Compressed Sensing Recovery Algorithms and VLSI Implementation

Kuan-Ting Lin, Kai-Jiun Yang, Pu-Hsuan Lin and Shang-Ho Tsai
Department of Electrical Engineering, National Chiao Tung University, Hsinchu, Taiwan
E-mail: kaijiuny.ece98g@g2.nctu.edu.tw, oxygen.ece92@nctu.edu.tw and shanghot@mail.nctu.edu.tw

***Abstract*—This paper proposes two recovery algorithms modified from subspace pursuit(SP) for compressed sensing problems. These algorithms can reduce the complexity of SP and maintain high recovery rate. Complexity analysis and simulation results are provided to demonstrate the improvements. Additionally this work has implemented the VLSI circuit APR of the proposed algorithm using TSMC 90 nm process. The target clock frequency is 100MHz, and the corresponding APR dimension is $11.69mm^2$. Based on the post-layout simulation the average power consumption is 431 mW.**

I. Introduction

Compressed sensing is a kind of algorithm for sparse signal processing, and lately it has been feverishly discussed in image, radar, and other communication systems [1]. The algorithm states that the signal sampling can be below Nyquist frequency if the signal is sparse enough. However, the signal recovery needs more computing efforts. The recovery problem can be viewed as l_0-optimization [2]. Unfortunately, the l_0-optimization problem is NP-hard. To solve the problem, Cades and Tao found that the l_0-optimization problem can be transferred to l_1-optimization problem as long as the sensing matrix satisfied Restricted Isometry Property (RIP) and can be solved by linear programming (LP) technique [3]. However, the complexity of the procedure which obtains LP solution is still too high. The physical implementation of the recovery becomes impractical due to the cost.

To overcome the problem, several LP-based recovery techniques were proposed as alternatives. Orthogonal matching pursuit (OMP) [4] which is iterative greedy algorithm becomes popular for solving CS problem. The prerequisite of using OMP was defined in [5]. These algorithms are favored because of low complexity and simple geometric interpretation. Therefore the hardware implementation becomes feasible when these algorithms are applied.

Subspace pursuit (SP) algorithm [6] greatly improves the recovery rate and its complexity is comparable to that of OMP and much lower than that of LP. However, when it comes to hardware implementation, the complexity is still unacceptable. On the other hand gradient pursuit (GP) [7] uses simpler method but the results are coarse. In this work we propose two recovery algorithms to reduce the complexity, namely gradient subspace pursuit (GSP) and modified subspace pursuit (MSP). GSP works well for zero-one signals while MSP is for Gaussian-distributed signals.

The following notations shall run through the contents: the dimensions of the original signal $\mathbf{x}$ and the received signal $\mathbf{y}$ are $N \times 1$ and $M \times 1$ respectively. K denotes the sparsity of $\mathbf{x}$. $\mathbf{\Phi}$ is the $M \times N$ sensing matrix. In recovery calculation, $\mathbf{r}$ is an $M \times 1$ vector and represents the residual at each iteration. Ω is the selected index set, which contains the indices obtained by the recovery algorithm at each iteration. The symbol $\mathbf{\Phi}_\Omega$ denotes the sub-matrix indexed by Ω, and the superscript represents the result at the specified iteration. For example, if Ω equals to $\{1, 3, 6\}$, $\mathbf{\Phi}_\Omega$ is a matrix combined with the first, the third and the sixth columns in $\mathbf{\Phi}$, and $\mathbf{r}^n$ is the residual at the n iteration. Inner product is symbolized by angle brackets e.g. $<\mathbf{x}\ ,\ \mathbf{y}> = \mathbf{x}^T\mathbf{y}$. Pseudo inverse is written as $\mathbf{\Phi}^\dagger$ i.e. $\mathbf{\Phi}^\dagger = (\mathbf{\Phi}^H\mathbf{\Phi})^{-1}\mathbf{\Phi}$.

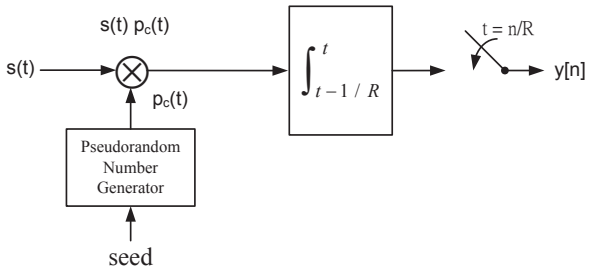

Fig. 1. Block diagram of a random demodulator.

