

# Partitioned Compressive Sensing with Neighbor-Weighted Decoding

H.T. Kung and Stephen J. Tarsa  
Harvard University, Cambridge, MA

**Abstract**—Compressive sensing has gained momentum in recent years as an exciting new theory in signal processing with several useful applications. It states that signals known to have a sparse representation may be encoded and later reconstructed using a small number of measurements, approximately proportional to the signal’s sparsity rather than its size. This paper addresses a critical problem that arises when scaling compressive sensing to signals of large length: that the time required for decoding becomes prohibitively long, and that decoding is not easily parallelized.

We describe a method for *partitioned compressive sensing*, by which we divide a large signal into smaller blocks that may be decoded in parallel. However, since this process requires a significant increase in the number of measurements needed for exact signal reconstruction, we focus on mitigating artifacts that arise due to partitioning in approximately reconstructed signals. Given an error-prone partitioned decoding, we use large magnitude components that are detected with highest accuracy to influence the decoding of neighboring blocks, and call this approach *neighbor-weighted decoding*. We show that, for applications with a predefined error threshold, our method can be used in conjunction with partitioned compressive sensing to improve decoding speed, requiring fewer additional measurements than unweighted or locally-weighted decoding.

## I. INTRODUCTION

The theory of compressive sensing states that, for a signal  $X$  of length  $N$  that is known to have a representation with at most  $k$  non-zero elements,  $X$  can be encoded as follows:

$$Y = \Phi X$$

where  $\Phi$  is an  $M \times N$  randomized measurement matrix used to form linear combinations of  $X$ . The result is a vector of measurements  $Y$  of length  $M$ , where  $M \ll N$ , and  $M$  is approximately proportional to  $k$ .  $X$  can later be reconstructed (or “decoded”) by finding a coefficient vector  $\hat{S}$  in the basis of sparse representation  $\Psi$  with minimum  $\mathcal{L}^1$ -norm:

$$\min_{Y=\Phi\Psi\hat{S}} \|\hat{S}\|_1$$

In words: if a  $k$ -sparse representation of a signal is known to exist in *some* basis, reconstruction of the signal can be achieved based on a small number of measurements relative to the signal size. In practical scenarios,  $X$  is often not exactly sparse, but has  $k$  components with large magnitude from which an approximation of  $X$  may be reconstructed [1].

The implication of compressive sensing is that data acquisition devices can enjoy reduced output data rates without the need for heavy computation at encoding, even if the domain

of sparse representation is not known *a priori*. For example, wireless sensor networks may realize large bandwidth savings without media-specific compression hardware built into sensors, lowering device cost, and reducing power consumption [2]. When linear combinations happen in the analog domain, as authors have interpreted wireless packet collisions [3], compressive sensing is a natural fit for high speed data compression. In a different context, exploiting the sparsity of Magnetic Resonance Images (MRIs) reduces imaging time [4]. Applications beyond traditional signal and image processing are starting to appear, such as using compressive sensing for continuous fine-grain status monitoring in data centers for anomaly detection [5], and to reduce measurements for off-chip analysis in IC Trojan detection [6].

However, a major limitation is that the computational cost of decoding scales poorly. Several algorithms for decoding are commonly used, such as  $\mathcal{L}^1$  minimization via a simplex method in linear programming. This algorithm has the tightest error bound, but has an empirical running time of  $O(N^3)$  [7] [8], which is prohibitively expensive for large  $N$ . Another method is a greedy adaptation of Orthogonal Matching Pursuit (OMP), which has a larger error bound and sometimes fails under poor initial approximations, but a running time of  $O(N^2 \log N)$  [9]. CoSamp improves upon OMP with more rigorous error bounds, and a running time of  $O(N \log^2 N)$  [10]. In some cases, application specific algorithms such as *min total variation* may be used to improve performance [11] [4]. While trading off factors like accuracy and robustness to poor initial approximations improves decoding time, we consider the running time to be  $O(N^3)$ , since low-complexity methods must resort to linear programming upon failure.

In this paper, we present a method for *partitioned compressive sensing* that can be used along with any solver to reduce computational costs and make decoding amenable to parallel processing. Specifically, the signal  $X$  is partitioned into  $B$  blocks of size  $N/B$  by encoding with a block diagonal sensing matrix  $\Phi_B$ . This produces  $B$  independent sets of measurements,  $y_1, \dots, y_B$  as follows:

$$\begin{bmatrix} y_1 \\ \vdots \\ y_B \end{bmatrix} = \begin{bmatrix} \phi_1 & & \\ & \ddots & \\ & & \phi_B \end{bmatrix} X$$

At decoding, solution vectors  $\hat{s}_i$  of length  $N/B$  are decoded by solving  $B$  independent problems:

$$\min_{y=\Phi_i\Psi_i\hat{s}_i} \|\hat{s}_i\|_1 \quad i = 1 \dots B,$$

where the  $\Psi_i$ 's are bases of reduced dimension, e.g. in the case of Fourier-style transforms,  $N/B$ -point transforms (throughout this paper, we will use DCT-II transforms for illustration, though results generalize to any choice of basis). After solving each problem independently, a decoded signal of length  $N$  is obtained by concatenating independent blocks  $\hat{x}_i = \Psi_i\hat{s}_i$ :

$$\hat{X}_B = [\hat{x}_1, \dots, \hat{x}_B] \text{ for } i = 1 \dots B$$

When used in combination with  $\mathcal{L}^1$  minimization via a linear program, partitioning reduces running time to  $O(B(\frac{N}{B})^3)$ . The reduction in decoding time can be significant; for example when  $N = 10,000$  and  $B = 10$ , the running time is reduced by  $10^3$ , or three orders of magnitude.

