Measurement Combining and Progressive Reconstruction in Compressive Sensing

Hsieh-Chung Chen*, H. T. Kung*, Dario Vlah*, Bruce Suter†,

* *Harvard University, Cambridge, MA*
† *Air Force Research Laboratory, Rome, NY*

***Abstract*—Compressive sensing has emerged as an important new technique in signal acquisition due to the surprising property that a sparse signal can be captured from measurements obtained at a sub-Nyquist rate. The decoding cost of compressive sensing, however, grows superlinearly with the problem size. In distributed sensor systems, the aggregate amount of compressive measurements encoded by the sensors can be substantial, and the decode cost for all the variables involved can be large. In this paper we propose a method to *combine* measurements from distributed sensors. With our method we can transport and store a single combined measurement set, rather than multiple sets for all sensors. We show that via source separation and joint decoding, it is possible to recover an approximate to the original signal from combined measurements using *progressive* reconstruction which focuses on individual sensors. This results in a reduction in the number of variables used in decoding and consequently a reduced decoding time. We show that the computed approximation to the signal can still have sufficient accuracy for target detection. We describe the combining approach and the associated progressive reconstruction, and we illustrate them with image recovery for simple target detection examples.**

I. Introduction

A compressive sensing encoding system can be described as follows [1]:

$$y = \Phi x$$

where x is an N-dimensional signal being sampled, Φ is an $M \times N$ measurement matrix containing random entries, and y is a vector of M measurements, which are random linear combinations of components of x. Suppose that x is K-sparse in the sense that it can be expressed as a linear combination of K basis vectors in some basis, that is, $x = \Psi s$ where Ψ is an orthonormal transform, and s is a vector with no more than K nonzero coefficients. Then, if $M > cK \log(N/K)$ for some small constant c, it is possible to decode s with high probability, and retrieve $x = \Psi s$. Note that the number of measurements (M) is commensurate with the sparsity of a signal. If K is very small in some properly chosen domain, then M can be much smaller than N. This implies high compression rates. A rich volume of literature examines this topic including the landmark work of Candès, Tao [2] and Donoho [1].

In this paper, we address challenges concerning the use of compressive sensing in distributed sensor systems. In such a system, each sensor may gather compressive measurements for a specific region of a partitioned domain. That is, these distributed sensors perform partitioned encoding–one set of compressive measurements per partition–rather than standard encoding, as depicted in the middle and left sections of Fig. 1, respectively. However, the aggregate size of all the measurements is proportional to the number of sensors and can result in large costs of transporting and storing these measurements.

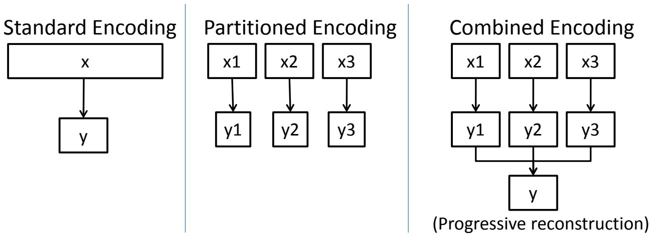


Fig. 1. (Left) Standard encoding, where the given signal x covers the entire domain. (Middle) Partitioned encoding, where partitions of x cover various regions of the domain. (Right) Combined encoding merges the measurements from all partitions to create a single measurement set.

To reduce the number of measurements, we consider in this paper a combining approach, as depicted by the Combined Encoding scheme of Fig. 1. We show that we can combine multiple measurement sets from different sensors, by simply adding together the corresponding measurements to form a single measurement set. As a result, we can transport and store a single measurement set, rather than measurement sets of multiple subproblems as depicted in Fig. 2.

In measurement combining, we will use a single measurement set to recover multiple sets of variables–one for each sensor in the distributed system. Since there are multiple sensors, this increases the problem size multiple times, and leads to another issue: high decoding cost. We note that the computational cost of the ℓ_1-minimization step in decoding grows superlinearly with the problem size. If the signal has N components, then ℓ_1-minimization by linear programming [2] can yield a decoding time of at least $\mathcal{O}(N^3)$. This means that the decoding cost can be prohibitively expensive when N is large. Other decoding methods such as matching pursuit [3], [4] are less expensive, but they only guarantee weaker error bounds in the recovered solution, and provide less stable recovery. When these low-complexity methods fail, we still need to resort to ℓ_1-minimization. It is therefore desirable to divide the problem into subproblems, each focusing on just one region at a time to reduce decoding time.

