) > 0$  for some  $i$ , then  $x^*(i) \geq 0$ .
- 3) If  $y(i) < 0$  for some  $i$ , then  $x^*(i) \leq 0$ .
- 4)  $|x^*(i)| \leq |y(i)|, \forall i \in \mathbb{Z}_N$ .

*Proof.* 1) Define  $S = \{i \in \mathbb{Z}_N : y(i) \neq 0\}$  and  $\bar{S} = \mathbb{Z}_N \setminus S$ . Given  $\mathbf{x} \in \mathbb{R}^N$ , define  $\tilde{\mathbf{x}} \in \mathbb{R}^N$  as  $\tilde{x}(i) = x(i)$  for  $i \in S$ , and  $\tilde{x}(i) = 0$  for  $i \in \bar{S}$ . For each group  $i \in \mathbb{Z}_N$ , we have  $\|\mathbf{x}_{i,K}\|_2 \geq \|\tilde{\mathbf{x}}_{i,K}\|_2$ . Since  $\phi(t)$  is increasing for  $t \geq 0$ , we have  $\phi(\|\mathbf{x}_{i,K}\|_2) \geq \phi(\|\tilde{\mathbf{x}}_{i,K}\|_2)$ . Therefore, for all  $\mathbf{x} \in \mathbb{R}^N$ ,

$$\begin{aligned} F(\mathbf{x}) &= \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 + \sum_i \lambda \phi(\|\mathbf{x}_{i,K}\|_2; a) \\ &= \frac{1}{2} \|\mathbf{y} - \tilde{\mathbf{x}}\|_2^2 + \frac{1}{2} \sum_{i \in \bar{S}} |x(i)|^2 + \sum_i \lambda \phi(\|\mathbf{x}_{i,K}\|_2; a) \\ &\geq \frac{1}{2} \|\mathbf{y} - \tilde{\mathbf{x}}\|_2^2 + \sum_i \lambda \phi(\|\tilde{\mathbf{x}}_{i,K}\|_2; a) \\ &= F(\tilde{\mathbf{x}}). \end{aligned}$$

This implies  $x^*(i) = 0$  for  $i \in \bar{S}$ .

2) Proof by contradiction. Suppose  $y(i) > 0$ , but  $x^*(i) < 0$  for some  $i$ . Define  $\tilde{\mathbf{x}}$  by  $\tilde{x}(i) = 0$ , and  $\tilde{x}(n) = x^*(n)$  for  $n \neq i$ . It can be shown as in 1) that  $F(\mathbf{x}^*) > F(\tilde{\mathbf{x}})$ . This contradicts the optimality of  $\mathbf{x}^*$ .

3) The proof is like 2).

4) Proof by contradiction. Suppose  $y(i) \geq 0$ , but  $x^*(i) > y(i)$  for some  $i$ . Define  $\tilde{\mathbf{x}}$  by  $\tilde{x}(i) = y(i)$ , and  $\tilde{x}(n) = x^*(n)$  for  $n \neq i$ . It can be shown as in 1) that  $F(\mathbf{x}^*) > F(\tilde{\mathbf{x}})$ . This contradicts the optimality of  $\mathbf{x}^*$ . Together with 2), it follows that if  $y(i) \geq 0$ , then  $0 \leq x^*(i) \leq y(i)$ . Similarly, if  $y(i) \leq 0$ , then  $y(i) \leq x^*(i) \leq 0$ .  $\square$

## ACKNOWLEDGMENT

The authors thank the anonymous reviewers for suggestions and corrections that improved the manuscript.