Ideally, we would like  $\hat{X}_B$  to equal  $\hat{X}$ , the signal recovered without partitioning. We can expect this lucky case to only arise when each block can be exactly recovered, meaning the total number of measurements must be increased by a factor of  $B$ . That is, for the  $M \times N$  measurement matrix,  $M$  becomes  $M_B$ , on the order of  $B * k \log(\frac{N}{Bk})$ , instead of  $k \log(\frac{N}{k})$ . Ultimately, although a huge reduction in decoding time is achieved, increasing the number of measurements by approximately a factor of  $B$  swallows up the data-reduction gains afforded by compressive sensing (In [12], for applications like target detection tasks, one can use a progressive reconstruction algorithm to overcome the problem of increased measurements due to partitioning). Therefore, we focus on approximate recovery of blocks, with the goal of taking as few additional measurements as possible in order to meet a predetermined error threshold for decoded signals.

To summarize our contributions: (1) We identify an incompatibility problem at the partition boundaries of reconstructed signals, the presence of which we call a ‘‘blocking artifact’’ in this paper (see Section II). This problem is due to inconsistency in approximate recovery of neighboring blocks. To our knowledge, we are the first in the literature to identify this incompatibility issue for partitioned compressive sensing. (2) We propose *neighbor-weighted decoding* for mitigating blocking artifacts (Section III). This method is encoding-blind in the sense that it does not require comprehension of the input signal. However, during decoding, our method can adapt to the results of decoded neighboring blocks in order to maximize their compatibility along partition boundaries. Adaptation is implemented by importing neighbors’ dominant components and weighting those components in decoding. (3) We suggest a performance metric for evaluating incompatibility mitigation schemes (Section IV) and relate it to perceptual artifacts that result from blocking artifacts. We use this metric to show the effectiveness of neighbor-weighted decoding.

## II. RECOVERY INCOMPATIBILITY IN PARTITIONED COMPRESSIVE SENSING

In general, when the number of measurements  $M$  is not large enough to support exact recovery, decoding is lossy;

only coefficients with the largest magnitudes can be recovered, either exactly or approximately [13]. We term this set of coefficients for a fixed  $M$  the *dominant coefficients* of the signal, and their corresponding signal components *dominant components*, or in the cases where  $\Psi$  is a Fourier transform, *dominant frequencies* (in this paper, we use these terms interchangeably depending on context). Furthermore, during decoding, information is lost when the smallest coefficients are erroneously truncated to 0 magnitude, and we call this set the *tail coefficients*, and their corresponding components *tail components/frequencies*. With these definitions, it is known that the decoding quality under partitioning will be determined by the characteristics of the two sets [13].

### A. Incompatibility and Blocking Artifacts

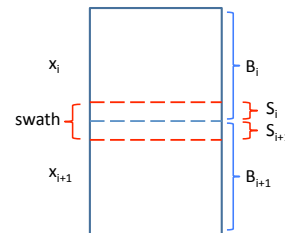


Fig. 1. Two neighboring blocks  $x_i$  and  $x_{i+1}$ , and a swath centered at the partition boundary.  $B_i$ ,  $S_i$ ,  $B_{i+1}$ , and  $S_{i+1}$  denote sets of the dominant components in their respective regions.

First, we define a desired state of *compatibility* that exists when  $\hat{X}_B$  is the same as the  $\hat{X}$  that would have been recovered without partitioning, up to an error threshold. In scenarios where  $\hat{X}_B$  is incompatible with  $\hat{X}$ , inconsistent recovery arising from partitioned decoding, such as the loss of certain frequencies, is most apparent in the form of *blocking artifacts* along a swath at the boundary between two blocks. Note that our use of the term blocking artifacts refers specifically to errors that arise due to partitioned compressive sensing, and not to abrupt transitions between blocks that the original image may already exhibit. Thus we are interested in mitigating blocking artifacts induced by the compressive sensing recovery process that are not present in the original image.

Here we will precisely state our goal and motivate our solution approach. Consider two neighboring blocks,  $x_i$  and  $x_{i+1}$ , and a swath along the common boundary of the two blocks, as depicted in Figure 1. In the figure, the blue dashed line denotes the partition boundary between blocks, and the red lines bound the swath. For block  $x_i$ , we define  $B_i$  as a set of some of the largest components over the entire block, and also define  $S_i$  as a set of some of the largest components local to the region where the swath overlaps  $x_i$ . Similarly, we define  $B_{i+1}$  and  $S_{i+1}$  for block  $x_{i+1}$ . Note that a smooth transition between blocks  $x_i$  and  $x_{i+1}$  suggests overlap between the sets  $S_i$  and  $S_{i+1}$ . We assume that all of these sets,  $B_i$ ,  $S_i$ ,  $B_{i+1}$ , and  $S_{i+1}$  have enough components to represent acceptable approximations of their regions.

The fact that the original image does not have abrupt transitions between blocks means that components in  $S_{i+1}$

must be in  $B_i$ , i.e.  $S_{i+1} \subseteq B_i$ . Otherwise, those components of  $S_{i+1}$  not in  $B_i$  would contribute to the boundary region of  $x_i$ , but not to the boundary region of  $x_{i+1}$ , resulting in an abrupt transition. For a similar reason, we must have  $S_i \subseteq B_{i+1}$ . Thus we have the assumptions:

- (1)  $S_{i+1} \subseteq B_i$  and  $S_{i+1} \subseteq B_{i+1}$
- (2)  $S_i \subseteq B_{i+1}$  and  $S_i \subseteq B_i$

Moreover, adjacent regions on either side of the partition boundary must have compatible leading components:

- (3)  $S_i = S_{i+1}$

However, the relationships in (1), (2), and (3) may not hold for recovered signals. We use  $\hat{B}_i$ , and  $\hat{B}_{i+1}$ , to denote the sets of dominant components in recovered versions of both blocks,  $\hat{x}_i$  and  $\hat{x}_{i+1}$ , based on a fixed number of measurements for each. For example, when there are insufficient measurements for block  $x_{i+1}$ ,  $\hat{B}_{i+1}$  may not contain all of the components of  $B_{i+1}$ . As a result, we may have  $S_{i+1} \not\subseteq \hat{B}_{i+1}$  and by (3) also  $S_i \not\subseteq \hat{B}_{i+1}$ . In this case, when the recovered signals are images, we will see blocking artifacts along the boundary.

