

# Estimating Sparse Signals with Smooth Support via Convex Programming and Block Sparsity

Sohil Shah<sup>1</sup>, Tom Goldstein<sup>1</sup>, and Christoph Studer<sup>2</sup>

<sup>1</sup>University of Maryland, College Park; <sup>2</sup>Cornell University, Ithaca, NY

sohilas@umd.edu, tomcg@cs.umd.edu, studer@cornell.edu

## Abstract

*Conventional algorithms for sparse signal recovery and sparse representation rely on  $\ell_1$ -norm regularized variational methods. However, when applied to the reconstruction of sparse images, i.e., images where only a few pixels are non-zero, simple  $\ell_1$ -norm-based methods ignore potential correlations in the support between adjacent pixels. In a number of applications, one is interested in images that are not only sparse, but also have a support with smooth (or contiguous) boundaries. Existing algorithms that take into account such a support structure mostly rely on non-convex methods and—as a consequence—do not scale well to high-dimensional problems and/or do not converge to global optima. In this paper, we explore the use of new block  $\ell_1$ -norm regularizers, which enforce image sparsity while simultaneously promoting smooth support structure. By exploiting the convexity of our regularizers, we develop new computationally-efficient recovery algorithms that guarantee global optimality. We demonstrate the efficacy of our regularizers on a variety of imaging tasks including compressive image recovery, image restoration, and robust PCA.*

## 1. Introduction

A large number of existing models used in sparse signal processing and machine learning rely on  $\ell_1$ -norm regularization in order to recover sparse signals or to identify sparse features for classification tasks. Sparse  $\ell_1$ -norm regularization is also prominently used in the image-processing and computer vision domain, where it is used for segmentation, tracking, and background subtraction tasks. In computer vision and image processing, we are often interested in regions that are not only sparse, but also *spatially smooth*, i.e., regions with contiguous support structure. In such situations, it is desirable to have regularizers that promote the selection of large, contiguous regions rather than merely sparse (and potentially isolated) pixels. In contrast, simple  $\ell_1$ -norm regularization adopts an unstructured approach that induces

sparsity wherein each variable is treated independently, disregarding correlation among neighboring variables. For example, smooth support structure is relevant to compressive background subtraction [10, 9] which detects contiguous regions of movement against a stationary background.

For imaging applications,  $\ell_1$ -norm regularization may result in regions with spurious active (or isolated) pixels or non-smooth boundaries in the support set. This issue is addressed by the image-segmentation literature, where spatially correlated priors (such as total variation or normalized cuts) are used to enforce smooth support boundaries [5, 14, 34, 12]. An important hallmark of existing image-segmentation methods is that they are able to enforce spatially contiguous support. However, the concept of correlated support has yet to be ported to more complex reconstruction tasks, including (but not limited to) robust PCA and compressive background subtraction. The development of such structured sparsity models has been an active research topic [9, 2, 19, 21, 1, 20], with new models and applications still emerging [23, 22]. In this paper, we develop a class of convex priors based on overlapping block/group sparsity, which are able to enforce sparsity of the support set and promote *spatial smoothness*.

### 1.1. Relevant Previous Work

Existing work on spatially-smooth support-set regularization can be divided into two main categories: (i) non-convex models that rely on graphs and trees, and (ii) convex models that rely on group-sparsity inducing norms. Cevher *et al.* [9] promote sparsity using Markov random fields (MRFs) in combination with compressive-sensing signal recovery, which is referred to as lattice matching pursuit (LaMP). LaMP recovers structured sparse signals using fewer noisy measurements than methods that ignore spatially correlated support sets. Baraniuk *et al.* [2] prove theoretical guarantees on robust recovery of structured sparse signals using a non-convex algorithm; their approach has been validated using wavelet-tree-based hierarchical group structure, as well as signals with non-overlapping blocks in the support set. Huang *et al.* [19] developed a theory of greedy approximation methods for general non-convex structured sparse

models. All these methods, however, are limited in that they are either non-convex, computationally expensive, or do not allow for overlapping (or not aligned) group structure. Jenatton *et al.* [21] showed the possibility of coming up with a problem-specific optimal group-sparsity-inducing norm using prior knowledge of the underlying structure. While they consider a convex relaxation of the structured sparsity problem, it remains unclear how their proposed active-set algorithm for least squares regression can be generalized to a broader range of applications.

## 1.2. Contributions

Our work is inspired by the  $\ell_1/\ell_2$ -norm spatial coherence priors used in [21], as well as group sparsity priors used in statistics (e.g., group lasso) [24, 40]. Our main contributions can be summarized as follows: (i) We propose new regularizers for imaging and computer vision applications including compressive image recovery, sparse & low rank decomposition, and a block-sparse generalization of total variation. (ii) We develop computationally efficient global minimization algorithms that are suitable for overlapping pixel-cliques. Existing methods for group sparsity use the alternating direction method of multipliers (ADMM), and have excessive memory requirements for large clique sizes. We therefore discuss a new approach using fast convolution algorithms to perform gradient descent with low memory requirements and a complexity that is independent of the clique size. (iii) We propose the use of our regularizers within greedy pursuit methods for compressive reconstruction. (iv) We demonstrate that our algorithms can be used to suppress artifacts and enhance the quality of sparse recovery methods when applied to a variety of imaging applications.

## 1.3. Notation

For any column vector  $\mathbf{x} \in \mathbb{R}^n$ , we define its  $\ell_\alpha$ -norm with  $\alpha \geq 1$  as  $\|\mathbf{x}\|_\alpha = (\sum_{i=1}^n |x_i|^\alpha)^{1/\alpha}$ . For  $\mathbf{x} \in \mathbb{R}^n$ , the vector  $\mathbf{x}_c$  consists only of the entries associated to the index set  $c$ . The support set (i.e., the set of indices of non-zero entries) of a vector or vectorized image  $\mathbf{x}$  is denoted by  $\text{supp}(\mathbf{x})$ . For a matrix  $\mathbf{A} \in \mathbb{R}^{M \times N}$  with rank  $r = \min\{M, N\}$  and singular values  $\sigma_i$ , the nuclear norm is defined by  $\|\mathbf{A}\|_* = \sum_{i=1}^r \sigma_i$ . We use  $\|\mathbf{A}\|_1 = \sum_{ij} |A_{ij}|$  to denote the element-wise  $\ell_1$ -norm for  $\mathbf{A}$ .