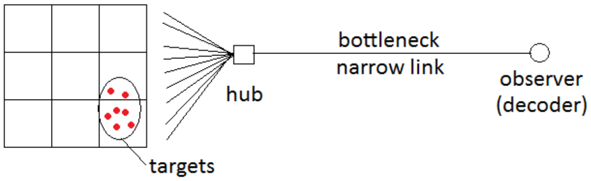


Fig. 2. A partitioned scenario of tracking targets in an area divided into multiple regions. In the example shown, there are 9 regions. For each region, we assume that there is one sensor encoding target information in that region, and sending compressed measurements to a hub to be combined. The hub combines the compressed measurements and transmits them to the decoder over a low-bandwidth link which benefits from measurement compression/combining.

We show that we can reconstruct an approximation to the original signal from the combined measurements by progressively decoding the subset of variables associated with just a single sensor at a time. This can significantly reduce decoding time in distributed sensor systems, as we will see in Section III. Further, we demonstrate in Section IV that the reconstructed signal resulting from progressive reconstruction can still have sufficient accuracy for target detection tasks.

Measurement combining and progressive reconstruction together can therefore reduce both the number of measurements and decoding time in distributed sensor systems, while yielding enough decoding accuracy for target detection. The approach is applicable to various application scenarios. For instance, as illustrated in Section II, when an area is monitored by multiple sensors, each sensor may monitor its own nearby region for targets. Then we can use the method of this paper to combine the measurements from such sensors, decode the combined measurements progressively, and finally detect targets from the reconstructed signal.

The measurement combining and progressive reconstruction ideas are based on some of our earlier theoretical work [5]. There we showed that via source separation and joint decoding, it is possible to separate out distinguished signal components in subproblems from combined measurements. The current paper focuses on applications of these ideas in distributed sensor systems, and illustrates the combining approach with some simple image recovery examples for target detection.

II. Design Space on Decoding for Partitioned Compressive Sensing

In this section, we describe the measurement combining method in detail using an illustrative scenario. Suppose we want to detect K targets from compressed measurements taken over α regions, as depicted in Fig. 2. Each sensor obtains a source signal vector x_i of length N from its own region, and then applies an $M \times N$ measurement matrix Φ_i where $M \ll N$, to obtain a compressed measurement vector $y_i = \Phi_i x_i$ of length M. The sensor then sends y_i to a hub node where the measurements may be combined.

Let us next describe a design space of methods that our system could use to reconstruct the signals of all regions. The first two are conventional methods that will serve as baselines for comparison, while the last one is the focus of this paper.

- **Conventional Decoding.** Suppose measurement matrices Φ_i are all chosen independently at random. Furthermore, suppose the hub computes *a sum*, y, of the measurements y_i before sending them out; then, we can write the sum as follows:

$$\sum y_i = \begin{bmatrix} \Phi_1 & \Phi_2 & \ldots & \Phi_\alpha \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_\alpha \end{bmatrix} \tag{1}$$

We can let $y = \sum y_i$, use x to denote the column vector on the right-hand side of Eq. (1), and let Φ be an $M \times \alpha N$ matrix which is the column concatenation of $M \times N$ matrices $\Phi_1 \ldots \Phi_\alpha$. Then we have

$$y = \Phi x$$

This is in the conventional compressive sensing form. Thus, we can decode x using the standard formulation

$$y = \Phi \Psi s \tag{2}$$

with a transform Ψ operating on an αN-size vector of coefficients s. While this formulation requires a low number of measurements, $cK \log(\alpha N/K)$, it has a high decoding time. For example, if ℓ_1-minimization is used, there can be an α^3-fold increase in the decoding time for each subproblem, $y_i = \Phi_i x_i$, due to an α-fold increase of the problem size from N to αN. The increase of variables is an issue we will address in this paper.