II. System Model

Random demodulator (RD) was proposed for sub-Nyquist sampling to compress $\mathbf{x}$ into $\mathbf{y}$ [8], and the structure is as shown in Fig. 1. RD demodulates the input signal by multiplying a random sequence. The demodulated signal is passed to a low-pass filter, and output signal is acquired by relatively low sampling rate. The sensing matrix satisfies the property of RIP and consists the matrixes $\mathbf{H}$, $\mathbf{D}$, and $\mathbf{F}^{-1}$, where $\mathbf{H}$ represents a low-pass filter and sampling, $\mathbf{D}$ represents the random sequence, and $\mathbf{F}$ is the discrete fourier transform matrix. Random demodulator can be mathematically expressed as a diagonal matrix $\mathbf{D}$

$$\mathbf{D} = \begin{pmatrix} a_1 & 0 & \dots & 0 \\ 0 & a_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_N \end{pmatrix}$$

where $a_1, a_2, \dots, a_N$ are $\{+1, -1\}$ random values. The effects of the low-pass filter and the low-rate sampler shape the $M \times N$

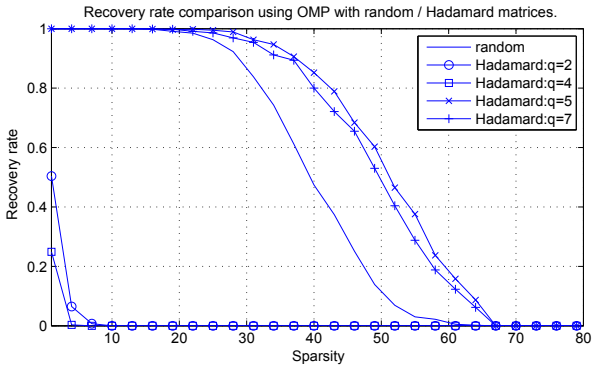

Fig. 2. Recovery rate of OMP caused by random matrix and Hadamard matrix which $N = 256$ and $M = 128$.

matrix **H**. Assuming $N/M = 2$, **H** becomes

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & 1 \end{pmatrix}$$

The whole process is equivalent to

$$\mathbf{y} = \mathbf{HDs}. \tag{1}$$

The sparsity of the signal is characterized in frequency domain to be consistent to the CS model. Assuming **x** is the frequency domain representation of **s**, then

$$\mathbf{x} = \mathbf{Fs}, \tag{2}$$

where **F** is an $N \times N$ discrete Fourier transform matrix. As a result, (1) is rewritten as

$$\mathbf{y} = \mathbf{HDF}^{-1}\mathbf{x} = \mathbf{\Phi x}, \tag{3}$$

where $\mathbf{HDF}^{-1}$ can be regarded as $\mathbf{\Phi}$ in the CS structure and the matrix satisfies the RIP such that the system can recover **x** from **y**. Fig. 2 shows the recovery rate caused by random matrix and Hadamard matrix, where q is the index of the selected rows in Hadamard matrix. For instance, if $q = 5$, the columns with indices $\{5, 10, ..., 5M\}$ are selected and wrapped as the measurement matrix. Observe that the OMP recovery rate is low when q is a multiple of 2, but the recovery rate is better than that of random matrix when q is not a multiple of 2.

III. Determination of Random Sequence

The recovery rate is affected by the recovery algorithm and the sensing matrix. Different sensing matrixes lead to different recovery performance. The correlation among the columns within the sensing matrix should be low such that the corresponding signals can be mutually independent. The indices which indicate the location of the sparse signal are generated through the summation of the inner product **Ds**. If the correlation of the two columns in the sensing matrix is high, the summations shall be alike. As a result, it is difficult to identify which column contributes more.