## 2. Problem Formulation

Consider the measurement model  $\mathbf{y} = \Phi \mathbf{x}_0 + \mathbf{z}_0$ , where  $\mathbf{y} \in \mathbb{R}^M$  is the observed signal,  $\mathbf{x}_0 \in \mathbb{R}^N$  is the original sparse signal we wish to recover,  $\mathbf{z}_0 \in \mathbb{R}^M$  is a non-sparse component of the signal (comprising both the background image and potential noise),  $\Phi \in \mathbb{R}^{M \times N}$  is the linear operator that models the signal acquisition process. Based on this model, we study signal recovery by solving convex

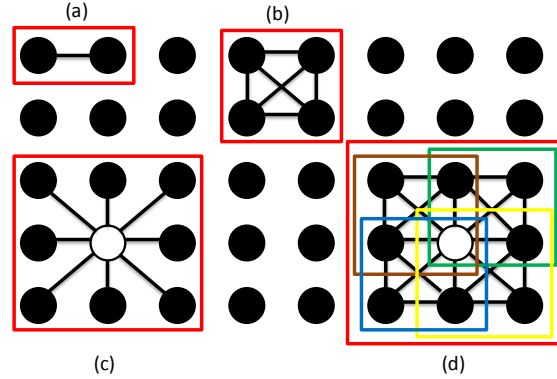


Figure 1: Illustration of cliques and overlapping cliques.

optimization problems of the following general form:

$$\{\hat{\mathbf{z}}, \hat{\mathbf{x}}\} = \arg \min_{\mathbf{z} \in \mathbb{R}^M, \mathbf{x} \in \mathbb{R}^N} D(\mathbf{x}, \mathbf{z} | \mathbf{y}, \Phi) + J(\mathbf{x}). \quad (1)$$

Here,  $D : \mathbb{R}^M \times \mathbb{R}^M \rightarrow \mathbb{R}$  is a convex data-consistency term, and  $J : \mathbb{R}^M \rightarrow \mathbb{R}$  is a regularizer that enforces both sparsity and support smoothness on the vector  $\mathbf{x}$ . The proposed regularizer is a hybrid  $\ell_1/\ell_2$ -norm penalty of the form

$$J(\mathbf{x}) = \sum_{c \in \mathcal{C}} \|\mathbf{x}_c\|_2, \quad (2)$$

where  $\mathcal{C}$  is a set of cliques over the graph  $\mathcal{G}$  defined over the pixels of  $\mathbf{x}$ . This regularizer (2) is a natural generalization of the *group (or block) sparsity* model that has been explored in the literature for a variety of purposes including statistics and radar [21, 20, 23, 24]. We focus on the case where the collection of sparse cliques consist of regularly-spaced groups of adjacent pixels. For example, consider two types of cliques shown in Figure 1(a) and 1(b). Notice the (a) 2-clique and (b) 4-clique wherein all nodes are connected to each other. These cliques can be translated over the entire image graph to generate various overlapping clique geometries as shown in (c) and (d), respectively. In (c), eight overlapping cliques, each of size two, overlap at a central point. In the image processing literature this is referred to as an 8-connected neighborhood [11]. In contrast, Figure 1(d) uses a higher-order connectivity model, which is obtained using four rectangular cliques of size four (each shown in a different color). Overlapping group-sparsity models of the form depicted in Figure 1(d) effectively enforce spatial coherence of the recovered support. When such an overlapping group-sparsity model is used, all pixels in a clique tend to be either zero or non-zero at the same time (see, e.g., [1]). Since each pixel shares multiple overlapping cliques with its neighbors, this regularizer suppresses “rogue” (or isolated) pixels from entering the support without their neighbors and hence, promotes smooth (contiguous) support boundaries.

### 2.1. Applications

The proposed regularizer (2) can be used as a building block for various applications in computer vision, image

processing, and compressive sensing. In what follows, we will focus on the following three imaging applications:

1) *Compressive sensing signal recovery*: Consider a signal  $\mathbf{x} \in \mathbb{R}^N$  that is  $K$ -sparse, i.e., only  $K \ll N$  entries of  $\mathbf{x}$  are non-zero. In the CS literature, the signal is acquired via  $M < N$  linear projections  $\mathbf{y} = \Phi \mathbf{x}$ . The  $K$ -sparse signal  $\mathbf{x}$  can then be recovered if, for example, the matrix  $\Phi$  satisfies the  $2K$ -RIP or similar conditions [2, 8]. The underlying recovery problem is usually formulated as follows:

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbb{R}^N} \|\mathbf{y} - \Phi \mathbf{x}\|_2^2 \quad \text{subject to } \|\mathbf{x}\|_0 = K. \quad (3)$$

When the sparse signals are images, simple sparse recovery may not exploit the entire image structure; this is particularly true for background-subtracted surveillance video. Background subtraction is used in applications where one is interested only in inferring foreground objects and activities. Background subtraction is easily achieved in the compressive domain by computing the difference between adjacent image data or by subtracting a long term signal mean (or median). Background-subtracted frames are generally more sparse than frames containing background information, and can thus be reconstructed from far fewer measurements  $M$ .

We propose to extend the problem in (3) by adding a regularizer of the form (2) to promote correlation in the support set of the foreground objects. The optimization problem defined in (3) is non-convex and is commonly solved using greedy algorithms [35, 30, 9]. We will show that the use of our prior (2) leads to faster signal recovery with a small number of measurements compared to existing methods.

2) *Total-variation denoising*: Total variation (TV) denoising restores a noisy image  $\mathbf{y}$  (e.g., vectorized image) by finding an image that lies close to  $\mathbf{y}$  in an  $\ell_2$ -norm sense, while simultaneously having small total variation; this can be accomplished by solving

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbb{R}^N} \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|^2 + \lambda \|\nabla_d \mathbf{x}\|_1, \quad (4)$$

where  $\nabla_d : \mathbb{R}^N \rightarrow \mathbb{R}^{2N}$  is a discrete gradient operator that acts on an  $N$ -pixel image, and produces a stacked horizontal and vertical gradient vector containing all first-order differences between adjacent pixels. TV-based image processing assumes that images have a piecewise constant representation, i.e., the gradient is sparse and locally contiguous [32, 16]. Numerous generalizations of TV exist, including the recently proposed vectorial TV for color images [6, 31]. Such regularizers are of the form of (4) merely by changing the definition of the discrete gradient operator.