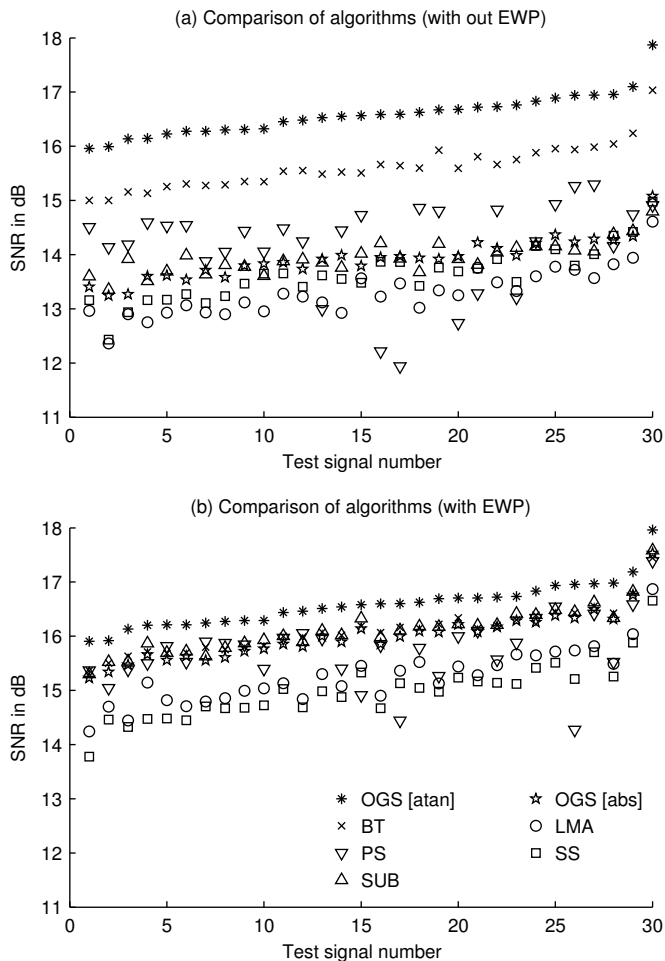


Fig. 11. SNR comparison of speech enhancement algorithms (30 male sentences, input SNR of 10 dB). Each algorithm is used without EWP (a) and with EWP (b). The sentences are ordered according to the SNR of OGS[atan].

a proximity operator is defined in terms of a convex penalty function [17]. Hence, the proposed approach appears to fall outside the proximal framework. Due to the effectiveness of the proximal framework for solving inverse problems much more general than denoising (e.g. deconvolution), it will be of interest in future work to explore the extent to which the proposed method can be used for more general inverse problems by using proximal-like techniques.

## VI. CONCLUSION

This paper formulates group-sparse signal denoising as a convex optimization problem with a non-convex regularization term. The regularizer is based on overlapping groups so as to promote group-sparsity. The regularizer, being concave on the positive real line, promotes sparsity more strongly than any convex regularizer can. For several non-convex penalty functions, parameterized by a variable,  $a$ , it has been shown how to constrain  $a$  to ensure the optimization problem is strictly convex. We also develop a fast iterative algorithm for the proposed approach. Numerical experiments demonstrate the effectiveness of the proposed method for speech enhancement.