When we increase the number of measurements, additional dominant components will be decoded, and  $\hat{B}_i$  and  $\hat{B}_{i+1}$  will approach  $B_i$ , and  $B_{i+1}$ . By assumptions (1) and (2), with sufficient measurements, we will have:

- (4)  $\hat{S}_{i+1} \subseteq \hat{B}_i$  and  $\hat{S}_{i+1} \subseteq \hat{B}_{i+1}$
- (5)  $S_i \subseteq \hat{B}_{i+1}$  and  $\hat{S}_i \subseteq \hat{B}_{i+1}$

We call properties (4) and (5) together the *compatibility condition*. In this paper, we address the challenge of satisfying this condition without a significant increase in measurements.

To illustrate the idea behind the proposed neighbor-weighted decoding approach, suppose

$$\begin{aligned} B_i &= \{f_1, f_a, f_b, f_2\} \\ B_{i+1} &= \{f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_a, f_b\} \\ S_i &= \{f_a, f_b\} \\ S_{i+1} &= \{f_a, f_b\} \end{aligned}$$

where each set is a sequence of components in decreasing order by magnitude. We can check that this example satisfies assumptions (1), (2), and (3). Now suppose that there are insufficient measurements for blocks  $x_i$  and  $x_{i+1}$  to warrant exact recovery. As a result, we recover only dominant components:

$$\begin{aligned} \hat{B}_i &= \{f_1, f_a, f_b\} \\ \hat{B}_{i+1} &= \{f_3, f_4, f_5, f_6, f_7, f_8\} \end{aligned}$$

Here, (4) and (5) do not hold and blocking artifacts arise.

Our neighbor-weighted decoding will mitigate this problem by adapting the decoding process for neighboring blocks. For example, for block  $x_{i+1}$ , we will import top components decoded from block  $x_i$  and weight them in decoding to assure

their appearance in  $\hat{B}_{i+1}$ . In the example, this corresponds to importing  $f_1, f_a$ , and  $f_b$  from  $\hat{B}_i$ , and using weighted decoding (Section III) to obtain

$$\begin{aligned} \hat{B}_{i+1} &= \{f_3, f_4, f_5, f_1, f_a, f_b\} \\ \hat{S}_{i+1} &= \{f_a, f_b\} \end{aligned}$$

to satisfy the compatibility condition.

The following theorem provides a condition for sufficiency on the number of measurements required by neighbor-weighted decoding to satisfy the compatibility condition. We say that a signal is  $(K, \epsilon)$ -sparse if only  $K$  out of its  $N$  components have a magnitude greater than a small  $\epsilon > 0$ . Suppose that  $M$  measurements are used to encode a signal, then we know from [14] that the reconstructed signal based on  $\mathcal{L}^1$  minimization has a small error when  $M > cK \log(N/K)$  for some positive constant  $c$ .

*Theorem.* Consider any two neighboring blocks  $x_i$  and  $x_{i+1}$ . Suppose that the signals for blocks  $x_i$  and  $x_{i+1}$  are of length  $N_B$  and are  $(K_i, \epsilon)$ -sparse and  $(K_{i+1}, \epsilon)$ -sparse respectively. Further, suppose that we use  $M_i$  and  $M_{i+1}$  measurements for decoding block  $x_i$  and  $x_{i+1}$ , with  $M_i = O(K_i \log(\frac{N_B}{K_i}))$  and  $M_{i+1} = O((K_i + K_{i+1}) \log(\frac{N_B}{K_i + K_{i+1}}))$ . Then under neighbor-weighted decoding, recovered signals for the blocks satisfy the compatibility condition.

*Proof:* We note that  $M_i$  measurements are sufficient to decode block  $x_i$  with enough accuracy so that the top  $K_i$  decoded components will include  $S_{i+1}$ , and also  $S_i$  by assumption (3). Thus, we have  $S_{i+1} \subseteq \hat{B}_i$  and  $S_i \subseteq \hat{B}_i$ . In decoding block  $x_{i+1}$ , we import all of these  $K_i$  components and weight them in the decoding process. We note that  $M_{i+1}$  measurements are sufficient to decode the top  $K_i + K_{i+1}$  components for block  $x_{i+1}$ , which include the imported  $K_i$  components of  $S_i$ . We have  $S_{i+1} \subseteq \hat{B}_{i+1}$  and  $S_i \subseteq \hat{B}_{i+1}$ . This achieves compatibility by (4) and (5). ■

We notice from the proof that importing needs to be carried out in only one direction when two neighboring blocks have different sparsities. We can further tighten the condition on  $M_{i+1}$  given in our theorem by letting  $M_{i+1} = O((K'_i + K_{i+1}) \log(\frac{N_B}{K'_i + K_{i+1}}))$  where  $K'_i$  is the number of those top  $K_i$  components for block  $x_i$  which are not in  $S_{i+1}$ .

### B. Causes of Blocking Artifacts

We next illustrate two cases where blocking artifacts can be introduced by partitioned compressive sensing.

**Case 1** The dominant frequencies of a block  $x_i$  differ from those of its neighbor  $x_{i+1}$ , even though there is a smooth transition between the two blocks in the space/time domain of the original signal. To be precise,  $B_i \neq B_{i+1}$  even though  $S_i = S_{i+1}$ . After applying partitioned compressive sensing with  $M_i$  and  $M_{i+1}$  insufficient to satisfy (4) and (5), assumption (3) breaks:  $\hat{S}_i \neq \hat{S}_{i+1}$ .

