

A Tree Based Recovery Algorithm for Block Sparse Signals

Wenbin Guo, Xing Wang, Yang Lu and Wenbo Wang

Wireless Signal Processing and Network Lab

Key Laboratory of Universal Wireless Communication, Ministry of Education

Beijing University of Posts & Telecommunications (BUPT), Beijing 100876, P. R. China

Email: gw@bupt.edu.cn, wangxing613@126.com, luyangnnu@126.com, wbwang@bupt.edu.cn

Abstract—The structure of block sparsity in multi-band signals is prevalent. Performance of recovery algorithms that taking advantage of the block sparsity structure is promising in the compressed sensing framework. In this paper, we propose a binary tree based recovery algorithm for block-sparse signals, where we exploit the fact that each block may have zero and nonzero elements both. The proposed algorithm improves the current algorithms through iteratively separating the recovered blocks of signals into two smaller blocks. Therefore, greedy searching based algorithm is possible to obtain more accurate basis for signal recovery. Simulations are performed and the results show the improvements over current block-based recovery algorithms.

I. INTRODUCTION

Compressed sensing is an efficient signal acquisition technique that enables signals to be sampled below the Nyquist rate given that the signal can be sparsely represented in an orthonormal basis. According to the compressed sensing framework[1], [2], the signal recovery problem is equivalent to solving an underdetermined linear equations by taking the sparse structure of the signals into consideration. In summary, the recovery of an unknown signal is converted to the recovery problem of a $N \times 1$ vector \mathbf{x} from measurements \mathbf{y} , where $\mathbf{y} = \mathbf{A}\mathbf{x}$ is a vector of $M \times 1$ with $M < N$ and \mathbf{A} is a matrix with M rows and N columns. Throughout the paper, the vector \mathbf{x} is also referred as a signal that has been sampled.

It is well known that the solution of an underdetermined linear equations is not unique. However, a unique solution is possible with sparsity constraints of the vector given enough measurements. Here, the sparsity constraints of a vector mean that there are only k unknown positioned nonzero elements with $k \ll N$. It has been shown that the unknown vector can be uniquely recovered with only $M \propto O(k \log N/k)$ measurements if \mathbf{A} is properly chosen[3]. This result has given rise to a number of various recovery algorithms which have been proven to recover a sparse vector \mathbf{x} under a variety of different conditions on \mathbf{A} [3]-[6]. There are two kinds of widely studied algorithms, Basis Pursuit (BP) or ℓ_1 -minimization approach [3], and greedy searching based algorithms, such as Orthogonal Matching Pursuit (OMP)[7], [8] and CoSaMP[9]. In these algorithms, the nonzero elements are assumed to be randomly positioned in a sparse vector. However, the signals may have block-sparsity structure with

clustered nonzero elements in dealing with multi-band signals or in measurements of gene expression levels[10][11][12].

In this paper, we consider efficient methods for the recovery of sparse signals that exhibit block sparse structure. The related works have been shown in [13]-[17]. In [14], [15], a ℓ_2/ℓ_1 -norm algorithm has been proposed to recover the block-sparse signals. It has been shown in [15] that such mixed norm algorithm is guaranteed to recover any block-sparse signal if matrix \mathbf{A} satisfies the condition of small block-restricted isometry constant. In [13], an extension of the CoSaMP algorithm [9] with iterative hard thresholding [16] is proposed and shown to exhibit provably recovery guarantees and robustness properties to the block-sparsity signals. In [17], a block orthonormal matching pursuit algorithm (BOMP) is proposed by extending the OMP algorithm to solve the block sparsity problem and the performance dominates other algorithms.

However, in [13], [17], it's assumed that the block is idealized where the elements of a block are all zeros or non-zeros. Actually, in many practical system, the block-sparsity exhibits imperfection. For example, multi-band signals with different bandwidth. In other words, a block may contain both zero and nonzero elements. Although such imperfection is mentioned in [14], [15], the sparse vector \mathbf{x} is divided into blocks of different sizes and the proposed recovery method is based on the hypothesis that the location and the number of nonzero element have been known as a priori information.