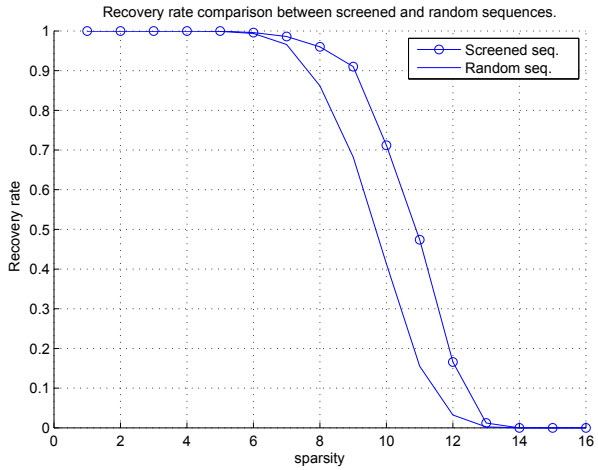

Fig. 3. The recovery rate evaluated by the screened random sequence and arbitrary random sequence which *N*=64 and *M*=32.

Reducing the correlation of the column pair in the sensing matrix improves the recovery rate. Observing the elements in the sensing matrix $\mathbf{\Phi} = \mathbf{HDF}^{-1}$, where $\mathbf{F}^{-1}$ is a fixed matrix, and **H** is also a fixed matrix if the size M and N are decided. **D** is the random sequence, which is the only configurable element in the sensing matrix. The challenge of reducing the correlation is now reduced to finding appropriate random sequence.

The diagonal matrix **D** is generated by randomizing the diagonal elements with +1 and -1. Applying the max-min method as shown in Algorithm 1, the column-wise low correlation **D** is applied for the following CS computation. During the simulation *L*=10000 random sequences were tested to find the desired **D**. Afterwards the column-wise low correlation $\mathbf{\Phi}$ can provide better recovery rate. Fig. 3 shows that the recovery rate evaluated by the screened random sequence and arbitrary random sequence.

Algorithm 1 Max-min algorithm for random sequence selection.

Input: L random sequences

Output: $\mathbf{D}_{\text{opt}}$: the random matrix with minimal correlation.

1) **Initialization**:
   Generate $\mathbf{\Phi}_1$, $\mathbf{\Phi}_2$, ... , $\mathbf{\Phi}_L$ from L random sequences, which $\mathbf{\Phi}_i$ = $[v_1^i, v_2^i, \dots, v_N^i]$ is column vector set.
2) **Correlation Calculation**
   $C_k = \max |(v_i^k)^H v_j^k|$,
   where $k \in \{1, \cdots, L\}$ and $i, j \in \{1, \cdots, N\}$
3) **Choose the best sequence**
   $\mathbf{D}_{\text{opt}} = \arg\min_k C_k$, where $k \in \{1, \cdots, N\}$

IV. Proposed Algorithms

Subspace pursuit algorithm provides high recovery rate, but its complexity is too high to be implemented into hardware.

We propose two modified algorithms: gradient subspace pursuit(GSP) and modified subspace pursuit (MSP) to minimize the complexity as follows:

A. Gradient Subspace Pursuit

SP algorithm provides $2K$ indices to obtain better location information at the cost of complex $M \times 2K$ pseudo inverse matrix computing. However, only K indices which correspond to the elements with the largest magnitude are used. GP algorithm uses vector orientation for updating and does not use matrix inversion. Although the result is rough but the complexity is greatly reduced. Since the precision of the recovered magnitudes can be refined during the iteration in SP, the initial estimation needs not to be precise. Therefore, we proposed to combine the features in SP and GP to have a good balance. The combined gradient subspace pursuit (GSP) algorithm is detailed in Algorithm 2. Fig. 4 shows the flow of GSP and the simulation results are as shown in Fig. 5 and 6.

Algorithm 2 Gradient Subspace Pursuit

Input: K, $\mathbf{\Phi}$, $\mathbf{y}$

Output: $\mathbf{x}$ with $\mathbf{x}(n) = \mathbf{x}^c$ for $n \in \Omega^c$ and $\mathbf{x}(n) = 0$ otherwise.