Figure 2 illustrates this scenario. The  $200 \times 100$  image is partitioned into equal sized top and bottom blocks. Examining a 20-pixel wide swath  $x_{SW}$  centered at the border between blocks, we see that columns on the right side of the swath would be dominated by low frequencies, even though all columns in the lower block are dominated by high frequency components, which are represented by fine stripes in the image. A partitioned decoding of the whole image with  $M_B = 25 * 200$  is shown, and as expected, low frequency components are lost in decoding, and only high frequency stripes are recovered, leading to a blocking artifact.

**Case 2** Neighboring blocks  $x_i$  and  $x_{i+1}$  have the same dominant frequencies, but the magnitudes of tail components in each block are very different. When these tail components are erroneously truncated during decoding due to insufficient measurements, *truncation error* is introduced, whereby dominant components are perturbed in an effort to fit the measurements  $y_i$  and  $y_{i+1}$ . When neighboring blocks are subject to different truncation errors, their dominant frequencies are perturbed differently, resulting in varying degrees of error for corresponding components. When these frequencies meet at the block boundary, we have the situation  $S_i = S_{i+1}$  but  $\hat{S}_i \neq \hat{S}_{i+1}$ , leading to a blocking artifact.

Figure 3 shows the spectra for two blocks with different tail behaviors: the top block is exactly sparse, so there is no truncation error, while the bottom block’s long tail leads to truncation error in dominant frequencies.

Both of these scenarios arise naturally when the composition of signals varies over time or space, and the incompatibility of partitioned decoding is a function of the input signal.

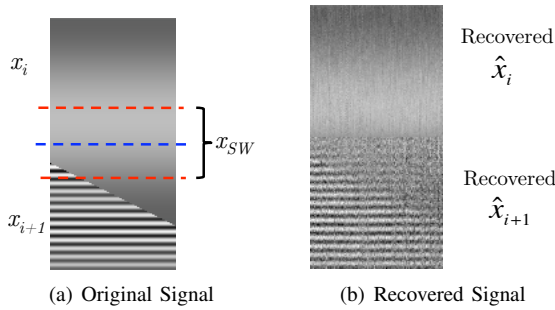
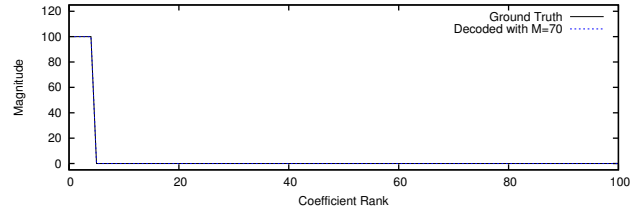


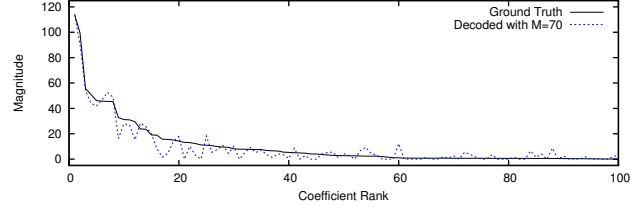
Fig. 2. An image sparse in the DCT domain that exhibits blocking artifacts when decoded via partitioned compressive sensing. (a) The image is partitioned on the blue dashed line into two blocks and decoded independently, treating each column as a separate signal. (b) In its recovered form, the lower block erroneously drops low frequencies, leading to a blocking artifact that is perceptually noticeable in the image.

### III. APPROACH, ANALYSIS, AND IMPLEMENTATION DETAILS FOR MITIGATING BLOCKING ARTIFACTS

The discussion and examples from the previous section lead to a strategy for mitigating blocking artifacts by influencing the coefficient vector found during the decoding process. However, first we must introduce *weighted decoding*, the tool we will use to realize such a strategy.



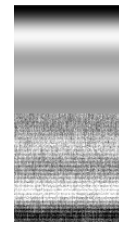
(a) Sorted Components of Block  $i$



(b) Sorted Components of Block  $i+1$



(c) Input signal



(d) Recovered signal

Fig. 3. Ground-truth sorted component magnitudes for a partitioned signal with the same dominant frequencies, but different tail behaviors. The blue dashed line shows recovered component magnitudes for  $M = 70$ . The top block (a) is exactly sparse and its dominant components are recovered exactly, while the bottom block (b) is approximately sparse, and truncation error causes the perturbation of dominant frequencies. (c) and (d) show the input and reconstructed images, respectively. Note in (d) that these errors create blocking artifacts at the partition boundary.

In weighted decoding, a diagonal matrix  $W$  with non-negative weights  $w_1, w_2, \dots, w_n$  along its diagonal is used to modify the constraints of the minimization step:

$$\min_{Y=\Phi\Psi W\hat{S}'} \|\hat{S}'\|_1$$

The resulting coefficient vector is then adjusted to recover  $\hat{S}$ , by  $\hat{S} = W\hat{S}'$ . This decreases the penalty for finding solutions with energy in components of interest, improving the accuracy of recovered solutions (after adjustment) with respect to these components, and reducing overall error. This approach was first introduced in [15], where iterative weighted decoding was used to “enhance” signals’ sparsities, akin to amplifying spikes and dampening noise. Other uses include [16], where weighting is used to combine measurements from distributed sensors for collaborative decoding.

In order to achieve compatibility and mitigate blocking artifacts, we import the top decoded components from neighboring blocks to apply weights during a second pass decoding. Returning to the example from Figure 2, we demonstrate how this process works: Figures 4(a) and 4(c) show the frequency spectra for the top and bottom blocks of a column in the

problematic right side of our example image. As observed in the image itself, the spectra confirm that the top block is dominated by a set of low frequency components, which we expect to be recovered first under a small number of measurements. The bottom block is dominated by higher frequencies, but still exhibits significant contributions from low frequencies. As seen previously, under small  $M$ , those low frequencies will be lost. Furthermore, Figures 4(b) and 4(d) show that the blocks have large tails, indicating that solutions will be prone to truncation errors. Differences in truncation errors will cause blocking artifacts as described in Case 2.