We propose to extend total variation by penalizing the gradient of cliques in order to enforce a greater degree of spatial coherence. In particular, we consider

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbb{R}^N} \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|^2 + J(\nabla_d \mathbf{x}), \quad (5)$$

where  $J(\cdot)$  denotes the regularizer (2). Furthermore, we

explore formulations where the discrete gradient operator is given by the decorrelated color TV operator described in [31]. With our approach, we also show the application of proposed structured sparsity prior on 3-D blocks. Note that [33, 27] explores the use of 1-D and 2-D overlapping group sparsity for TV image denoising, but using a majorization-minimization algorithm combined with ADMM.

3) *Robust PCA (RPCA)*: Suppose  $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_L]$  is a matrix of  $L$  measurement vectors, and  $\mathbf{Y}$  is the sum of a low rank matrix  $\mathbf{Z}$  and a sparse matrix  $\mathbf{X}$ . For this case, Candès *et al.* show that exact recovery of these components is possible using the following formulation [7]:

$$\{\hat{\mathbf{Z}}, \hat{\mathbf{X}}\} = \arg \min_{\mathbf{Z}, \mathbf{X} \in \mathbb{R}^{N \times L}} \|\mathbf{Z}\|_* + \lambda \|\mathbf{X}\|_1 \quad (6)$$

subject to  $\mathbf{Y} = \mathbf{Z} + \mathbf{X}$ .

The nuclear-norm in (6) promotes a low rank solution for  $\mathbf{Z}$ ; the  $\ell_1$ -norm penalty on promotes sparsity in  $\mathbf{X}$ . For this reason the solution to (6) is sometimes referred to as a sparse-plus-low-rank decomposition. A well-known application of RPCA is background subtraction in videos with a stationary background. For such datasets, the shared background in the frames  $\{y_i\}$  can be represented using a low-rank subspace. The moving foreground objects often have sparse support, and thus are absorbed into the sparse term  $\mathbf{X}$ .

We propose to replace the  $\ell_1$ -norm regularization prior on  $\mathbf{X}$  in (6) with the proposed regularizer in (2); this enables us to promote spatial smoothness in the support set of the foreground objects. Here, we build on the work of [15], where structured sparsity with non-overlapping blocks is used in RPCA for foreground detection, and [38], where a hybrid of ALM and network flow methods [28] are used to solve  $\ell_1/\ell_\infty$  regularized RPCA problems.

## 2.2. Optimization Algorithms

We now develop efficient numerical methods for solving problems involving the regularizer (2). A common approach to enforce group sparsity in the statistics literature is consensus ADMM [13, 4], which we will briefly discuss in Section 2.2.1. For image processing and vision applications, where the datasets as well as the cliques tend to be large, the high memory requirements of ADMM render this approach unattractive. As a consequence, we propose an alternative method that uses fast convolution algorithms to perform gradient descent that exhibits low memory requirements and requires low computational complexity. In particular, our approach is capable of handling large-scale problems, such as those in video applications, which are out of the scope of memory-hungry ADMM algorithms.

We note that numerical methods for overlapping group sparsity have been studied in the context of statistical regression [20, 39, 4], but for different purposes. Yuan *et al.* [39] solves the regression variable selection problem using an

accelerated gradient descent approach, whereas Deng [13] and Boyd [4] use consensus ADMM, which does not scale to high-dimensional problems. Compared to these methods, our approach provides significant speedups (see Section 3).

### 2.2.1 Proximal Minimization and ADMM

The simplest instance of the problem (1) is the proximal operator for the penalty term  $J$  in (2), defined as follows:

$$\text{prox}_J(\mathbf{v}, \lambda) = \arg \min_{\mathbf{x}} \|\mathbf{x} - \mathbf{v}\|^2 + \lambda J(\mathbf{x}). \quad (7)$$

Proximal minimization is a key sub-step in a large number of numerical methods. For example, the ADMM for TV minimization [32, 16] requires the computation of the proximal operator of the  $\ell_1$ -norm. For such methods, the regularizer (2) is easily incorporated into the numerical procedure by replacing this proximal minimization with (7).

In the simplest case where the cliques in  $\mathcal{C}$  are small and no other regularizers are needed, the proximal minimization (7) can be computed using ADMM [4, 16]. Similar approaches have been used for other applications of overlapping group sparsity [13]. It is key to realize that the regularization term in (7) can be reformulated as follows:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathbb{R}^N} \|\mathbf{x} - \mathbf{v}\|_2^2 + \lambda \sum_{i=1}^s \sum_{c \in \mathcal{C}_i} \|\mathbf{x}_c\|_2. \quad (8)$$

Here,  $\mathcal{C}_1, \dots, \mathcal{C}_s$  are clique subsets for which the cliques in  $\mathcal{C}_i$  are *disjoint*. For example, consider the case where the set of cliques contains all  $2 \times 2$  image patches as shown in Figure 1(d). For such a scenario, we need four subsets of disjoint cliques to represent every possible patch. The reformulated problem for the example graph will be of the form (8) with  $s = 4$ . In general, if cliques are formed by translating an  $l \times l$  patch,  $l^2$  subsets of cliques are required so that every subset contains only disjoint cliques.

To apply ADMM to this problem, we need to introduce  $s$  auxiliary variables  $\mathbf{z}^1, \dots, \mathbf{z}^s$  each representing a copy of the original pixel values. The resulting problem is

$$\begin{aligned} \{\hat{\mathbf{x}}, \hat{\mathbf{z}}^i \forall i\} = \arg \min_{\mathbf{x}, \{\mathbf{z}^i\}_{i=1}^s} & \|\mathbf{x} - \mathbf{v}\|_2^2 + \lambda \sum_{i=1}^s \sum_{c \in \mathcal{C}_i} \|\mathbf{z}_c^i\|_2 \\ & \text{subject to } \mathbf{z}^i = \mathbf{x}, \forall i. \end{aligned} \quad (9)$$

This is an example of a consensus optimization problem, which can be solved using ADMM (see [13] for more details). An important property of this ADMM reformulation is that each vector  $\mathbf{z}^i$  can be updated in closed form—an immediate result of the disjoint clique decomposition.