## REFERENCES

- [1] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.
- [2] I. Bayram. Mixed norms with overlapping groups as signal priors. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, pages 4036–4039, May 2011.
- [3] I. Bayram and O. D. Akyildiz. Primal-dual algorithms for audio decomposition using mixed norms. *Signal, Image and Video Proc.*, 8(1):95–110, January 2014.
- [4] M. Berouti, R. Schwartz, and J. Makhoul. Enhancement of speech corrupted by acoustic noise. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, volume 4, pages 208–211, April 1979.
- [5] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, 1987.
- [6] A. Blumensath. Accelerated iterative hard thresholding. *Signal Processing*, 92(3):752–756, 2012.
- [7] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.
- [8] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [9] E. J. Candes, C. A. Sing-Long, and J. D. Trzasko. Unbiased risk estimates for singular value thresholding and spectral estimators. *IEEE Trans. Signal Process.*, 61(19):4643–4657, October 2013.
- [10] E. J. Candès, M. B. Wakin, and S. Boyd. Enhancing sparsity by reweighted  $\ell_1$  minimization. *J. Fourier Anal. Appl.*, 14(5):877–905, December 2008.
- [11] R. Chartrand. Exact reconstruction of sparse signals via nonconvex minimization. *IEEE Signal Processing Letters*, 14(10):707–710, 2007.
- [12] R. Chartrand and B. Wohlberg. A nonconvex ADMM algorithm for group sparsity with sparse groups. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, May 2013.
- [13] P.-Y. Chen and I. W. Selesnick. Translation-invariant shrinkage/thresholding of group sparse signals. *Signal Processing*, 94:476–489, January 2014.
- [14] X. Chen, Q. Lin, S. Kim, J. G. Carbonell, and E. P. Xing. Smoothing proximal gradient method for general structured sparse regression. *Ann. Appl. Stat.*, 6(2):719–752, 2012.
- [15] I. Cohen. Optimal speech enhancement under signal presence uncertainty using log-spectral amplitude estimator. *IEEE Signal Processing Letters*, 9(4):113–116, April 2002.
- [16] P. L. Combettes and J.-C. Pesquet. Proximal thresholding algorithm for minimization over orthonormal bases. *SIAM J. Optim.*, 18(4):1351–1376, 2008.
- [17] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In H. H. Bauschke et al., editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer-Verlag, 2011.
- [18] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3d transform-domain collaborative filtering. *IEEE Trans. Image Process.*, 16(8):2080–2095, August 2007.
- [19] W. Deng, W. Yin, and Y. Zhang. Group sparse optimization by alternating direction method. *Rice University CAAM Technical Report TR11-06, 2011*, 2011.
- [20] D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.
- [21] Y. C. Eldar and M. Mishali. Robust recovery of signals from a structured union of subspaces. *IEEE Trans. Inform. Theory*, 55(11):5302–5316, 2009.
- [22] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Statist. Assoc.*, 96(456):1348–1360, 2001.
- [23] M. Figueiredo and J. Bioucas-Dias. An alternating direction algorithm for (overlapping) group regularization. In *Signal Processing with Adaptive Sparse Structured Representations (SPARS)*, Edinburgh, UK, June 2011.
- [24] M. Figueiredo, J. Bioucas-Dias, and R. Nowak. Majorization-minimization algorithms for wavelet-based image restoration. *IEEE Trans. Image Process.*, 16(12):2980–2991, December 2007.
- [25] S. Foucart. Hard thresholding pursuit: an algorithm for compressive sensing. *SIAM J. Numer. Anal.*, 49(6):2543–2563, 2010.
- [26] S. Foucart and M.-J. Lai. Sparsest solutions of underdetermined linear systems via  $\ell_q$ -minimization for  $0 < q \leq 1$ . *J. of Appl. and Comp. Harm. Analysis*, 26(3):395–407, March 2009.
- [27] G. Gasso, A. Rakotomamonjy, and S. Canu. Recovering sparse signals with a certain family of nonconvex penalties and DC programming. *IEEE Trans. Signal Process.*, 57(12):4686–4698, December 2009.
- [28] D. Geman and G. Reynolds. Constrained restoration and the recovery of discontinuities. *IEEE Trans. Pattern Anal. and Machine Intel.*, 14(3):367–383, March 1992.
- [29] S. Ghael, A. M. Sayeed, and R. G. Baraniuk. Improved wavelet denoising via empirical Wiener filtering. In *Proceedings of SPIE*, volume 3169 (Wavelet Apps. Signal. Image Proc. V), October 1997.
- [30] J.-B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of Convex Analysis*. Springer, 2001.
- [31] Y. Hu and P. C. Loizou. A generalized subspace approach for enhancing speech corrupted by colored noise. *IEEE Trans. on Acoust., Speech, Signal Proc.*, 11(4):334–341, July 2003.
- [32] L. Jacob, G. Obozinski, and J.-P. Vert. Group lasso with overlap and graph lasso. *Int. Conf. Machine Learning (ICML)*, pages 433–440, 2009.
- [33] R. Jenatton, J.-Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. *J. Mach. Learning Research*, 12:2777–2824, October 2011.
- [34] N. Kingsbury and T. Reeves. Redundant representation with complex wavelets: how to achieve sparsity. In *Proc. IEEE Int. Conf. Image Processing*, volume 1, pages 45–48, 2003.
- [35] M. Kowalski. Sparse regression using mixed norms. *J. of Appl. and Comp. Harm. Analysis*, 27(3):303–324, 2009.
- [36] M. Kowalski and B. Torrèsani. Sparsity and persistence: mixed norms provide simple signal models with dependent coefficients. *Signal, Image and Video Processing*, 3(3):251–264, 2009.
- [37] I. Kozlov and A. Petukhov. Sparse solutions of underdetermined linear systems. In W. Freeden et al., editor, *Handbook of Geomathematics*. Springer, 2010.
- [38] P. C. Loizou. *Speech enhancement: theory and practice*. CRC Press, 2007.
- [39] D. A. Lorenz. Non-convex variational denoising of images: Interpolation between hard and soft wavelet shrinkage. *Current Development in Theory and Application of Wavelets*, 1(1):31–56, 2007.
- [40] S. Mallat. *A wavelet tour of signal processing*. Academic Press, 1998.
- [41] S. Mosci, S. Villa, A. Verri, and L. Rosasco. A primal-dual algorithm for group sparse regularization with overlapping groups. In *Advances in Neural Information Processing Systems 23*, pages 2604–2612, 2010.
- [42] P. Moulin and J. Liu. Analysis of multiresolution image denoising schemes using generalized Gaussian and complexity priors. *IEEE Trans. Inform. Theory*, 45(3):909–919, April 1999.
- [43] M. Nikolova. Estimation of binary images by minimizing convex criteria. In *Proc. IEEE Int. Conf. Image Processing*, pages 108–112 vol. 2, 1998.
- [44] M. Nikolova. Markovian reconstruction using a GNC approach. *IEEE Trans. Image Process.*, 8(9):1204–1220, 1999.
- [45] M. Nikolova. Analysis of the recovery of edges in images and signals by minimizing nonconvex regularized least-squares. *Multiscale Model. Simul.*, 4(3):960–991, 2005.
- [46] M. Nikolova, J. Idier, and A. Mohammad-Djafari. Inversion of large-support ill-posed linear operators using a piecewise Gaussian MRF. *IEEE Trans. Image Process.*, 7(4):571–585, 1998.
- [47] M. Nikolova, M. Ng, S. Zhang, and W. Ching. Efficient reconstruction of piecewise constant images using nonsmooth nonconvex minimization. *SIAM J. Imag. Sci.*, 1(1):2–25, 2008.
- [48] M. Nikolova, M. K. Ng, and C.-P. Tam. Fast nonconvex nonsmooth minimization methods for image restoration and reconstruction. *IEEE Trans. Image Process.*, 19(12):3073–3088, December 2010.
- [49] J. Oliveira, J. Bioucas-Dias, and M. A. T. Figueiredo. Adaptive total variation image deblurring: A majorization-minimization approach. *Signal Processing*, 89(9):1683–1693, September 2009.
- [50] G. Peyre and J. Fadili. Group sparsity with overlapping partition functions. In *Proc. European Sig. Image Proc. Conf. (EUSIPCO)*, Aug. 29 - Sept. 2 2011.
- [51] J. Portilla and L. Mancera.  $L_0$ -based sparse approximation: two alternative methods and some applications. In *Proceedings of SPIE*, volume 6701 (Wavelets XII), San Diego, CA, USA, 2007.
- [52] A. T. Puig, A. Wiesel, G. Fleury, and A. O. Hero. Multidimensional shrinkage-thresholding operator and group LASSO penalties. *IEEE Signal Processing Letters*, 18(6):363–366, 2011.
- [53] S. Ramani, T. Blu, and M. Unser. Monte-carlo SURE: A black-box optimization of regularization parameters for general denoising algorithms. *IEEE Trans. Image Process.*, 15(9):1540–1554, September 2008.