While completely eliminating blocking artifacts requires enough additional measurements for exact recovery, we will show that this process of *neighbor-weighted decoding* mitigates the problem, allowing us to remove most blocking artifacts with a smaller increase in measurements than locally-weighted or unweighted partitioned decoding. Figure 5 shows the result of our approach for the aforementioned problematic column. After neighbor-weighted decoding, the solution shown in red accurately captures low frequency components, and reduces the number of erroneous peaks assigned to low magnitude components.

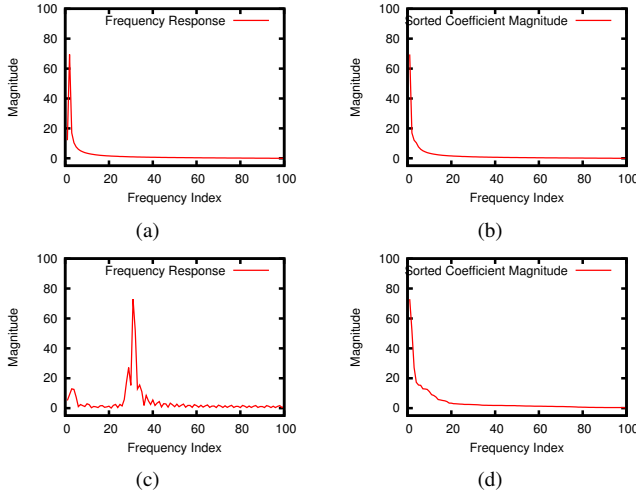


Fig. 4. (a) Component magnitudes by frequency and (b) sorted magnitude for the top block of a column with significant blocking artifacts under partitioned decoding. (c) and (d) show the corresponding frequency response and sorted component magnitudes for the bottom block, which erroneously drops low frequencies under small  $M$ . Notice that each block has different dominant frequencies, but that the bottom block also contains contributions from frequencies in the dominant set of the top block. Additionally, sorted magnitudes show that neither block is exactly sparse, indicating that differences in truncation error contribute to blocking artifacts.

#### A. Analysis of Gains Due to Neighbor-Weighted Decoding

We derive an expression for the reduction in measurements required to produce an approximately recovered signal free of blocking artifacts using neighbor-weighted decoding. We assume two blocks  $x_i$  and  $x_{i+1}$ , sparse in a chosen basis, and rank their components in descending order by magnitude. Under unweighted decoding, the number of measurements required to recover a specific component is approximately

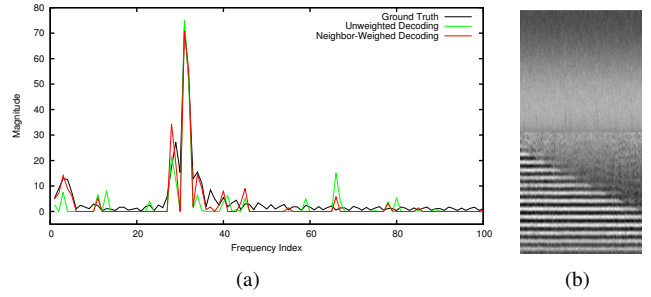


Fig. 5. (a) Recovered component magnitudes for the signal shown in Figure 4(c)/(d) with and without neighbor-weighted decoding for  $M = 25$ . Our strategy results in a significant improvement in the recovery of low frequencies, which are dropped in unweighted decoding under insufficient  $M$ . (b) The result of using neighbor weighted decoding, and increasing the number of measurements to  $200 \times 35$ , the amount required to achieve an error of  $P=0.45$ . (See Section IV-A for a discussion of this error metric).

proportional to its rank. For simplicity, we assume exact proportionality, noting that this constant cancels in our final expression.

Suppose that, in order to be free of blocking artifacts, our target application requires that block  $x_{i+1}$  include components  $f_a$  and  $f_b$ . Let  $M_{i+1}$  be the number of measurements required under unweighted decoding to recover these two components:

$$(7) \quad M_{i+1} = \max(\text{rank}_{i+1}(f_a), \text{rank}_{i+1}(f_b))$$

Furthermore, we assume a scenario in which conditions  $C1$  and  $C2$  below hold:

$$C1 : \text{rank}_{i+1}(f_a) = \alpha \cdot \text{rank}_{i+1}(f_b), \text{ for } \alpha > 1.$$

$$C2 : f_a \text{ is among the } D_i \text{ dominant components of block } x_i$$

This yields two new expressions. By (7),

$$(8) \quad M_{i+1} = \alpha \cdot \text{rank}_{i+1}(f_b)$$

$$(9) \quad D_i \geq \text{rank}_i(f_a)$$

Suppose that we use neighbor-weighted decoding for block  $x_{i+1}$  where  $D_i$  dominant components are imported from block  $x_i$ . Then by the theorem in Section II A, the required number of measurements  $M'_{i+1}$  under neighbor-weighted decoding is:

$$(10) \quad \begin{aligned} M'_{i+1} &= D_i + \text{rank}_{i+1}(f_b) \\ &\geq \text{rank}_i(f_a) + \text{rank}_{i+1}(f_b) \end{aligned}$$

We define the *gain of neighbor-weighted decoding* as the ratio  $M_{i+1}/M'_{i+1}$ . Then by (8) and (10),

$$(11) \quad \text{Gain} = \frac{\alpha \cdot \text{rank}_{i+1}(f_b)}{D_i + \text{rank}_{i+1}(f_b)}$$

Suppose that the input signal for block  $x_i$  is exactly sparse with sparsity  $k_i$ . Then we have  $k_i \geq D_i$ . Thus, by (11):

$$(12) \quad \text{Gain} \geq \frac{\alpha \cdot \text{rank}_{i+1}(f_b)}{k_i + \text{rank}_{i+1}(f_b)}$$

Furthermore, if  $\text{rank}_{i+1}(f_b) \geq k_i$ , then by (12)

$$(13) \quad \text{Gain} \geq \frac{\alpha}{2}$$

From (7) we see that if block  $x_{i+1}$  has a large  $\alpha$  and  $x_i$  is exactly sparse with a small  $k_i$ , then the gain is at least  $\alpha/2$ .