### 2.2.2 Forward-Backward Splitting (FBS) with Fast Fourier Transforms

The above discussed ADMM approach has several drawbacks. First, it is difficult to incorporate more regularizers

(in addition to the support regularizer  $J$ ) without the introduction of an excessive amount of additional auxiliary variables. Furthermore, the method becomes inefficient and memory intensive for large clique sizes and large data-sets (as it is the case for multiple images). For instance in RPCA, if the cliques are generated by  $l \times l$  patches,  $l^2$  variables  $\{\mathbf{z}^i\}$  are required, each having the same dimensionality as original image data-set  $NL$ . Additionally, the dual variables for each equality constraints in (9) will require another  $l^2 NL$  storage entries. As a consequence, for large values of  $l$ , the memory requirements of ADMM become prohibitive.

We propose a new forward-backward splitting algorithm that exploits fast convolution operators and prevents the excessive memory overhead of ADMM-based methods. To this end, we propose to “smoother” the objective via *hyperbolic regularization* of the  $\ell_2$ -norm as

$$\|\mathbf{x}_c\|_2 \approx \|\mathbf{x}_c\|_{2,\epsilon} = \sqrt{x_1^2 + \dots + x_n^2 + \epsilon^2} \quad (10)$$

for some small  $\epsilon > 0$ . For the sake of clarity, we describe the forward-backward splitting approach in the specific case of robust PCA. Note, however, that other regularizers are possible with only minor modifications.

Using the proposed support prior (6), we write

$$\{\hat{\mathbf{Z}}, \hat{\mathbf{X}}\} = \arg \min_{\mathbf{Z}, \mathbf{X}} \|\mathbf{Z}\|_* + \lambda J_\epsilon(\mathbf{X}) + \frac{\mu}{2} \|\mathbf{Y} - \mathbf{Z} - \mathbf{X}\|_F^2 \quad (11)$$

where

$$J_\epsilon(\mathbf{X}) = \sum_{t=1}^L \sum_{c \in \mathcal{C}} \|\mathbf{X}_{t,c}\|_{2,\epsilon} \quad (12)$$

is the smoothed support regularizer, and  $\mathbf{X}_{t,c}$  refers to the clique  $c$  drawn from column  $t$  of  $\mathbf{X}$ . We note that this formulation differs from that in Liu *et al.* [26], where the structured sparsity is induced across columns of  $\mathbf{X}$  rather than blocks, and is solved using conventional ADMM.

The forward-backward splitting (or proximal gradient) method is a general framework for minimizing objective functions with two terms [17]. For the problem (11), the method alternates between gradient descent steps that only act on the smooth terms in (11), and a backward/proximal step that only acts on the nuclear norm term. The gradient of the (smoothed) proximal regularizer in (11) is given column-wise (i.e., image-wise) by

$$\nabla J_\epsilon(\mathbf{X}_t) = \sum_{c \in \mathcal{C}} \mathbf{X}_{t,c} \|\mathbf{X}_{t,c}\|_{2,\epsilon}^{-1}. \quad (13)$$

The gradient formula (13) requires the computation of the sum (12) for every clique  $c$ , and then, a summation over the reciprocals of these sums; this is potentially expensive if done in a naïve way. Fortunately, every block sum can be computed simultaneously by squaring all of the entries in  $\mathbf{X}$ , and then convolving the result with a block filter. The result of this convolution contains the value of  $\|\mathbf{X}_{t,c}\|_{2,\epsilon}^2$  for all cliques  $c$ . Each entry in the result is then raised to the  $-1/2$  power, and convolved again with a block filter to compute the entries in the gradient (13). Both of these

---

**Algorithm 1** Forward-backward proximal minimization

---

**Input:**  $\mathbf{Y}, \mu > 0, \lambda, \mathcal{C}_i, \alpha > 0$   
**Initialize:**  $\mathbf{X}^{(0)} = \mathbf{0}, \mathbf{Z}^{(0)} = \mathbf{0}$   
**Output:**  $\mathbf{X}^{(n)}, \mathbf{Z}^{(n)}$

- 1: **while** not converged **do**
  - 2:   **Step 1:** Forward gradient descent on  $X$ ,
  - 3:    $\mathbf{X}_k^{(n)} = \mathbf{X}_k^{(n-1)} - \alpha\lambda\nabla J_\epsilon(\mathbf{X})$   
           $+ \alpha\mu(\mathbf{Y}_k - \mathbf{Z}_k^{(n-1)} - \mathbf{X}_k^{(n-1)})$
  - 4:   **Step 2:** Forward gradient descent on  $\mathbf{Z}$ ,
  - 5:    $\mathbf{Z}_k^{(n)} = \mathbf{Z}_k^{(n-1)} + \alpha\mu(\mathbf{Y}_k - \mathbf{Z}_k^{(n-1)} - \mathbf{X}_k^{(n-1)})$
  - 6:   **Step 3:** Backward gradient descent on  $\mathbf{Z}$ ,
  - 7:    $\mathbf{Z}^{(n)} = \text{prox}_*(\mathbf{Z}^{(n)}, \alpha)$
  - 8: **end while**
- 

two convolution operations can be computed quickly using fast Fourier transforms (FFTs), so that the computational complexity becomes independent of clique size.

Algorithm 1 shows the pseudocode for solving (11). In Steps 1 and 2, the values of  $\mathbf{X}$  and  $\mathbf{Z}$  are updated using gradient descent on (11), ignoring the nuclear norm regularizer. Step 3 accounts for the nuclear-norm term using its proximal mapping, which is given by

$$\text{prox}_*(\mathbf{Q}, \delta) = \mathbf{U}(\text{sign}(\mathbf{S}) \circ \max\{|\mathbf{S}| - \delta, 0\})\mathbf{V}^T,$$

where  $\mathbf{Q} = \mathbf{U}\mathbf{S}\mathbf{V}^T$  is a singular value decomposition of  $\mathbf{Q}$ ,  $|\mathbf{S}|$  denotes element-wise absolute value, and  $\circ$  denotes element-wise multiplication.

The forward-backward splitting (FBS) procedure in Algorithm 1 is known to converge for sufficiently small stepsizes  $\alpha$  [3]. Practical implementations of FBS 1 include adaptive stepsize selection [36], backtracking line search, or acceleration [3]. We use the FASTA solver from [17], which combines such acceleration techniques.