In this paper, we consider the recovery method for the generalized block-sparsity signals such that the blocks may have both zero and nonzero elements. We extend the BOMP algorithm proposed in [17] through iteratively separating the detected non-zero block into two smaller blocks and take advantage of the information for the next recovery process. As such, a multi-step recovery algorithm is proposed for the generalized block-sparsity signals without knowing the block structure. Simulation results under noisy measurements show a significant better performance than the current proposed algorithms.

The remainder of the paper is as follows. In section II, necessary background and current algorithms for the block-sparsity signals are reviewed. The proposed algorithm is depicted in Section III. Simulation results and performance

comparison are shown in section IV. In section V, we conclude the results.

II. BLOCK-SPARSE SIGNAL RECOVERY

We consider the problem of recovering a vector \mathbf{x} from noisy measurements \mathbf{y} given by

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z}, \quad (1)$$

where $\mathbf{x} \in \mathbb{C}^{N \times 1}$ is the unknown signal, $\mathbf{A} \in \mathbb{C}^{M \times N}$ is the measurement matrix of size $M \times N$ with $M < N$, $\mathbf{z} \in \mathbb{C}^{M \times 1}$ are the noises added to the measurements $\mathbf{y} \in \mathbb{C}^{M \times 1}$ with i.i.d Gaussian distribution of zero mean and variance σ^2 for each entry. We assume that \mathbf{x} is k -sparse, where the number k of non-zero entries is far less than N . Let us define a block in \mathbf{x} is a cluster of continuous zero or non-zero elements, where the run length of the cluster is defined as the block size. Denoting the number of the non-zeros blocks is m and the total number of blocks is L . Thus, the block- m -sparse signal is defined as a sparse signal with $m \ll L$. However, such definition of block-sparse signal is not helpful in practice since the block size may be different, which leads to the difficulty in the recovery from the measurements. In this paper, we assume that the sparse vector \mathbf{x} is separated into blocks with equal block size and refer the block wise vector as the block-sparse signal. An example of a block-sparse signal is shown in Fig.1(b), where the sparse vector is separated into blocks with equal block size d . Throughout the paper, we refer such blocked signal as block-sparse signal with size d . As shown in the Fig.1, each block may have all-zero elements or a part of non-zero elements.

A block-sparse signal with d can be formed as

$$\mathbf{x}(d) = (\mathbf{x}[1], \mathbf{x}[2], \dots, \mathbf{x}[n])^T, \quad (2)$$

where $\mathbf{x}[p] = (x_{pd+1}, \dots, x_{pd+d})$ denotes the p th block with $1 \leq p \leq n$ and $N = nd$. Assume that N/d is an integer, the measure matrix \mathbf{A} is represented as the following block form

$$\mathbf{A} = (\mathbf{A}[1], \dots, \mathbf{A}[n]), \quad (3)$$

where $\mathbf{A}[p] = (\mathbf{a}_{pd}, \dots, \mathbf{a}_{pd+d})$ and \mathbf{a}_l is the l th column vector of \mathbf{A} . Note that the block-sparse signal degenerates to a conventional sparse signal with $d = 1$. Thus, the recovery method for the block wise signal is a more general problem.

A. The mixed ℓ_2/ℓ_1 -norm recovery algorithm

In [14], [15], a mixed ℓ_2/ℓ_1 -norm algorithm was proposed to recover the block-sparse signal. The sparse signal \mathbf{x} is recovered by solving the following convex second order cone programming problem

$$\begin{aligned} \min_{\mathbf{x}} \quad & \|\mathbf{x}\|_{2,I} \\ \text{s.t.} \quad & \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \leq \sigma^2, \end{aligned} \quad (4)$$

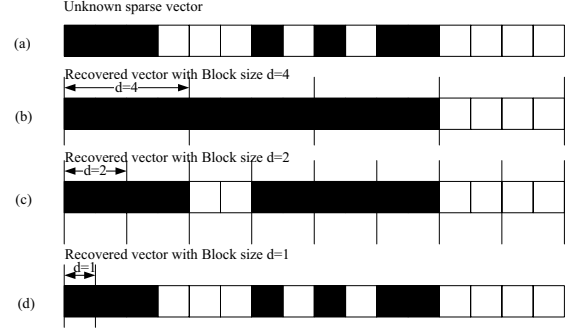


Fig. 1. A block-sparse vector recovered by BOMP with block size $d=4,2,1$. Here, the dark elements are non-zero and the white elements are zeros.

where the mixed norm of \mathbf{x} is defined as

$$\begin{aligned} \|\mathbf{x}\|_{2,I} &= \sum_{p=1}^n \|\mathbf{x}[p]\|_2 \\ \|\mathbf{x}[p]\|_2 &= \sqrt{\mathbf{x}[p]^H \mathbf{x}[p]}. \end{aligned}$$

The problem in Eq.(4) can be efficiently solved by the current optimization packages, such as CVX [18].