- **Partitioned Decoding.** A simple way to avoid solving an αN-size problem instance is for the hub *not* to sum up the measurements y_i, but instead forward them directly to the decoder. Thus, the decoder ends up with α size-N compressive sensing problems

$$y_i = \Phi_i x_i$$

which can be decoded much faster using the form

$$y_i = \Phi_i \Psi_i s_i$$

where the transform Ψ_i now operates on size-N vectors. However, this formulation requires that the hub transmit a larger amount of data; specifically, the total number of measurements to be transmitted is $\alpha cK \log(N/K)$. (Note that if fewer than $cK \log(N/K)$ measurements are used per region, then the signal may not be decodable if all K targets happen to be in the same region.)

- **Progressive Reconstruction.** This is the decoding method we propose in this paper. As in conventional decoding, the hub will again compute the sum of measurements $y = \sum y_i$ as in Eq. (1). How can we decode it faster than solving a size-αN problem? To answer this, let us consider the following decoding setup:

$$y = \begin{bmatrix} \Phi_1 & \dots & \Phi_\alpha \end{bmatrix} \begin{bmatrix} \Psi_1 & & \\ & \dots & \\ & & \Psi_\alpha \end{bmatrix} \begin{bmatrix} s_1 \\ \dots \\ s_\alpha \end{bmatrix}$$

This is, again, a size-αN problem with a high decoding cost; however, note that each Ψ_i here is α-times smaller than the one used in Eq. (2). We decode s_i separately using the constraint

$$y = \Phi_i \Psi_i s_i$$

Note that this constraint is approximate since it ignores nonzero contributions of $s_{j \neq i}$ to y [5]; nevertheless, in certain applications such as our target detection scenarios with sparse clusters, it is expected that these ignored contributions will be relatively small. Furthermore, as soon as larger s_i's are decoded for a region, we can use them to represent the entire region. This reduces the total number of variables in the subsequent decoding of other regions. The quality of each decoding step improves as more regions are decoded and represented with a reduced number of variables. Thus we call the method *progressive reconstruction*.

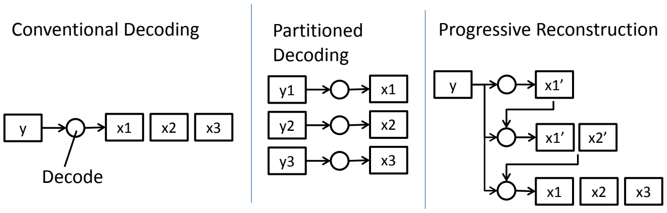


Fig. 3. (Left) Reconstructing all regions of the combined measurement at once will incur a high decoding cost, due to a large number of variables. (Middle) Decoding each partition separately will result in a lower decoding cost, but at the expense of lower compression rate. (Right) Progressive reconstruction focuses on a single region at time while considering contributions from a few most significant variables in previously decoded regions. It has a low decoding cost, and requires substantially fewer measurements when compared to partitioned decoding.

Fig. 3 depicts the three approaches described in this Section. We note that conventional decoding involves high decoding cost. Partitioned decoding has a reduced cost, but at the expense of a poor compression rate. So, we propose the third solution—progressive reconstruction. Our method has relatively high compression rate and low decoding cost, while achieving a decoding accuracy sufficient for target detection. In the rest of the paper we describe progressive reconstruction in detail, and evaluate its performance.

III. Progressive Reconstruction on Combined Measurements

For simplicity, we describe our progressive reconstruction method for a partitioned problem involving only $\alpha = 3$ subproblems. Note that the description generalizes to any number of subproblems. Suppose that the signals for the three regions are: x_1, x_2, and x_3. As described in Section II, we encode them with the measurement matrices Φ_i to yield three measurement sets, $y_i = \Phi_i x_i$, for $i = 1$, 2 and 3. We combine the y_i's to form y:

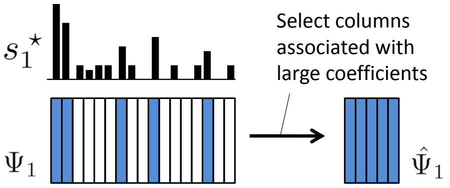


Fig. 4. Forming a reduced basis for region 1.

$$y = y_1 + y_2 + y_3 \tag{3}$$

In our scenario we assume that the targets are *clustered*, meaning that the large variables are concentrated in only a few of the regions.