We note that FBS 1 only requires a total of  $4NL$  storage entries for  $\mathbf{X}$ ,  $\mathbf{Y}$ ,  $\mathbf{Z}$  and gradient  $\nabla J_\epsilon(\mathbf{X})$ . However, in order to solve RPCA formulation using ADMM we require  $2l^2NL$  storage entries for auxiliary variables (as discussed before) and  $4NL$  storage entries for the variables  $\mathbf{X}$ ,  $\mathbf{Y}$ ,  $\mathbf{Z}$  and dual variable of  $\mathbf{Y} = \mathbf{X} + \mathbf{Z}$ , leading to total of  $(2l^2 + 4)NL$  storage entries. Since the memory usage and runtime of FBS is independent of the clique size, the advantage of FBS over ADMM is much greater for larger cliques.

### 2.2.3 Matching Pursuit Algorithm

For compressive-sensing problems involving large random matrices, *matching pursuit* algorithms (such as CoSaMP [30]) are an important class of sparse recovery methods. When signals have structured support, model-based matching pursuit routines have been proposed that require non-convex minimizations over Markov random fields [9]. In this section, we propose a model-based matching pursuit algorithm that achieves structured compressive

---

**Algorithm 2** CoLaMP - Convex Lattice Matching Pursuit

---

**Input:**  $\mathbf{y}, \Phi, K, \lambda, \epsilon$   
**Initialize:**  $\mathbf{x}^{(0)} = \mathbf{0}, \mathbf{s}^{(0)} = \mathbf{0}, \mathbf{r}^{(0)} = \mathbf{y}$   
**Output:**  $\mathbf{x}^{(n)}$

- 1: **while**  $n \leq \text{max.iterations}$  and  $\|\mathbf{r}^{(n)}\|_2 > \epsilon$  **do**
  - 2:   **Step 1:** Form temporary target signal
  - 3:    $\mathbf{v}^{(n)} \leftarrow \Phi^T \mathbf{r}^{(n-1)} + \mathbf{x}^{(n-1)}$
  - 4:   **Step 2:** Refine signal support using convex prior
  - 5:    $\mathbf{x}_r^{(n)} = \arg \min_{\mathbf{x}} \|\mathbf{x} - \mathbf{v}^{(n)}\|_2^2 + \lambda J(x)$ ,
  - 6:    $\mathbf{s} \leftarrow \text{supp}(\mathbf{x}_r^{(n)})$
  - 7:   **Step 3:** Estimate target signal
  - 8:   Solve  $\Phi_s^T \Phi_s \mathbf{x}_s = \Phi_s^T \mathbf{y}$ , with  $\Phi_s = \Phi(:, \mathbf{s})$
  - 9:   Set all but largest  $K$  entries in  $\mathbf{x}_s$  to zero,
  - 10:    $\mathbf{x}^{(n)}(s) = \mathbf{x}_s(s)$
  - 11:   **Step 4:** Calculate data residual
  - 12:    $\mathbf{r}^{(n)} \leftarrow \mathbf{y} - \Phi \mathbf{x}^{(n)}$
  - 13:    $n \leftarrow n + 1$
  - 14: **end while**
- 

signal recovery using *convex* sub-steps for which global minimizers are efficiently computable.

The proposed method, Convex Lattice Matching Pursuit (CoLaMP), is a greedy algorithm that attempts to solve

$$\begin{aligned} \hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \|\Phi \mathbf{x} - \mathbf{y}\|_2^2 + \lambda J(\mathbf{x}) \\ \text{subject to } \|\mathbf{x}\|_0 \leq K. \end{aligned} \quad (14)$$

The complete method is listed in Algorithm 2. In Step 1, CoLaMP proceeds like other matching pursuit algorithms; the unknown signal is estimated by multiplying the residual by the adjoint of the measurement operator. In Step 2, this estimate is refined by solving a support regularized problem of the form (7). We solve this problem either via ADMM or the FBS method in Algorithm 1). In Step 3, a least-squares (LS) problem is solved to identify the signal that best matches the observed data, assuming the correct support was identified in Step 2. This LS problem is solved by a conjugate gradient method. Finally, in Step 4, the residual (the discrepancy between  $\Phi \mathbf{x}$  and the data vector  $\mathbf{y}$ ) is calculated. The algorithm is terminated if the residual becomes sufficiently small or a maximum number of iterations is reached.

CoLaMP has several desirable properties. First, the support set regularization (Step 2) helps to prevent signal support from growing quickly, and thus minimizes the cost of the least-squares problem in Step 3. Secondly, the use of a convex prior guarantees that a global minimum is obtained for every subproblem in Step 2, regardless of the considered clique structure. This is in stark contrast to other model-based recovery algorithms, such as LaMP<sup>1</sup>, and model-based CoSaMP [2], which requires the solution to non-convex optimization problems to enforce structured support set models.

---

<sup>1</sup>It is possible to restrict LaMP to planar Ising models, in which case a global optimum is computable [9].

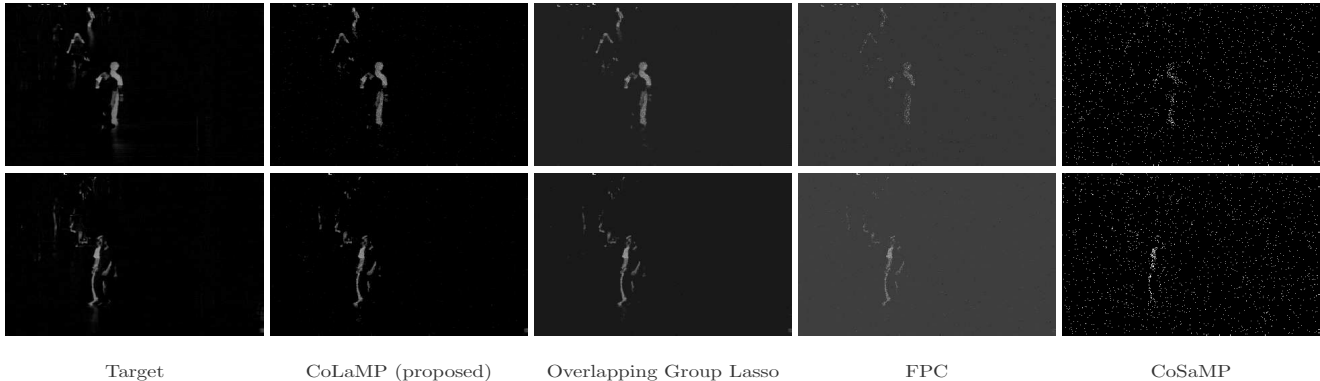


Figure 2: Compressed sensing recovery results for background subtracted images using  $M = 3K$ .