B. The BOMP algorithm

In [17], a block wise OMP algorithm (BOMP) is proposed by extending the OMP algorithm. In such greedy searching algorithm, the sparse vector is recovered by iteratively searching the optimal “matching basis”, which is a sub-block of matrix \mathbf{A} in BOMP algorithm or a column vector of \mathbf{A} in OMP algorithm. The BOMP algorithm is as follows.

Algorithm 1 Block-Orthogonal Matching Pursuit

Input: $\mathbf{y}, \mathbf{A}, d, \epsilon$

Initialization: $k \leftarrow 0, \mathbf{r}_0 \leftarrow \mathbf{y}, C \leftarrow \emptyset, \Delta \leftarrow \infty, \tilde{\mathbf{x}} \leftarrow \mathbf{0}$;

Repeat until $\Delta < \epsilon$

$j = \arg \max_{\{i=1,2,\dots,n\}} \|\mathbf{A}^H[i]\mathbf{r}_k\|_2$;

$k \leftarrow k + 1$;

$C \leftarrow C \cup \{j\}$;

$\tilde{\mathbf{x}}_k \leftarrow \min_{\tilde{\mathbf{x}}[i], i \in C} \left\| \mathbf{y} - \sum_{i \in C} \mathbf{A}[i]\tilde{\mathbf{x}}_{k-1}[i] \right\|_2$

$\mathbf{r}_k \leftarrow \mathbf{y} - \sum_{i \in C} \mathbf{A}^H[i]\tilde{\mathbf{x}}_k[i]$

$\Delta \leftarrow \|\mathbf{r}_k\|_2$

Output: $\tilde{\mathbf{x}}_k$.

In [17], it is shown that the BOMP algorithm is much better than the traditional OMP algorithm in the successful recovery rate. In the BOMP algorithm, a correct block is picked every step and the energy of the residual decays exponentially[17].

However, as illustrated by the Fig.1, the recovered vector is different with different block size d . From the above

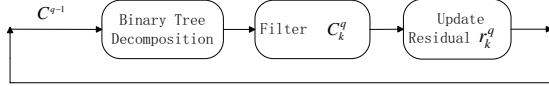


Fig. 2. A Conceptual Diagram of the Proposed BT-BOMP

mentioned BOMP algorithm, we observe that there are d basis (block size) of matrix \mathbf{A} been added at each step for the recovery of the sparse vector \mathbf{x} . The selected basis are used to determine the corresponding positions' values of the unknown vector $\tilde{\mathbf{x}}$ through $(\tilde{\mathbf{A}}^H \tilde{\mathbf{A}})^{-1} \tilde{\mathbf{A}}^H \mathbf{y}$, which is well known that is not sparse. In other words, the recovered values of the corresponding positions in each step of the BOMP algorithm are almost non-zeros. Therefore, the BOMP algorithm is suitable for the regular structure block-sparse signal, where the run length of the non-zero elements are same. However, in certain situations, such as multi-band signals with different bandwidth, there may have both non-zero and zero elements in a block, which will lead to inaccurate recovery of vector \mathbf{x} . Another issue is that the block size d is not easy to determined without a prior information of the signal. Thus, we propose a binary tree based BOMP algorithm (BT-BOMP) for the recovery of sparse vector \mathbf{x} without knowing its block structure.

III. THE BINARY TREE BASED BOMP ALGORITHM

The key innovation of the binary tree based BOMP algorithm (BT-BOMP) lies in that we start a BOMP algorithm with block size 2^m and separate each recovered non-zero block into two half blocks with block size 2^{m-1} . The BT-BOMP algorithm takes advantage of the recovered non-zero blocks and obtains a finer supporting set of basis by searching their half blocks. As such, a more accurate recovered vector can be obtained. The searching procedure continues until the block size with $d = 1$.