- [54] B. D. Rao, K. Engan, S. F. Cotter, J. Palmer, and K. Kreutz-Delgado. Subset selection in noise based on diversity measure minimization. *IEEE Trans. Signal Process.*, 51(3):760–770, March 2003.
- [55] W. Rudin. *Principles of mathematical analysis*. McGraw-Hill, New York, third edition, 1976.
- [56] I. Selesnick. Penalty and shrinkage functions for sparse signal processing. *Connexions*, 2012. <http://cnx.org/content/m45134/>.
- [57] I. W. Selesnick. The estimation of Laplace random vectors in additive white Gaussian noise. *IEEE Trans. Signal Process.*, 56(8):3482–3496, August 2008.
- [58] I. W. Selesnick and I. Bayram. Sparse signal estimation by maximally sparse convex optimization. *IEEE Trans. Signal Process.*, 62(5):1078–1092, March 2014.
- [59] I. W. Selesnick and P.-Y. Chen. Total variation denoising with overlapping group sparsity. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, pages 5696–5700, May 2013.
- [60] L. Sendur and I. W. Selesnick. Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency. *IEEE Trans. Signal Process.*, 50(11):2744–2756, November 2002.
- [61] K. Siedenburg and M. Dörfler. Persistent time-frequency shrinkage for audio denoising. *J. Audio Eng. Soc.*, 61(1):29–38, 2013.
- [62] C. M. Stein. Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics*, 9(6):1135–1151, November 1981.
- [63] D. Wipf and S. Nagarajan. Iterative reweighted  $\ell_1$  and  $\ell_2$  methods for finding sparse solutions. *IEEE J. Sel. Top. Signal Processing*, 4(2):317–329, April 2010.
- [64] G. Yu, S. Mallat, and E. Bacry. Audio denoising by time-frequency block thresholding. *IEEE Trans. Signal Process.*, 56(5):1830–1839, May 2008.
- [65] L. Yuan, J. Liu, and J. Ye. Efficient methods for overlapping group lasso. In *Advances in Neural Information Processing Systems 24*, pages 352–360. 2011.
- [66] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68(1):49–67, February 2006.

```

function [x, cost] = ogs1(y, K, lam, pen, rho, Nit)

% [x, cost] = ogs1(y, K, lam, pen, rho, Nit)
% Overlapping Group Shrinkage/Thresholding (1D)
%
% Input
% y : input signal (1D array)
% lam : regularization parameter
% K : group size
% pen : penalty ('abs', 'log', 'atan', 'rat')
% rho : normalized non-convexity parameter, 0 <= rho <= 1
% Nit : number of iterations
%
% Output
% x : output signal
% cost : cost function history
%
% Note: for the L1 penalty ('abs'), 'rho' must be 0.

% Po-Yu Chen and Ivan Selesnick
% NYU-Poly, 2013

% Perform some error checking
if strcmp(pen, 'abs')
    if rho ~= 0
        error('Error: need rho = 0 for abs penalty')
    end
end
if rho == 0
    pen = 'abs';
elseif (rho < 0) || (rho > 1)
    error('Error: need 0 <= rho <= 1.')
end

a_max = 1/(lam*K);
a = rho * a_max;

switch pen
    case 'abs'
        phi = @(x) abs(x);
        wfun = @(x) abs(x);
        a = 0;
    case 'rat'
        phi = @(x) abs(x)./(1+a*abs(x)/2);
        wfun = @(x) abs(x) .* (1 + a*abs(x)/2).^2;
    case 'log'
        phi = @(x) 1/a * log(1 + a*abs(x));
        wfun = @(x) abs(x) .* (1 + a*abs(x));
    case 'atan'
        phi = @(x) 2/(a*sqrt(3)) * (atan((1+2*a*abs(x))/sqrt(3)) - pi/6);
        wfun = @(x) abs(x) .* (1 + a*abs(x) + a^2.*abs(x).^2);
    otherwise
        error('penalty must be ''abs'', ''log'', ''atan'', or ''rat''')
end

y = y(:);
h = ones(K, 1);
x = y;
cost = zeros(1, Nit);

for it = 1 : Nit
    r = sqrt( conv(abs(x).^2, h, 'full') );
    cost(it) = 0.5 * sum(abs(x-y).^2) + lam * sum(phi(r));
    v = 1 + lam*conv( 1./(( wfun(r) ), h, 'valid');
    x = y./v;
end