### 3. Numerical Experiments

We now apply the proposed regularizer to a range of datasets to demonstrate its efficacy for various applications. Unless stated otherwise, we showcase our algorithms using overlapping cliques of size  $2 \times 2$  as shown in Figure 1(b). Note that the numerical algorithms need not be restricted to those discussed above as different schemes (such as primal-dual decomposition) are needed for different situations.

#### 3.1. Compressive Image Recovery

We first consider the recovery of background-subtracted images from compressive measurements. We use the “walking2” surveillance video data [37] with frames of dimension  $288 \times 384$ . Test data is generated by choosing two frames from a video sequence and computing the pixel-wise difference between their intensities. We compare the output of our proposed CoLaMP algorithm to that of other state-of-the-art recovery algorithms, such as overlapping group lasso [24], fixed-point continuation (FPC) [18] and CoSaMP [30]. Note that CoSaMP defines the support set using the  $2K$  largest components of the error signal. The group lasso algorithm is equivalent to minimizing the objective in (14) using variational method. Unlike the CoLaMP algorithm, this method does not consider prescribed signal sparsity  $K$ . An example recovery using  $M = 3K$  measurements is shown in Figure 2. The sparsity level  $K$  is chosen such that the recovered images account for 97% of the compressive signal energy. The average  $K$  across datasets is 2800 and we fix  $\lambda = 2$ . Note that the spatially clustered pixels are recovered almost perfectly. Further, we randomly generated 50 such test images from the above dataset and compared the performance of the CoLaMP, group lasso, and FPC algorithms under varying numbers of measurements from  $1K$  to  $5K$ . The performance is measured in terms of the magnitude of reconstruction error normalized by the original image magnitude. Results are shown in Figure 4 (left). We clearly see that the proposed smooth sparsity prior significantly improves the

reconstruction quality over FPC. Furthermore, our algorithm is  $7 \times$  faster than the group lasso algorithm. For  $M/K = 3$ , the average runtime is 215s for CoLaMP and 1510s for the group lasso algorithm.

#### 3.2. Robust Signal Recovery

We next showcase the suitability of CoLaMP for signal recovery from noisy compressive measurements. We consider a  $100 \times 100$  Shepp—Logan phantom image with a support size of  $K = 2636$ . A Gaussian random measurement matrix was used to sample  $M = 2K$  measurements, and the measurements were corrupted with additive white Gaussian noise. The signal-to-noise ratio of the resulting measurements is 10 dB. Figure 3 shows the original and recovered images for various recovery algorithms. We also show the output from the first few iterates of the CoLaMP algorithm. The support of the target signal is almost exactly recovered within four iterations of CoSaMP and stabilizes by the end of 10 iterations. Figure 3 also shows the recovery times of various algorithms running on the same laptop computer. CoLaMP is approximately  $40 \times$  faster than the CoSaMP algorithm and it is at least  $2 \times$  faster than FPC.

To enable a fair comparison, we also show the output obtained with CoLaMP using the 8-connected pixel clique in Figure 1(c), as well as the output of the group lasso algorithm [40], where each clique is of size  $2 \times 2$ . All these algorithms and our proposed method are implemented using ADMM. Not surprisingly, while all these algorithms beat CoLaMP in terms of runtime, their recovered signals do not match CoLaMP in terms of perceived closeness to target signal as shown in Figure 3. The CoLaMP results are regularized by  $\lambda_0 = 16$ . We then used an increasing value of  $\lambda_n = 1.02^n \lambda_0$  where  $n$  is the iteration number. In practice, we obtain better results if  $\lambda$  increases over time as it will heavily penalize sparse, blocky noise. For all other algorithms, we used the implementations provided by the authors.

For detailed quantitative comparisons, we repeat the above experiment using 100 Gaussian random measurement

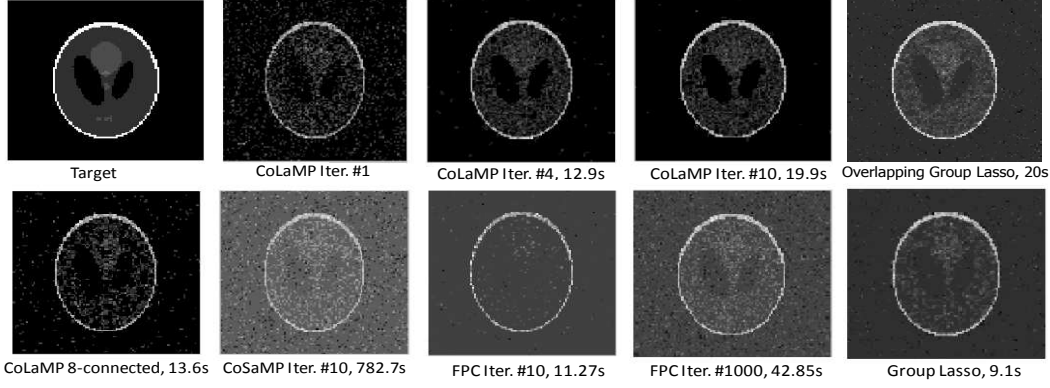


Figure 3: Robust recovery results for the phantom image from a noisy compressed signal.

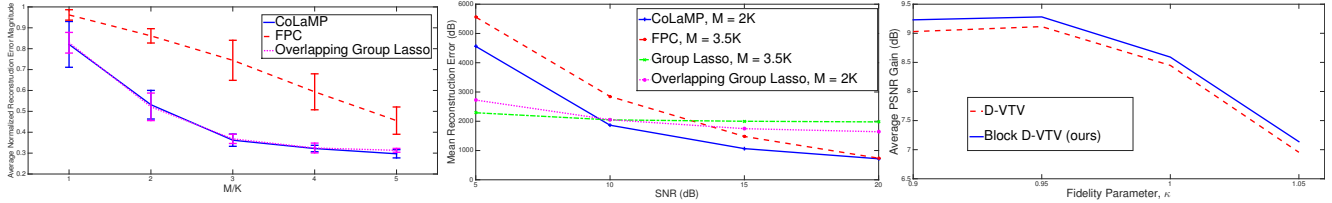


Figure 4: Quantitative Comparison: (left) Recovery performance of compressed sensing on background subtracted images; (center) Robust compressed sensing recovery error at various SNR; (right) Average denoising gain in PSNR (dB) for various values of  $\kappa$

matrices and record the average reconstruction error with SNR varying from 5 dB to 20 dB. For each algorithm,  $M$  is fixed to the minimal measurement number required to give close to perfect recovery in the presence of noise. For CoLaMP and overlapping group lasso, we set  $M = 2K$ , whereas for FPC and non-overlapping group lasso we set  $M = 3.5K$ . Figure 4 (center) illustrates that CoSaMP outperforms FPC at all SNRs even with  $1.5K$  fewer measurements. Group lasso performs best at low SNR while its performance flattens out starting at 10 dB.