The first step of the reconstruction method is to identify the region that contains the largest number of targets. This region can be chosen according to prior knowledge; for example, we might be able to predict the targets' location based on past observations. In other applications where no such side information is available, the best region can still be determined by observing preliminary decoding results. A generic method works by decoding the combined measurements as follows:

$$s_i^\star = \arg\min |s_i|_{\ell_1} \text{ subject to } y = \Phi_i \Psi_i s_i \tag{4}$$

for each region i. In the best region, by definition the energy will be more concentrated on a few coefficients, allowing us to find a sparser solution. Given sufficient measurements, we will have enough constraints so that a sparse solution for other regions is unlikely. Therefore, we identify the region with the smallest $|s_i^\star|_{\ell_1}$ to be the best region.

Finding this region is important because y has contributions from all three x_i's, and each $s_i^\star$ may contain interference from other regions. Of these, the signal x_i corresponding to a region that has most targets will experience the least interference from other sources, and will have the best reconstruction result.

We can now begin the progressive decoding stage. Suppose we find that the targets are concentrated in region 1 from the result of the first step. We then identify the basis associated with the largest variables in the reconstructed $s_1^\star$, denoted as $\hat{s}_1$, and use it to represent x_1 in the subsequent steps. Specifically, we form $\hat{\Psi}_1$ by keeping only the columns associated with the larger coefficients, as shown in Fig. 4.

We continue the progressive decoding by reconstructing s_2 in the second round as follows:

$$s_2^\star = \arg\min \left\| \begin{bmatrix} \hat{s}_1 \\ s_2 \end{bmatrix} \right\|_{\ell_1} \text{ subj. } y = \begin{bmatrix} \Phi_1 \hat{\Psi}_1 & \Phi_2 \Psi_2 \end{bmatrix} \begin{bmatrix} \hat{s}_1 \\ s_2 \end{bmatrix}$$

Since $\hat{s}_1$ includes all the large variables in s_1, there is little or no interference from x_1 when we reconstruct s_2. Similar to the first round, we find the most important basis for s_2, and proceed to decode s_3 in the same fashion.

In each round, we slightly increase the number of variables to decode by including the basis vectors that are deemed to be important from all previous rounds. If no round misses important basis columns for the regions in question, then at the end of this process we will be solving an interference-free signal.

Compared to the progressive reconstruction process, conventional decoding which decodes signals from all regions in one shot exploits the sparsity better, as it represents the signal with a larger selection of basis vectors (a larger dictionary). However, in a real-world application like target tracking, we might have knowledge about the locations of the targets from a previous state. For example, if we know that most targets are in region i at a previous time, it is reasonable to assume that region i and its surrounding regions are more likely to contain the targets. In the progressive reconstruction method, we could take advantage of such knowledge by having early rounds focus on these regions.

IV. An Illustrative Application Example

We present a simple example in distributed sensor systems to demonstrate the idea of measurement combining and progressive reconstruction.

Suppose we want to track K targets across 9 regions, as depicted in Fig. 2. The sensors compress the sensed images x_i by computing $y_i = \Phi_i x_i$. A hub gathers the measurements y_i and performs a simple summing operation to get $y = y_1 + y_2 + ... + y_9$, and then transmits y back to us.

In this example, we detect one or more copies of the same bee-like target as shown in Fig. 5. To identify targets, we build a dictionary where each basis function represents the target at a distinct position in the region, as shown in Fig. 5. More precisely, in the dictionary matrix, each column is a vector representation of the input image with the target at a distinct location. We can view the input signal x as the sum of those images each containing a single instance of the target at a distinct location. Recovering x means finding these image components, or a vector s such that

$$x = Ds$$

where D is the dictionary matrix defined above. When there are just a few targets present in the signal, s is sparse and this problem will be solvable by compressive sensing decoding using the dictionary as the basis Ψ_i described in Section III. This is similar to finding parameterized shapes in images as described in [6]. For simplicity, we assume the background can be subtracted from the measurements.

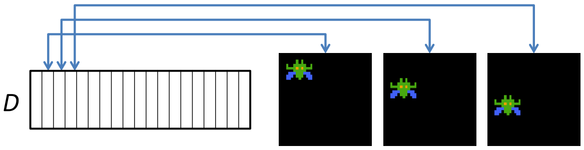


Fig. 5. Each column in the dictionary matrix represents the target at a different location. An image with k targets can be represented by k coefficients in this dictionary.