The conceptual diagram of the BT-BOMP algorithm of the q th step is shown in Fig.2. Comparing to the BOMP algorithm, the difference of BT-BOMP algorithm lies in the multi-step process and filtering the set of chosen indices for next step, which enables more robustness against the noise. The multi-step process can be expressed as a *binary sort tree*. As shown in the Fig.2, assume that the block is of size d in the $(q-1)$ th step, the block OMP algorithm is used to recover all the non-zero blocks with C^{q-1} being the set of these blocks. In the binary tree decomposition, each non-zero block in C^{q-1} is separated into two sub-blocks with half block size. In the filtering stage, the BT-BOMP algorithm searches the matching basis within the set of half blocks and filters the corresponding zero sub-blocks. The residual vector is updated after filtering stage.

The pseudo code of the BT-BOMP algorithm at q th step is given in the Algorithm 2, where the halting parameter ξ is the variance of the noise. Such process runs until the block size decreases to 1. The BT-BOMP algorithm starts from a

suitable block size $d = 2^m$, where the BOMP algorithm in section II is used as a warm-start. It's noteworthy that in the *Filtering* stage, the BT-BOMP algorithm selects the best matching blocks among the estimated set, while the BOMP algorithm only selects the best block among \mathbf{A} once.

Algorithm 2 Binary Tree based Block-Orthogonal Matching Pursuit

Input: priori knowledge $C^{q-1}, \mathbf{y}, \mathbf{A}$
Initialization: $k \leftarrow 0, \mathbf{r}_0^q \leftarrow \mathbf{y}, C^q \leftarrow \emptyset$
1: Binary tree decomposition:

$$\widehat{C}^q = \{2 * C^{q-1} - 1\} \cup \{2 * C^{q-1}\}$$
where \widehat{C}^q serves as an initial estimate.
Repeat
2: Filtering:

$$i_k = \arg \max_{\widehat{C}^q} \|\mathbf{A}^H [i] \mathbf{r}_{k-1}\|^2$$

3: Updating :
add the coordinate i_k to C^q : $C^q = \{i_k\} \cup C^q$
update the residual

$$\widehat{\mathbf{x}}^q = \arg \min_{\{\tilde{\mathbf{x}}_k[i]\}_{i \in C^q}} \left\| \mathbf{y} - \sum_{i \in C^q} \mathbf{A}[i] \tilde{\mathbf{x}}_k[i] \right\|^2;$$

$$\mathbf{r}_k^q = \mathbf{y} - \mathbf{A} \widehat{\mathbf{x}}^q$$

4: Halting:
if $\|\mathbf{r}_k^q\|_2 \leq \xi$ **then**
return $C^q, \widehat{\mathbf{x}}^q$;
else
quit the iteration, $q \leftarrow q + 1$;
end if
Until halting condition true;
Output: $C^q, \widehat{\mathbf{x}}^q$.

Since the BT-BOMP algorithm allows to start with a large block size d , the BOMP algorithm can find the true supporting block basis in matrix \mathbf{A} with high probability. Therefore, part of noises are suppressed at the filtering stage. As such, the BT-BOMP algorithm with multi-step process may obtain more accurate supporting basis and robustness against noise. The simulation results given in the next section testify our idea.

IV. SIMULATION RESULTS

In this section, we compare the simulation results of the proposed BT-BOMP algorithm with current algorithms for block-sparse signals. For all the simulations, we generate a random dictionaries \mathbf{A} by drawing i.i.d Gaussian variable for each element of \mathbf{A} . Such random matrix satisfies the block RIP conditions in [17]. The elements of the sparse vector \mathbf{x} are complex Gaussian variables with mean $\mu = 1$ and variance 0.1 for both real and imagine components.

A. Variant Noise Levels

In this experiment, the length N of the sparse vector \mathbf{x} is 960. The block-sparsity k is chosen to be 2, where only 2 blocks are non-zeros. All the algorithms are simulated with