### B. Implementation Details

In order to implement neighbor-weighted decoding, we must address several implementation details. The first is dominant-component detection. When deriving weights, we should use only those dominant components we trust to be accurate, since erroneous weights may degrade the quality of the solution. In our implementation, we make the conservative assumption that, given an approximately sparse signal and  $M$  measurements, the  $M/8$  largest component values will be identified with high accuracy during the first pass decoding. This is based on the observation that, when using Bernoulli random matrices, exactly sparse signals with  $M/4$  non-zero components will be recovered exactly. We found this method to perform more consistently over a variety of signals than “knee detection algorithms” that operate on the sorted component magnitudes.

The second issue is deciding in which direction to export/import weights. As seen in Figure 4(a)/(c), importing weights from the bottom partition to the top will not help, while importing in the opposite direction will provide significant benefit. We export weights from the block whose components have largest magnitude. This is based on the observation the largest magnitude components are decoded most accurately, and absent any natural abrupt transitions, are most likely to contribute to surrounding blocks.

The third issue is what weight values to use. We found decoding was not particularly sensitive to weight values, except under very small numbers of measurements. In all cases, we observed better performance when imported components were weighted higher than locally-weighted components. This is because, by virtue of being imported, their values are much smaller than those of the dominant set.

## IV. EVALUATION

### A. Error Metric

We are interested in reducing blocking artifacts that present along a swath of the signal at the border between partitions. Our metric should capture the decoded accuracy with respect to ground truth values over this swath of interest,  $x_{SW}$ . To this end, we define  $x_{SW}$  to be a swath of fixed size, apply a DCT transform to  $x_{SW}$ , and compare the result to the corresponding transform of the reconstructed swath  $\hat{x}_{SW}$ . We define our *error metric*  $P$  as the proportion of dominant components decoded to within 10% accuracy in this swath.

The choice of width for  $x_{SW}$  influences the range of values that  $P$  takes on with respect to a decoded solution. At the extremes, using a swath that encompasses two entire neighboring blocks yields a comparison of decoding quality over the entire signal. Using a very small swath centered on

the border between two partitions judges how closely the edges of two blocks agree.

### B. Performance in Target Scenario

In order to measure our success in the scenario described in Section II-A, we apply unweighted, locally-weighted, and neighbor-weighted decoding to a  $200 \times 100$ -pixel test image. We center  $x_{SW}$  at the partition boundary, and set its width to 30 pixels. This test case is constructed to have three sets of frequencies: those dominant in block  $x_1$ , those dominant in block  $x_2$ , and a third set that contributes to both blocks, but is dominant in neither of them. Figure 6 illustrates this process, where diagonal matrix “masks” are applied to space-domain layers composed of each set of frequencies, and the results added. The resulting image is shown in Figure 7(a).

Figure 7(b) shows the value of  $P$  as  $M$  is increased for unweighted, locally-weighted, and neighbor-weighted decoding.  $M$  represents the total number of measurements required to decode the entire  $200 \times 100$  pixel image. When  $M$  is low, neighbor weighted decoding gives a consistent reduction in error. As  $M$  is increased to the point where  $P = 0.70$ , enough measurements are taken so that important components are found in the first pass unweighted decoding, and there is little differentiation between the two methods. At this point, both methods apply weights to the same components, and their results have the same degree of error.

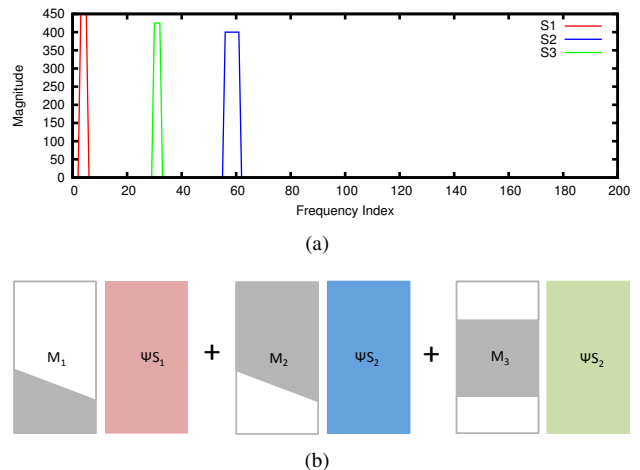


Fig. 6. (a) Magnitudes of the three sets of components used to generate the input image for evaluating the performance of neighbor-weighted decoding in the scenario from Section II-A. (b) An illustration of how these three sets of components were combined in order to generate the test case signal. By masking out the grey regions in the space domain for a given layer, the associated set of frequencies will not be present in the corresponding region of the image. As a result, the blue set of components becomes dominant in  $x_{i+1}$ , the red set in  $x_i$ , and the green set in neither, though it contributes equally to both blocks. The resulting image is shown in Figure 7(a)

### C. Perception

While our goal is to reduce artifacts due to decoding blocks independently, it is natural to try to interpret this as a reduction of visual artifacts in image-based applications. Though these are two different concepts, they agree in the