As described in Section III, we perform a reconstruction for each region under some noise model and determine where the targets are most concentrated. Then, by progressively decoding one of the remaining regions at a time, we gradually improve the reconstruction until all regions are decoded. The whole process takes at most $9 + 8 + 7.. + 1$ decoding steps, where there are roughly N variables to solve each step. Suppose that ℓ_1-minization is used in decoding. Then, the total cost is $45CN^3$, for some scaling constant C. In contrast, if we were to decode the entire image $(9N)$ at once, the cost would be $C(9N)^3 = 729CN^3$.

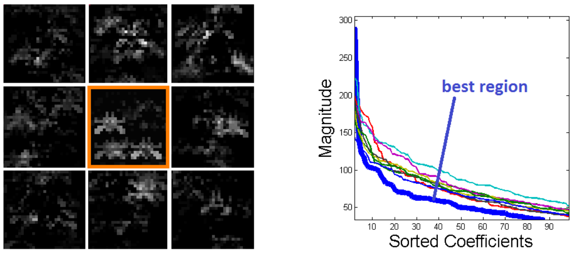
V. Experiments and Performance Results

We examine the performance of progressive reconstruction in four cases:

1) Four targets spread across two regions;
2) Five targets spread across three regions;
3) Five targets spread across three regions with Gaussian noise (SNR ≈ 13);
4) Three targets in a real-world satellite image.

The size of each region is 30×30 pixels. The first three cases each have 8100 pixels in total, whereas the last case has 5400 pixels. 60 measurements are used for the first two cases (compression ratio $< 1\%$), 405 measurements are used for the third case (compression ratio $= 5\%$), and 1200 measurements are used for the last case. We use ℓ_1-magic as the decoder for the first three cases, and CoSaMP [4] for the last example.

A. Case 1



(a) Reconstructed image

(b) Magnitude of reconstructed coefficients for 9 regions

Fig. 6. First decoding round for Case 1. Each region is decoded independently. The center region has the least interference from other regions, resulting in a better reconstruction. (a) The reconstructed images for each region. Only the center region displays pronounced targets. (b) The sorted magnitude of reconstructed coefficients for 9 regions. One can observe that the ℓ_1-norm of the best region is smaller.

To begin progressive reconstruction, we must first find a region to start with. In the first round, we decode the nine regions independently. Fig. 6(a) shows the decoded results for each region. The center region has the smallest ℓ_1-norm under the dictionary basis, so we start the second round based on the reconstruction of the center region. In the second round, we reconstruct the signal in two regions together using a reduced basis for the center region as described in Section III. We continue to recover other regions until all signals are decoded. Fig. 7 shows three snapshots of the decoding process. The

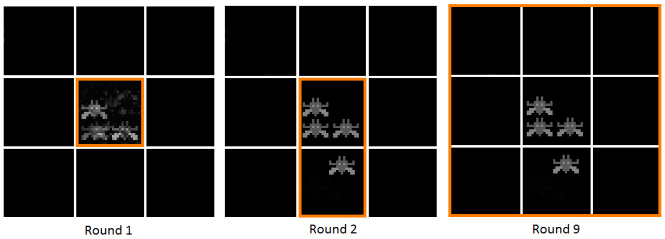


Fig. 7. Progressive reconstruction for Case 1. The first decoding round reveals that the targets are most concentrated in the middle region. The second round of decoding uses the information from the first round to improve the reconstruction. Since there are no targets left in other regions, the reconstruction in the second round is exact.

second round gives an exact reconstruction because there is no interference from other regions.

B. Case 2

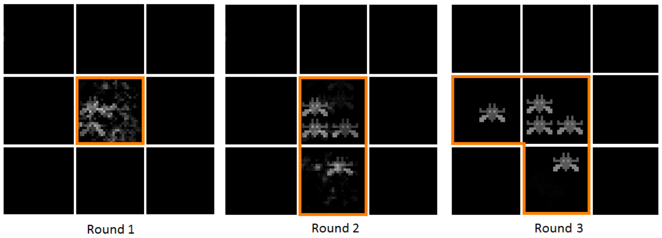


Fig. 8. Progressive reconstruction for Case 2. As in Case 1, the first decoding round reveals that the targets are most concentrated in the center region. The second round of decoding uses the information from the first round to improve the reconstruction. Since there is interference from a target that has not been detected yet, one can see that the reconstruction in the second round is still noisy. The reconstruction is improved again in the third round, and in this case it is exact.