### 3.3. Color Image Denoising

We now consider a variant of the denoising problem (5) where the image gradient is defined over color images using the decorrelated vectorized TV (D-VTV) proposed in [31]

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathbb{R}^{3N}} \sum_{c \in C} \lambda \|\nabla_d \mathbf{x}_c^\ell\|_2 + \|\nabla_d \mathbf{x}_c^{ch}\|_2$$

$$\text{subject to } \|\mathbf{x} - \mathbf{y}\|_2 \leq \kappa m. \quad (15)$$

Here,  $\nabla_d \mathbf{x}^\ell \in \mathbb{R}^{2N}$  and  $\nabla_d \mathbf{x}^{ch} \in \mathbb{R}^{4N}$  represent the stacked gradients of luminance and chrominance channels of the input color image, the constant  $m$  depends on the noise level, and  $\kappa$  is a fidelity parameter. To solve this problem numerically, we use the primal-dual algorithm described in [31], but we replace the shrinkage operator with the proximal operator (7) to adapt our clique-based regularizer.

Following a protocol similar to D-VTV [31], we conduct experiments using 300 images from the Berkeley Segmentation Database [29]. Noisy images with average PSNR 20 dB are obtained by adding white Gaussian noise. The resulting denoised output of our method (Block D-VTV) is

compared to D-VTV in Figure 5. The zoomed-in version reveals that our method exhibits less uneven color artifacts and less pronounced staircasing artifacts than the D-VTV results. A quantitative comparison measured using average PSNR gain (in dB) is drawn in Figure 4 (right) for various values of  $\kappa$ . Our method outperforms D-VTV by 0.25 dB. Also note that our method, Block D-VTV, obtains relatively better PSNR gain than the state-of-the-art D-VTV method at smaller values of  $\kappa$ . This observed gain is significant because smaller  $\kappa$  values lead to a tighter fidelity constraint and thus a smaller solution space around the noisy input. In such situations, Block D-VTV helps to improve image quality by leveraging input from neighboring pixels.

### 3.4. Video Decomposition

We finally consider the robust PCA (RPCA) problem for structured sparsity of size  $10 \times 10$  as formulated in (11) and using Algorithm 2. We consider the same airport surveillance video data [25] as in [7] with frames of dimension  $144 \times 176$ . For a clique formed from  $l \times l$  patches, we observed that  $\lambda = 1/(l\sqrt{n_1})$  works best for our experiments as opposed to  $\lambda = 1/\sqrt{n_1}$  used in [7]. This is because each element of the matrix  $\mathbf{X}$  is shared by  $l^2$  sparsity inducing terms. The resulting low rank components (background) and foreground components of three such example video frames are shown in Figure 6. For all the approaches, the low rank components are nearly identical. We observe that the rank of the low-rank component remains the same. As highlighted with the green box, the noisy sparse edges appearing in the original RPCA disappear from the foreground component using our proposed method. We also display the foreground



Figure 5: Restoration of noisy images using Block D-VTV and existing D-VTV (best viewed in color).

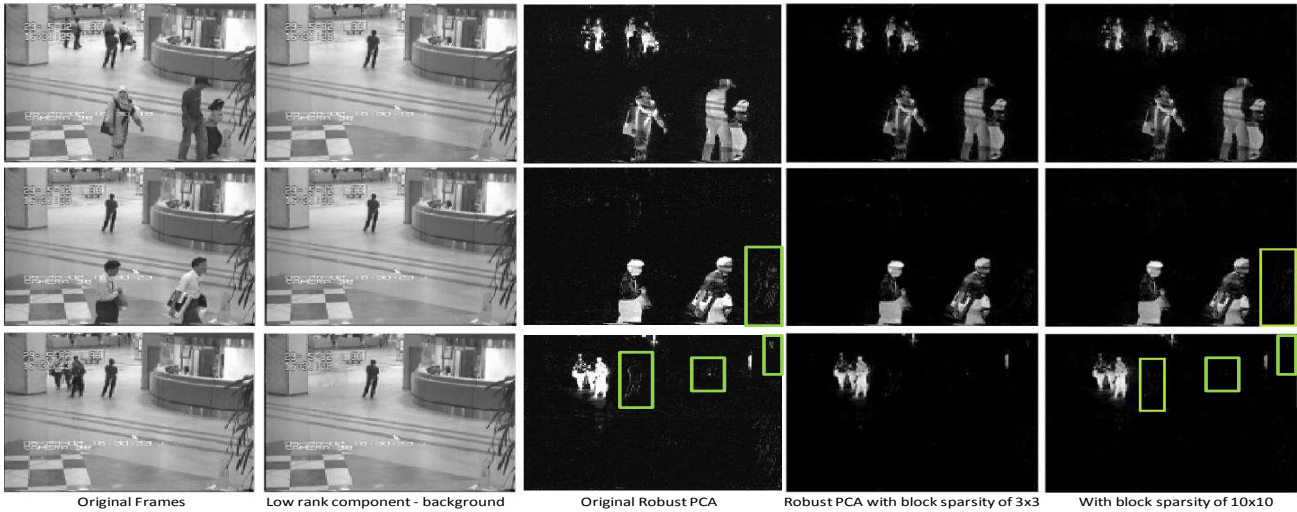


Figure 6: Sparse-and-low-rank decomposition using original robust PCA and proposed approach.

component obtained using smaller overlapping cliques of size  $3 \times 3$ , but solved using ADMM as opposed to forward-backward splitting (Algorithm 1). We found that for clique size of  $10 \times 10$  the ADMM method becomes intractable because it requires approximately  $50\times$  more memory than the proposed forward-backward splitting method with fast convolutions (i.e.,  $204NL$  vs.  $4NL$ ).