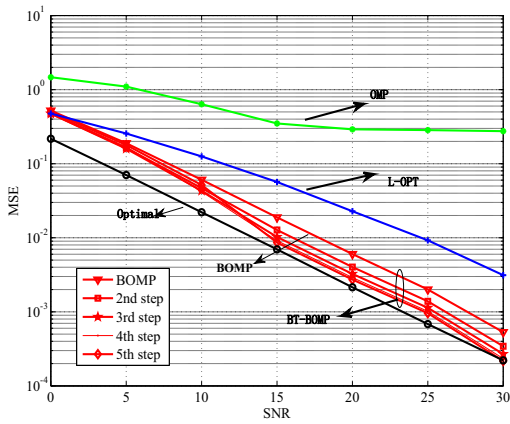


Fig. 3. Robustness of BOMP, BT-BOMP, L-OPT for a dictionary with $M=180$, $N=960$ and $d=16$ under variant noise levels

the same noisy measurements. In Fig.3, the mean square errors(MSE) of the recovery sparse vector are compared with various noise levels, where the MSE is defined as

$$MSE = \frac{\|\tilde{\mathbf{x}} - \mathbf{x}\|^2}{\|\mathbf{x}\|^2}.$$

It is seen that the MSE of the BT-BOMP algorithm is much less than other algorithms for the same noise level, where the L-OPT algorithm is for the mixed ℓ_2/ℓ_1 -norm algorithm solved by CVX optimization software[18]. In order to better understand the performance of the OMP-like algorithms, we give the optimal performance by assuming that the non-zero positions of the sparse vector \mathbf{x} are known and the corresponding basis are directly selected from the dictionaries \mathbf{A} for the recovery. It is seen from the Fig.1 that the MSE performance of the BT-BOMP algorithm in the first step is exactly the same as the BOMP algorithm since the BT-BOMP algorithm employs the BOMP algorithm as a warm-start. However, the BT-BOMP algorithm is significantly better than the BOMP algorithm with the multi-step process.

B. Various Block-sparsity Levels

We further compare the MSE performances under different block-sparsity between the proposed BT-BOMP algorithm and current algorithms. In this experiment, the SNR is all set to 20dB and $N = 960$. Fig.4 depicts the MSE performance with block-sparsity $k = 2, 3, 4$.

It is obvious that the BT-BOMP algorithm is the best. It's noteworthy that the performance of the BOMP and BT-BOMP algorithms is close as the block-sparsity being larger. It is because that we fixed the dimension of measurements in the simulations and the BOMP algorithm may not be able to select all of the correct non-zero blocks in the first step, which deteriorates the MSE performance. In order to overcome this, a larger block size can be selected as the first step in the BT-BOMP recovery procedure.

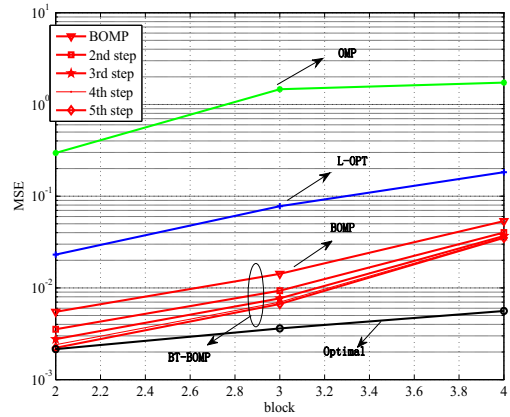


Fig. 4. Performance of BOMP, BT-BOMP, L-OPT for a dictionary with $M=180$, $N=960$ and $d=16$ under various block-sparsity levels

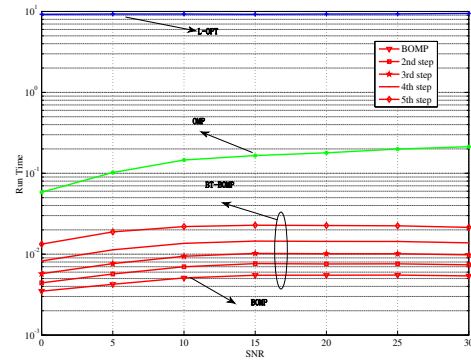


Fig. 5. Average Recovery Time of BOMP, BT-BOMP, L-OPT for a dictionary with $M=180$, $N=960$ and $d=16$

C. Complexity Comparison

In Fig.5, the time consumptions of the algorithms are compared, where we use the same computer and record the processing time for each algorithm. It is obvious that both the L-OPT and the OMP algorithms require long recovery time comparing to the BT-BOMP and BOMP algorithms. The BT-BOMP algorithm requires a little more time than the BOMP algorithm since it needs more time for multi-step processing.