```



```

function [x, cost] = ogs2(y, K1, K2, lam, pen, rho, Nit)

% [x, cost] = ogs2(y, K1, K2, lam, pen, rho, Nit)
% Overlapping Group Shrinkage/Thresholding (2D)
%
% Input
% y      : observed signal (2D array)
% lam    : regularization parameter
% K1, K2 : group size
% pen    : 'abs', 'log', 'atan' or 'rat'
% rho    : normalized non-convexity parameter, 0 <= rho <= 1
% Nit    : number of iterations
%
% Output
% x      : output signal
% cost   : cost function history
%
% Note: for the L1 penalty ('abs'), 'rho' must be 0.

% Po-Yu Chen and Ivan Selesnick
% NYU-Poly, 2013

% Perform some error checking
if strcmp(pen, 'abs')
    if rho ~= 0
        error('Error: need rho = 0 for abs penalty')
    end
end
if rho == 0
    pen = 'abs';
elseif (rho < 0) || (rho > 1)
    error('Error: need 0 <= rho <= 1.')
end

a_max = 1/(lam*K1*K2);
a = rho * a_max;

switch pen
    case 'abs'
        phi = @(x) abs(x);
        wfun = @(x) abs(x);
        a = 0;
    case 'rat'
        phi = @(x) abs(x)./(1+a*abs(x)/2);
        wfun = @(x) abs(x) .* (1 + a*abs(x)/2).^2;
    case 'log'
        phi = @(x) 1/a * log(1 + a*abs(x));
        wfun = @(x) abs(x) .* (1 + a*abs(x));
    case 'atan'
        phi = @(x) 2/(a*sqrt(3)) * (atan((1+2*a*abs(x))/sqrt(3)) - pi/6);
        wfun = @(x) abs(x) .* (1 + a*abs(x) + a^2.*abs(x).^2);
    otherwise
        disp('penalty must be abs, log, atan, or rat')
        x = []; cost = [];
        return
end

h1 = ones(K1, 1);
h2 = ones(K2, 1);
x = y;
cost = zeros(1, Nit);

for it = 1 : Nit
    r = sqrt( conv2(h1, h2, abs(x).^2, 'full') );
    cost(it) = 0.5 * sum(sum(abs(x-y).^2)) + lam * sum(sum(phi(r)));
    v = 1 + lam*conv2(h1, h2, 1./wfun(r), 'valid');
    x = y./v;
end

```