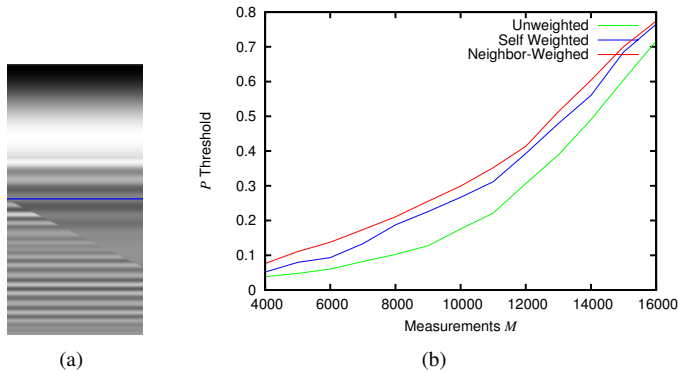


Fig. 7. (a) An input image used for the evaluation of neighbor-weighted decoding. The blue line denotes the partition boundary. (b) As  $M$  increases,  $P$  represents the proportion of dominant components recovered to within 10% accuracy in a 30 pixel-wide swath centered on the border between top and bottom partitions of the image. At an error threshold of  $P = 0.25$ , neighbor-weighted decoding has a compression ratio of 42%, while locally-weighted decoding achieves 48% and unweighted decoding 58%.

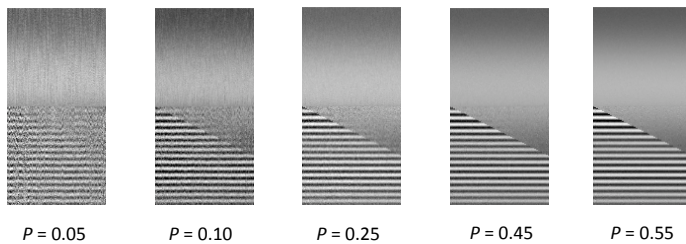


Fig. 8. For images, improving the decoding accuracy of dominant components in a border swath can result in a reduction in perceptual artifacts along partition boundaries; example images are shown with their accompanying  $P$  metric for a 20 pixel wide swath centered at the partition boundary.

sense that when compressive sensing is applied to signals that will be perceptually interpreted, decoding errors lead to perceptual errors. However, they differ in that signals may naturally contain abrupt transitions between blocks that strike the eye oddly, but are not a decoding error. Based on the analysis in Section II-A, this case is excluded by assuming that  $S_i \subseteq B_{i+1}$  and  $S_{i+1} \subseteq B_i$ . Also, given insufficient  $M$  for exact reconstruction,  $\hat{X}$  will contain visual artifacts not induced by partitioning that cannot be addressed by our method, and will exist regardless of partitioning.

To give context to the connection between blocking artifacts due to partitioning and the more general class of visual artifacts, Figure 8 shows several recovered versions of our demonstration image from Section II, and the corresponding values for error metric  $P$ . Note that our  $P$  is a high bar for perceptual accuracy; at values of  $P = 0.25$ , the transition between partitions becomes visually pleasing, and at  $P = 0.45$ , the transition is barely perceivable.

#### D. Performance on Natural Images

Figure 9 shows the result of applying neighbor-weighted decoding to a natural photograph. In this example, the top block has sparsity  $k_i = 8$ , and we can achieve exact recovery. The bottom block is not exactly sparse, and exact recovery

with few measurements is not possible;  $x_{i+1}$  has three dominant components, significant contributions from the top fifteen components, and a tail that extends to thirty components. We use  $M_{i+1} = 10$  to capture the dominant three components, and import the  $D_i = 8$  components from  $x_i$ , which ensures that the solver puts energy in several of the top fifteen components that were dropped under locally-weighted decoding. By expression (12) in Section III-A, we expect gain  $\geq \frac{5 \times 3}{8+3}$ ; this roughly corresponds to the improvement in compression ratio from 16% to 10% that we observe when we increase the number of measurements under locally-weighted decoding to achieve a similar approximation.

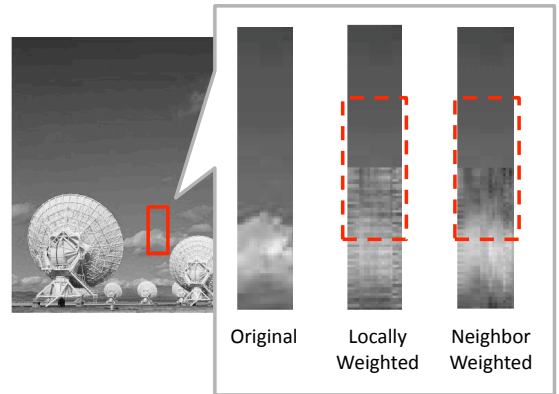


Fig. 9. Neighbor-weighted and locally-weighted decoding are applied to a  $600 \times 400$  pixel photograph. Inset, example recovered images using both methods. The top block is  $k_i = 8$  sparse and can be exactly recovered, while the bottom block has non-zero contributions from 30 components. Perceptually, the components of the upper block are responsible for differentiating the clouds from the sky in the lower block. When they are lost, that shape definition is lost and the clouds bleed into the swath. Restoring them with neighbor-weighted decoding reduces this effect, and results in a perceptually smooth transition between blocks. Therefore, we use  $M_{i+1} = 10$  measurements on the bottom block to recover its top 3 components, and then import components from the top block to yield an acceptable approximately recovered signal.

#### E. Meeting Error Thresholds

The discussion of perception and the results in Figure 8 lead to another practical implication when partitioned compressed sensing is applied: exact accuracy is not required, but instead we should try to meet an application-specific error threshold. Figure 10 shows the total number of measurements required to meet the range of error thresholds displayed in Figure 8 on our test image. Neighbor-weighted decoding saves an average of 2700 measurements over unweighted decoding at a threshold of  $P = 25\%$ , when most visual artifacts in the boundary swath are gone, and saves 1300 measurements over locally-weighted decoding, reductions of 24% and 13% respectively.