#### 4. Conclusions

We have proposed a novel structured support regularizer for convex sparse recovery. Our regularizer can be applied to a variety of problems, including sparse-and-low-rank decomposition and denoising. For compressive signal recovery using large unstructured matrices, our convex regularizer can be used to improve the recovery quality of existing matching-pursuit algorithms. Compared to existing algorithms for this task, our proposed approach enjoys the capability of fast signal reconstruction from fewer measurements while ex-

hibiting superior robustness against spurious artifacts and noise. For color image denoising, the restored images reveal more homogeneous color effects. For robust PCA, we achieve improved foreground-background separation with far fewer artifacts. We envision many more applications that could benefit of the proposed regularizer, including deblurring and inpainting. More sophisticated directions include using support regularization for structured dictionary learning [41] and multitask classification.

#### Acknowledgments

The work of S. Shah and T. Goldstein was supported in part by the US National Science Foundation (NSF) under grant CCF-1535902 and by the US Office of Naval Research under grant N00014-15-1-2676. The work of C. Studer was supported in part by Xilinx Inc., and by the US NSF under grants ECCS-1408006 and CCF-1535897.



## References

- [1] F. Bach, R. Jenatton, J. Mairal, G. Obozinski, et al. Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, pages 19–53, 2011. 1, 2
- [2] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde. Model-based compressive sensing. *IEEE Transactions on Information Theory*, 56(4):1982–2001, 2010. 1, 3, 5
- [3] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009. 5
- [4] 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. 3, 4
- [5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, 2001. 1
- [6] X. Bresson and T. F. Chan. Fast dual minimization of the vectorial total variation norm and applications to color image processing. *Inverse Problems and Imaging*, 2(4):455–484, 2008. 3
- [7] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011. 3, 7
- [8] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006. 3
- [9] V. Cevher, M. F. Duarte, C. Hegde, and R. Baraniuk. Sparse signal recovery using markov random fields. In *NIPS*, pages 257–264, 2009. 1, 3, 5
- [10] V. Cevher, A. Sankaranarayanan, M. F. Duarte, D. Reddy, R. G. Baraniuk, and R. Chellappa. Compressive sensing for background subtraction. In *ECCV 2008*, pages 155–168. 1
- [11] C.-C. Cheng, G.-J. Peng, and W.-L. Hwang. Subband weighting with pixel connectivity for 3-d wavelet coding. *IEEE Transactions on Image Processing*, 18(1):52–62, 2009. 2
- [12] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 24(5):603–619, 2002. 1
- [13] W. Deng, W. Yin, and Y. Zhang. Group sparse optimization by alternating direction method. In *SPIE Optical Engineering+ Applications*, pages 88580R–88580R. International Society for Optics and Photonics, 2013. 3, 4
- [14] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004. 1
- [15] Z. Gao, L.-F. Cheong, and M. Shan. Block-sparse rpca for consistent foreground detection. In *Computer Vision–ECCV 2012*, pages 690–703. Springer, 2012. 3
- [16] T. Goldstein and S. Osher. The split Bregman method for  $l_1$ -regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009. 3, 4
- [17] T. Goldstein, C. Studer, and R. G. Baraniuk. A field guide to forward-backward splitting with a FASTA implementation. *CoRR*, abs/1411.3406, 2014. 4, 5
- [18] E. T. Hale, W. Yin, and Y. Zhang. A fixed-point continuation method for  $l_1$ -regularized minimization with applications to compressed sensing. *CAAM TR07-07*, Rice University, 2007. 6
- [19] J. Huang, T. Zhang, and D. Metaxas. Learning with structured sparsity. *JMLR*, 12:3371–3412, 2011. 1
- [20] L. Jacob, G. Obozinski, and J.-P. Vert. Group lasso with overlap and graph lasso. In *ICML*, pages 433–440. ACM, 2009. 1, 2, 3
- [21] R. Jenatton, J.-Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. *JMLR*, 12:2777–2824, 2011. 1, 2
- [22] R. Jenatton, G. Obozinski, and F. Bach. Structured sparse principal component analysis. *arXiv preprint arXiv:0909.1440*, 2009. 1
- [23] L. A. Jeni, A. Lőrincz, Z. Szabó, J. F. Cohn, and T. Kanade. Spatio-temporal event classification using time-series kernel based structured sparsity. In *ECCV 2014*, pages 135–150. 1, 2
- [24] M. Leigsnering, F. Ahmad, M. Amin, and A. Zoubir. Multipath exploitation in through-the-wall radar imaging using sparse reconstruction. *Aerospace and Electronic Systems, IEEE Transactions on*, 50(2):920–939, 2014. 2, 6
- [25] L. Li, W. Huang, I.-H. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing*, 13(11):1459–1472, 2004. 7
- [26] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *PAMI*, 35(1):171–184, 2013. 4
- [27] J. Liu, T.-Z. Huang, I. W. Selesnick, X.-G. Lv, and P.-Y. Chen. Image restoration using total variation with overlapping group sparsity. *Information Sciences*, 295:232–246, 2015. 3
- [28] J. Mairal, R. Jenatton, F. R. Bach, and G. R. Obozinski. Network flow algorithms for structured sparsity. In *Advances in Neural Information Processing Systems*, pages 1558–1566, 2010. 3
- [29] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV 2001*, volume 2, pages 416–423. 7
- [30] D. Needell and J. A. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009. 3, 5, 6
- [31] S. Ono and I. Yamada. Decorrelated vectorial total variation. In *CVPR 2014*, pages 4090–4097. 3, 7
- [32] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992. 3, 4
- [33] I. W. Selesnick and P.-Y. Chen. Total variation denoising with overlapping group sparsity. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 5696–5700. IEEE, 2013. 3
- [34] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000. 1
- [35] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007. 3

- [36] S. J. Wright, R. D. Nowak, and M. A. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493, 2009. 5
- [37] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *IEEE CVPR*, 2013. 6
- [38] J. Yao, X. Liu, and C. Qi. Foreground detection using low rank and structured sparsity. In *Multimedia and Expo (ICME), 2014 IEEE International Conference on*, pages 1–6. IEEE, 2014. 3
- [39] L. Yuan, J. Liu, and J. Ye. Efficient methods for overlapping group lasso. In *NIPS*, pages 352–360, 2011. 3
- [40] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006. 2, 6
- [41] Y. Zhang, Z. Jiang, and L. S. Davis. Learning structured low-rank representations for image classification. In *CVPR, 2013*, pages 676–683. 8