1) **Initialization**:
   $\Omega^0 = \{\text{Indices of } K \max | < \mathbf{y}, \mathbf{\Phi} > |\}$.
   The residual $\mathbf{r}^0 = \mathbf{y} - \mathbf{\Phi}_{\Omega^0}\mathbf{\Phi}^{\dagger}_{\Omega^0}\mathbf{y}$.
   Set the counter $c = 1$.
2) **Identify1**:
   $\hat{\Omega}^c = \Omega^{c-1} \cup \{\text{Indices of } K \max | < \mathbf{r}^{c-1}, \mathbf{\Phi} > |\}$.
3) **Estimate1**:
   Initializing $\Omega^0_{temp} = \emptyset$, $\mathbf{p}^0 = \mathbf{r}^{c-1}$ and $\mathbf{v} = 0$

for $w = 1$ to $2K$ **do**
   $\Omega^w_{temp} = \Omega^{w-1}_{temp} \cup \{\text{Index of } \max | < \mathbf{p}^{w-1}, \hat{\mathbf{\Phi}} > |\}$.
   Update the orientation $\mathbf{d}_{\Omega^w_{temp}} = \mathbf{\Phi}^H_{\Omega^w_{temp}}\mathbf{p}^{w-1}$.
   $\mathbf{z}^w = \mathbf{\Phi}_{\Omega^w_{temp}}\mathbf{d}_{\Omega^w_{temp}}$.
   $a^w = < \mathbf{p}^{w-1}, \mathbf{z}^w > / \parallel \mathbf{z}^w \parallel_2^2$.
   $\mathbf{v}^w = \mathbf{v}^{w-1} + a^w \mathbf{d}_{\Omega^w_{temp}}$ and $\mathbf{p}^w = \mathbf{p}^{w-1} - a^w\mathbf{z}^w$.
   $\mathbf{t}^c = \mathbf{v}^w$
end for

4) **Identify2**: $\Omega^c = \{\text{Indices of } K \max \mathbf{t}^c\}$.
5) **Estimate 2 and iterate** :
   $\mathbf{x}^c = \mathbf{\Phi}^{\dagger}_{\Omega^c}\mathbf{y}$, $\mathbf{r}^c = \mathbf{y} - \mathbf{\Phi}_{\Omega^c}\mathbf{x}^c$. and $c = c + 1$
   Repeat (2)-(5) until $\parallel \mathbf{r}^c \parallel_2 > \parallel \mathbf{r}^{c-1} \parallel_2$.

From the simulation results, if the signal is zero-one signal, the recovery rate of the proposed GSP is almost the same as that of SP. That is because the errors caused by the coarse estimation in Estimation 1 only slightly affect the zero-one signal, and the number of the estimated signal is still correct. Therefore the recovery rate of the proposed GSP does not degrade.

When the signal is Gaussian-distributed signal, the recovery rate of GSP degrades as shown in Fig. 6. The degradation is caused by the Estimation 1 in GSP. Gaussian-distributed signal is likely to have the extremely small magnitude, and the small signal may cause errors during Estimation 1 in GSP, such that

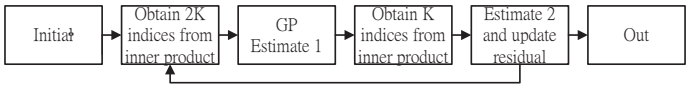

Fig. 4. The simple flow of GSP.

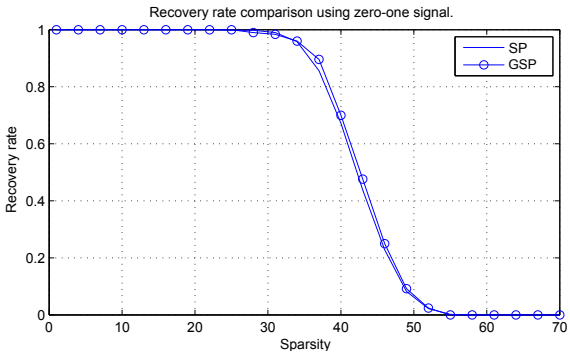

Fig. 5. The recovery rate of SP and GSP using zero-one signals. N=256, M=128. The slight difference between the recovery rate is due to different random seeds of input signals.

the number the recovered signal can be wrong. When the size is incorrect, the recovery fails.

B. Modified Subspace Pursuit

We propose a low complexity algorithm which performs well for Gaussian-distributed signals. Unlike GSP, exact estimation in Estimation 1 is preserved. Hence the pseudo-inverse matrix is inevitable. To practically implement the hardware, the pseudo matrix inversion needs to be re-formulated.