Here we use the same decoding method as Case 1, but now there are three regions that contain targets. Fig. 8 shows the first three rounds of the decoding process. The reconstruction in round 1 is worse than the previous case, beacuse now there is interference from two other targets instead of one. However, similar to Case 1, the interference is progressively reduced until it falls to zero, and the reconstruction becomes exact after the third round.

C. Case 3

In this case the signal is corrupted by noise. Fig. 9 shows the results from each round of the progressive reconstruction. The recovery in the first round is heavily distorted, but the three targets in the center region are still distinguishable. We only keep the bases associated with the largest 2% coefficients (refer to Fig. 4). Note that distortion is most significant in the newest regions being solved; this is because we used reduced bases to represent the signals for regions that have already been processed, whereas the latest region added to the decoding is represented with a full dictionary.

The results of progressive reconstruction and standard decoding are compared in Fig. 10. Both approaches locate the targets successfully, but the error from noise is distributed in a different way. While providing comparable results, the progressive reconstruction process is 10 times faster, assuming $\mathcal{O}(N^3)$ decoding time.

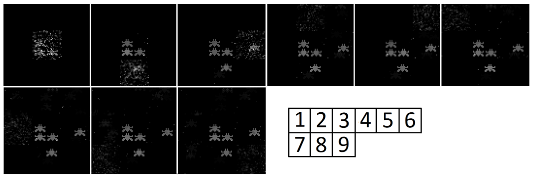


Fig. 9. Progressive reconstruction for Case 3, a noisy image. Decoding progress of all 9 rounds is shown. We focus on one region each round.

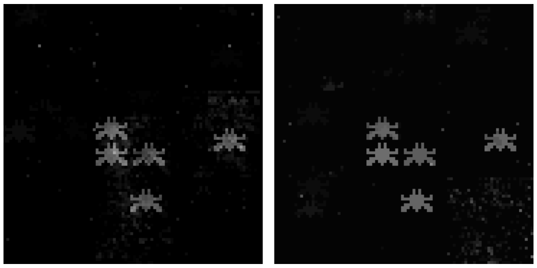


Fig. 10. Comparison between progressive decoding (right) and decoding all regions (left) at once.

D. Case 4

We consider a satellite image of three boats. Fig. 11(a) is the original image. For this example we use a down-sampled gray level version of the original image. Color gradients are applied to visually enhance the contrast in the intensity images. The ships are the targets that we want to identify in the reconstructed image. The background can be represented by a relatively small basis because its characteristics are known. In our experiment, we took the first 100 columns of a DCT matrix as the basis for the background.

We use basis S_i to represent the ships, and basis B_i to represent the background in region i. The decoding sequence is predetermined based on some side information so that we

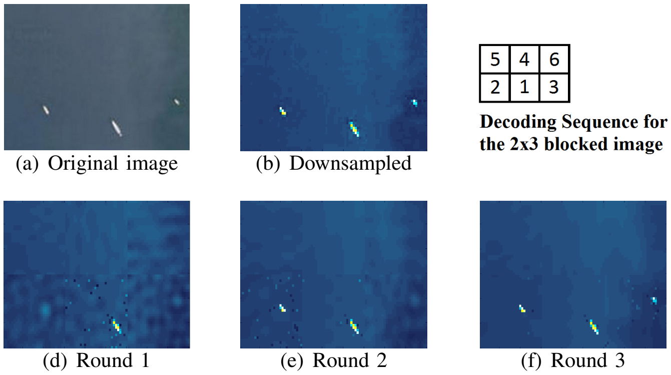

(a) Original image
(b) Downsampled
(d) Round 1
(e) Round 2
(f) Round 3

Fig. 11. (a) The original image. (b) Down-sampled image with gradient color. (d), (e) and (f) are the reconstructions after round 1,2 and 3. The decoding sequence is determined by side information (starting from the region with highest probability of containing a target). Reconstruction in the third round finds all targets and the background.

can decode the region with most targets first. For the first round, we decode with the basis $\Psi_1 = \begin{bmatrix} B & S_1 \end{bmatrix}$ where $B = \begin{bmatrix} B_1 & B_2 & ... \end{bmatrix}$.