## V. RELATED WORK

The foundations for compressive sensing were laid in the seminal papers [17] and [18]. Since then, methods for improving decoding speed have led to alternatives to  $\mathcal{L}^1$  minimization, such as OMP, Total Variation, and CoSamp [11] [9] [10]. As with any signal processing method for large or infinite

$P$ Threshold	Unweighted	Local- Only	Neighbor- Weighting
0.05	5800	3900	2000
0.10	7900	6500	4900
0.25	11300	9900	8600
0.45	13800	12800	12100
0.55	14500	13900	13200

Fig. 10. Given a predefined threshold for error metric  $P$  based on application needs, neighbor-weighted decoding results in a significant savings over unweighted decoding or local weighting. Here we show a range of error thresholds corresponding to the visual examples in Figure 8. Neighbor-weighted decoding saves  $\approx 2000$  measurements when  $P$  is low, and  $\approx 800$  measurements for high thresholds of  $P$  over locally-weighted decoding.

length signals, partitioned encoding arises as another natural tack. In the compressive sensing field, partitioned encoding has been studied for block encoding of natural images [19], with basis-specific enhancements used to improve reconstruction quality. Authors have also recently proven sufficiency of block-diagonal matrices for signal recovery [20], and extended these results to analyze signals heterogeneous across partitions [21].

In traditional signal processing literature, reducing artifacts in discrete signal processing is often done by overlapping boundaries of adjacent blocks using Lapped Orthonormal Transforms such as the MDCT [22]. We are currently exploring the application of this technique to partitioned compressive sensing. Early experiments have produced visually appealing results, and we note that decoding using lapped transforms is a technique that fits well within the general framework of this paper, and benefits from neighbor-weighted decoding.

## VI. CONCLUSION

In this paper, we presented a method for partitioned compressive sensing that reduces the running time of the decoding step and makes decoding amenable to parallelization. Under this framework, exact recovery is only possible when a large number of additional measurements are taken, meaning that approximate signal recovery is a more reasonable goal. We showed that artifacts arising in approximate decodings due to partitioning can be mitigated by weighting dominant frequencies from neighboring blocks. This strategy is called neighbor-weighted decoding. The resulting reduction in error makes partitioned decoding possible with fewer additional measurements than an unweighted or traditional locally-weighted decoding.

Our method is a step towards addressing the system issues that arise when applying compressive sensing in practical scenarios with large numbers of unknowns. As such, useful next steps would include validating our implementation assumptions against the signal characteristics of real-world datasets geared toward proposed applications. Partitioned compressive sensing is an obvious extension of current theory, however its utility will vary depending on a given scenario, presenting an exciting avenue of future research.

## ACKNOWLEDGMENTS

This material is based on research sponsored by Air Force Research Laboratory under agreement numbers FA8750-10-2-0115 and FA8750-10-2-0180. The U.S. Government is authorized to reproduce and distribute reprints for

Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory or the U.S. Government. The authors would like to thank the Office of the Secretary of Defense (OSD/ASD(R&E)/RD/IS&CS) for their guidance and support of this research.

## REFERENCES

- [1] E. Candès, "Compressive sampling," in *Proceedings of the International Congress of Mathematicians*, vol. 3. Citeseer, 2006, pp. 1433–1452.
- [2] C. Luo, F. Wu, J. Sun, and C. Chen, "Compressive data gathering for large-scale wireless sensor networks," in *Proceedings of the 15th annual international conference on Mobile computing and networking*. ACM, 2009, pp. 145–156.
- [3] S. Katti, S. Gollakota, and D. Katabi, "Embracing wireless interference: Analog network coding," in *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2007, pp. 397–408.
- [4] M. Lustig, D. Donoho, and J. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.
- [5] H. Kung, C.-K. Lin, and D. Vlah, "Cloudsense: Continuous fine-grain cloud monitoring with compressive sensing," in *3rd USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 2011)*, 2011.
- [6] Y. Gwon, H. Kung, and D. Vlah, "Distroy: Detecting integrated circuit trojans with compressive measurements," in *6th USENIX Workshop on Hot Topics in Security (HotSec)*, 2011.
- [7] G. Strang, *Linear Algebra and Its Applications*, 4th ed. Brooks/Cole, 2006.
- [8] R. Berinde and P. Indyk, "Sparse recovery using sparse matrices," *Computer Science and Artificial Intelligence Laboratory Technical Report*.
- [9] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *Information Theory, IEEE Transactions on*, vol. 53, no. 12, pp. 4655–4666, 2007.
- [10] D. Needell and J. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.
- [11] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1–4, pp. 259–268, 1992.
- [12] H.-C. Chen, H. Kung, D. Vlah, and B. Suter, "Measurement combining and progressive reconstruction in compressive sensing," in *MILCOM*, 2011.
- [13] E. J. Cands, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus de l'Academie des sciences, Paris*, 2008, Series I, vol. 346, pp. 589, 2008.
- [14] E. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *Information Theory, IEEE Transactions on*, vol. 52, no. 12, pp. 5406–5425, 2006.
- [15] E. Candès, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted  $l_1$  minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, 2008.
- [16] H.-C. Chen, H. Kung, D. Vlah, D. Hague, M. Muccio, and B. Poland, "Collaborative compressive spectrum sensing in a uav environment," in *MILCOM*, 2011.
- [17] E. J. Candès, J. K. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, 2006.
- [18] E. J. Candès, J. K., Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.
- [19] L. Gan, "Block compressed sensing of natural images," in *Digital Signal Processing, 2007 15th International Conference on*. IEEE, 2007, pp. 403–406.
- [20] H. Yap, A. Eftekhari, M. Wakin, and C. Rozell, "The restricted isometry property for block diagonal matrices," in *Proceedings of the Conference on Information Sciences and Systems (CISS)*, 2011.
- [21] J. Park, H. Yap, C. Rozell, and M. Wakin, "Concentration of measure for block diagonal matrices with applications to compressive sensing," in *IEEE Transactions on Signal Processing*, 2011.
- [22] H. Malvar, *Signal processing with lapped transforms*. Artech House, 1992, vol. 220.