The matrix inversion lemma is as shown in (4). The computation of $(\mathbf{C}-\mathbf{VA}^{-1}\mathbf{U})^{-1}$ can be complicated if the dimension is large, and it is essential for acquiring other members in the matrix. Since inversion of 2×2 matrix is simple, two columns in the sensing matrix are selected at each iteration in the process of SP algorithm. After K columns are processed, the columns are updated as they are in SP algorithm. The difference between SP and our proposed MSP is that SP chooses K indices with relatively big magnitude from $2K$ indices, but the MSP chooses K indices with relatively large magnitude from K+2 indices. Fig. 7 shows the flow of proposed MSP and the

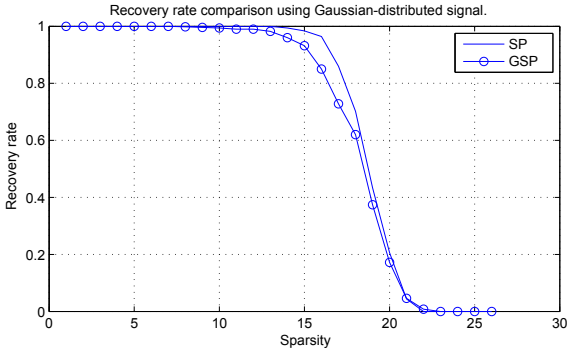

Fig. 6. The recovery rate of SP and GSP using Gaussian-distributed signals. N=256, M=128.

$$\begin{bmatrix} \mathbf{A} & \mathbf{U} \\ \mathbf{V} & \mathbf{C} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{U}(\mathbf{C} - \mathbf{V}\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}\mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{U}(\mathbf{C} - \mathbf{V}\mathbf{A}^{-1}\mathbf{U})^{-1} \\ -(\mathbf{C} - \mathbf{V}\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}\mathbf{A}^{-1} & (\mathbf{C} - \mathbf{V}\mathbf{A}^{-1}\mathbf{U})^{-1} \end{bmatrix} \tag{4}$$

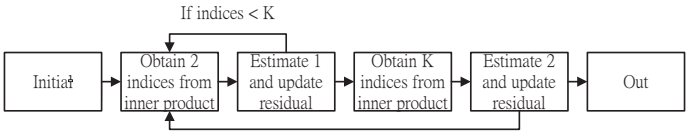

Fig. 7. The flow of proposed modified SP.

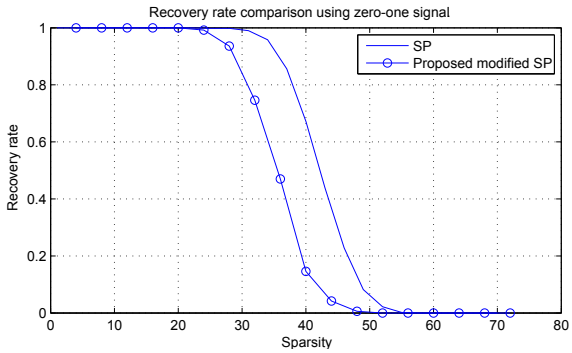

Fig. 8. The recovery rate of SP and proposed modified SP for zero-one signals. *N*=256, *M*=128.

algorithm is detailed in Algorithm 3. Though the modification might slow down the speed of convergence comparing to SP algorithm, the recovery rate is slightly better than that of SP for Gaussian-distributed signals as shown in Fig. 9. Moreover, the complexity is much lower than that of SP algorithm.

Algorithm 3 Modified Subspace Pursuit

Input: K, $\mathbf{\Phi}$, $\mathbf{y}$

Output: $\mathbf{x}$ with $\mathbf{x}(n) = \mathbf{x}^c$ for $n \in \Omega^c$ and $\mathbf{x}(n) = 0$ otherwise.

1) **Initialization 1**: $\Omega^0 = \emptyset$ and the residual $\mathbf{r}^0 = \mathbf{y}$

for $c = 1$ to $c \leq K$ **do**

2) **Identify**:
$\Omega^c = \Omega^{c-1} \cup \{\text{Indices of } 2 \max | < \mathbf{r}^{c-1}, \mathbf{\Phi} > |\}$.

3) **Estimate 1**: $\mathbf{x}^c = \mathbf{\Phi}^{\dagger}_{\Omega^c}\mathbf{y}$.