Then, we use the reduced basis $\Psi_2 = \begin{bmatrix} \hat{\Psi}_1 & S_2 \end{bmatrix}$ for round 2, where $\hat{\Psi_1}$ contains only the columns corresponding to larger variables. Similarly, for round i , $i > 1$, we use $\Psi_i = \begin{bmatrix} \hat{\Psi}_{i-1} & S_i \end{bmatrix}$ as the decoding basis.

In the first round (Fig. 11(d)), there is some high frequency noise due to the error caused by two missing targets. The missing targets cause false positive detection of large coefficients in the frequency domain, so when we form the reduced basis for the second round, some unimportant columns still remain. However, the new basis for the second round will now include S_2, which will change the outcome and give us another chance to get rid of the unimportant columns. The amount of noise decreases in the second round as there is only one target causing distortion. We can see that the target represented by S_2 is detected (lower left), and the background is smoother because most of the unimportant frequency coefficients have been discarded correctly. In the third round, all targets are revealed and represented by the suitable basis, and the background becomes very similar to the original image.

VI. Discussion

In the experiments, the reconstruction process can take advantage of additional side information. If we know the targets are clustered in a small set of regions, then our method can first identify the most populated region, and improve the reconstruction progressively at a low cost. Furthermore, if we know where these targets are most likely to be, this information can help us find a better reconstruction sequence and thus save more time. This can be a reasonable assumption for some surveillance systems where the targets are constantly monitored. When a number of measurements used is relatively small, there could be blocking artifacts resulting from the progressive reconstruction. In this case we can further apply methods such as those described in [7] to ensure that the recovered signals for neighboring partitions will be compatible. Compressive sensing for distributed sensor systems is also studied in the context of collaborative decoding at a central node, such as a UAV, where the decoded results for one sensor can enhance the decoding for other sensors [8].

There are other approaches that also directly benefit from side information. Reweighted approaches [9], [10] can use side information to guide the decoding process to a more desirable solution. However, these methods only improve the reconstruction quality but not the decoding time.

VII. Conclusion

In this paper, we take a perspective that an approximate signal reconstructed from compressive measurements just needs to be accurate enough for the target detection problem in hand. To this end, we presented a method of combining measurements in compressive sensing in order to lower the number of measurements, or equivalently to improve the compression ratio. In addition, we presented a progressive reconstruction method that can decode multiple signals from combined measurements with reduced decoding cost. We validate these results with target detection test cases involving both synthetic and real-world images. Our approach can reduce both measurement and decoding costs, while producing decoding results with sufficient accuracy for target detection. Results of this paper are applicable to target-detection applications based on compressive sensing for which problem partitioning is desired and distributed sensing systems where compressive measurements from multiple sensors can be integrated.

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] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, pp. 1289–1306, 2006. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.95.3320

[2] E. J. Cands and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[3] D. Needell and R. Vershynin, "Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 310–316, Apr. 2010. [Online]. Available: http://dx.doi.org/10.1109/JSTSP.2010.2042412

[4] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, May 2009. [Online]. Available: http://dx.doi.org/10.1016/j.acha.2008.07.002

[5] H.-C. Chen and H. T. Kung, "Separation-based joint decoding in compressive sensing," in *IEEE International Conference on Computer Communication Networks (ICCCN 2011)*, 2011.

[6] A. C. Gurbuz, J. H. McClellan, J. K. Romberg, and W. R. Scott, "Compressive sensing of parameterized shapes in images," in *ICASSP*. IEEE, 2008, pp. 1949–1952.

[7] H. T. Kung and S. Tarsa, "Partitioned Compressive Sensing with Neighbor-Weighted Decoding," in *MILCOM 2011*.

[8] 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*.

[9] R. Chartrand and W. Tin, "Iteratively reweighted algorithms for compressive sensing," *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008.*, pp. 3869–3872, 2008. [Online]. Available: http://dx.doi.org/10.1109/ICASSP.2008.4518498

[10] 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, Dec. 2008. [Online]. Available: http://dx.doi.org/10.1007/s00041-008-9045-x