D. Discussions

From the practical perspective, the most prominent feature of the BT-BOMP is that it shows better robustness against noise. We also want to point out that the BT-BOMP algorithm provides a generalized framework for all sparse signals. Note that even the block sparse signal exhibits irregular structure, the BT-BOMP algorithm still provides an accurate recovery according to the multi-step process. Besides, it turns out to the conventional OMP algorithm with $d=1$. As such, the BT-BOMP algorithm always provides more accurate solution than the BOMP algorithm with the cost of a little more complexity.

Comparing to the ℓ_2/ℓ_1 algorithm and the conventional *OMP* algorithm, it dominates in both performance and complexity.

V. CONCLUSIONS

In this paper, a binary tree-based block orthogonal matching pursuit algorithm is proposed for block sparse signals under the compressed sensing framework. The BT-BOMP algorithm is suitable for the generalized block sparse signals, where the size of non-zero blocks may be different. By the multi-step processing, the BT-BOMP algorithm may not need the knowledge of block structure and obtains the robustness against noise. The simulations show that the BT-BOMP algorithm is much better than the current state algorithm for block sparse signals.

ACKNOWLEDGEMENT

This work is supported in part by the Fundamental Research Funds for the Central Universities (BUPT2009RC0107), the National Basic Research Program of China (973 Program) (2009CB320400), the Important National Science & Technology Specific Projects (2009ZX03007-004) and the Joint Funds of NSFC-Guangdong under Grant U1035001.

REFERENCES

- [1] D. L. Donoho, "Compressed sensing," *IEEE Trans. on Information Theory*, vol. 52, pp. 1289-1306, Apr. 2006.
- [2] E. J. Cands, 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, Feb. 2006.
- [3] E. J. Candes and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Trans. on Information Theory*, vol. 52, pp. 5406-5452, 2006.
- [4] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *Constr. Approx.*, vol. 13, no. 1, pp. 57-98, 1997.
- [5] J. Tropp, "Greed is good: Algorithmic results for sparse approximations," *IEEE Trans. Inf. Theory*, , vol. 50, no. 10, pp. 2232-2242, Oct. 2004.
- [6] S. S. Chen, Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, pp. 33-61, 1998
- [7] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," *Proc. 27th Asilomar Conf. Signals, Systems, Comput.*, Nov. 1993, pp. 40-44.
- [8] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. on Information Theory*, pp. 4655-4666, 2007.
- [9] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comput. Harmon. Anal.*, vol. 26, no. 3, pp. 301-321, May 2009.
- [10] M. Mishali and Y. C. Eldar, "Blind multi-band signal reconstruction: Compressed sensing for analog signals," *IEEE Trans. on Signal Process.*, vol.57, no.3, pp.993-1009, Mar. 2009.
- [11] M. Mishali and Y. C. Eldar, "From theory to practice: Sub-Nyquist sampling of sparse wideband analog signals," *IEEE J. Sel. Topics Signal Process.*, vol.4, no.2, pp.375-391, Apr. 2009.
- [12] F. Parvaresh, H. Vikalo, S. Misra, and B. Hassibi, "Recovering sparse signals using sparse measurement matrices in compressed DNA microarrays," *IEEE J. Sel. Topics Signal Process.*, vol.2, no.3, pp.275-285, Jun. 2008.
- [13] M. A. Davenport and M. B. Wakin, "Analysis of Orthogonal Matching Pursuit Using the Restricted Isometry Property," *IEEE Trans. Inf. Theory*, vol. 56, pp. 1982-2001, 2010.
- [14] M. Stojnic, F. Parvaresh, and B. Hassibi, "On the reconstruction of block-sparse signals with an optimal number of measurements," *IEEE Trans. Signal Process.*, vol. 57, no. 8, pp. 3075-3085, May 2010.
- [15] Y. C. Eldar and M. Mishali, "Robust recovery of signals from a structured union of subspaces," *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 5302-5316, Nov. 2009.
- [16] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Appl. Comput. Harmon. Anal.*, vol. 27, no. 3, pp. 265-274, 2009.
- [17] Eldar, Y.C., Kuppinger, P., Boleskei, H., "Block-Sparse Signals: Uncertainty Relations and Efficient Recovery," *IEEE Trans. Signal Process.*, vol. 58, pp. 3042-3054 , 2010.
- [18] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," available at <http://cvxr.com/cvx>, December 2010.