4) **Iterate 1**: Update the residual: $\mathbf{r}^c = \mathbf{y} - \mathbf{\Phi}_{\Omega^c}\mathbf{x}^c$.

end for

5) **Initialization 2**: $\mathbf{p}^0 = \mathbf{r}^c$, $\hat{\Omega}^0 = \Omega^c$.

for $w = 1$ to $\Omega^w = \Omega^{w-1}$ **do**

6) **Check Indices** :
$\hat{\Omega}^w = \hat{\Omega}^{w-1} \cup \{\text{Indices of } 2 \max | < \mathbf{p}^{w-1}, \mathbf{\Phi} > |\}$.

7) **Estimate 2**: $\mathbf{t}^w = \mathbf{\Phi}^{\dagger}_{\hat{\Omega}^w}\mathbf{y}$.

8) **Identify 2**: $\Omega^w = \{\text{Indices of } K \max \mathbf{t}^w\}$.

9) **Estimate 3**: $\mathbf{x}^w = \mathbf{\Phi}^{\dagger}_{\Omega^w}\mathbf{y}$,

10) **Iterate 2**: $\mathbf{r}^w = \mathbf{y} - \mathbf{\Phi}_{\Omega^w}\mathbf{x}^w$.

end for

The proposed modified SP performs better in Gaussian-distributed signal than in zero-one signal. Additionally the recovery rate is lower than that of SP in zero-one signals as shown in Fig 8, but it is still much better than that of OMP.

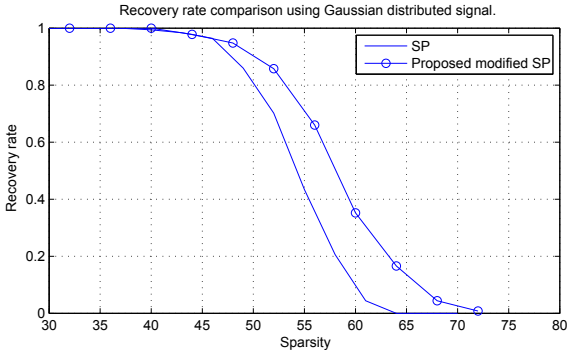

Fig. 9. The recovery rate of SP and proposed modified SP for Gaussian-distributed signals. *N*=256, *M*=128.

V. VLSI Implementation

The proposed MSP algorithm has been implemented into VLSI circuit using TSMC $90nm$ process with target clock rate at 100MHz. The dimension of the APR is $3.421mm \times 3.420mm$ and the gate count is $20300K$. Based on the post-simulation the estimated power consumption is 431mW.

VI. Conclusions

In this paper, we propose two recovery algorithms, namely gradient subspace pursuit and modified subspace pursuit. GSP performs better when the signal is zero-one signal, while MSP suits to Gaussian-distributed signal. To reduce the hardware complexity in implementing MSP algorithm, the matrix inversion lemma is applied. All of these methods provide feasible hardware solutions with comparable recovery rate.

References

[1] D. Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol.52, no.4, pp. 1289-1306, Apr. 2006.

[2] R. Venkataramani and Y. Bresler, "Sub-Nyquist sampling of multiband signals: perfect reconstruction and bounds on aliasing error," *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, vol.3, pp.1633-1636, 12-15 May 1998.

[3] E.J. Candes and T. Tao, "Decoding by linear programming," *Information Theory, IEEE Transactions on*, vol.51, no.12, pp.4203-4215, Dec. 2005.

[4] J.A. Tropp and A.C. Gilbert, "Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit," *Information Theory, IEEE Transactions on* , vol.53, no.12, pp.4655-4666, Dec. 2007.

[5] J.A. Tropp, "Greed is good: algorithmic results for sparse approximation," *Information Theory, IEEE Transactions on*, vol.50, no.10, pp.2231-2242, Oct. 2004.

[6] Dai Wei and O. Milenkovic, "Subspace Pursuit for Compressive Sensing Signal Reconstruction," *Information Theory, IEEE Transactions on*, vol.55, no.5, pp.2230-2249, May 2009.

[7] T. Blumensath, M.E. Davies, "Gradient Pursuits," *Signal Processing, IEEE Transactions on*, vol.56, no.6, pp.2370,2382, June 2008.

[8] J.A. Tropp, J.N. Laska, M.F. Duarte, J.K. Romberg, R.G. Baraniuk, "Beyond Nyquist: Efficient Sampling of Sparse Bandlimited Signals," *Information Theory, IEEE Transactions on*, vol.56, no.1, pp.520-544, Jan. 2010.