DENOISING USING MULTI-STAGE RANDOMIZED ORTHOGONAL MATCHING PURSUIT

Stefanos Koskinas and Ioannis Psaromiligkos

Department of Electrical & Computer Engineering, McGill University, Montreal, Quebec, Canada
Email: stefanos.koskinas@mail.mcgill.ca, yannis@ece.mcgill.ca

ABSTRACT

Orthogonal Matching Pursuit (OMP) can denoise a signal by greedily approximating a least-squares (LS) estimate as a linear combination of elements (atoms) of a dictionary. OMP iteratively decomposes a signal through deterministic atom selections at each iteration step. Recently proposed randomized OMP algorithms employ random atom selections instead and have the potential to further improve denoising. Typically, the best approximation from these algorithms can be obtained only within a narrow range of iterations. In this paper, we propose a novel multi-stage randomized OMP (MS-ROMP) denoising approach that performs successive ROMP runs, each denoising the obtained estimate from the previous one. We show through simulations that, under certain conditions, this can significantly improve denoising performance by producing a good approximation after any number of iterations beyond the sparsity level.

***Index Terms*—** Greedy approximation, orthogonal matching pursuit, randomized algorithms, signal denoising

1. INTRODUCTION

Denoising, i.e., extracting a clean signal from its noisy observations, is a vital task in the field of signal processing. The need for computationally efficient methods has led to the use of greedy approaches, such as Matching Pursuit (MP) [1], Orthogonal Matching Pursuit (OMP) [2], and other variants [3–6]. These approaches approximate the least squares (LS) solution by iteratively decomposing a signal into a linear combination of elements from a dictionary, selecting one element per iteration. Compared to direct LS, they require less computation by scaling down the matrix inversions required, but they can also provide better estimates as they implicitly perform regularization and, hence, avoid overfitting. Greedy approximation algorithms have also been investigated with respect to the popular and related compressive sensing topic [5, 7, 8].

In this paper, we consider OMP-based denoising algorithms that have demonstrated encouraging results, especially for the case of sparse signals [9, 10]. In the original OMP form [2], the atoms are selected from the dictionary deterministically. More recent work [11–13] has proposed selecting atoms randomly instead, which has shown potential to further improve denoising performance. Of particular interest to us is the recently proposed RandOMP algorithm [12]. RandOMP produces multiple decompositions that are differentiated by selecting atoms randomly according to a distribution that is exponentially biased towards larger projections. The decompositions are then averaged into a single one that can provide a better denoised signal than the OMP decomposition.

However, the quality of the estimate is sensitive to the number of iterations executed, since it depends on the sparsity and noise levels. In fact, a good approximation can be obtained only within a narrow range of iterations, after a good representation of the noiseless signal has been constructed but before overfitting to match the observation. In this paper, we modify the distribution of selecting atoms randomly into a linear bias towards larger projections and propose a novel algorithm with random atom selections and multiple stages[1], hence named multi-stage randomized OMP (MS-ROMP), in order to widen the range of iterations wherein a good estimate can be obtained. Each stage performs an independent ROMP run, which aims at decomposing the obtained estimate of the previous stage instead of the original observation. By iteratively decomposing the obtained estimate, noise components can be permanently discarded, thus desensitizing the dependence on the noise level and improving performance for a wider range of iterations, so that we may stop the algorithm after any number of iterations (beyond a certain minimum).

This paper is structured as follows. In Section 2, we define the denoising problem and describe RandOMP. In Section 3, we modify RandOMP's distribution of selecting atoms and propose our novel MS-ROMP algorithm. Finally, simulation results are discussed in Section 4 and our conclusions are summarized in Section 5.

2. PROBLEM FORMULATION AND BACKGROUND

2.1. Problem formulation

Let $x \in \mathbb{R}^N$ be an unknown signal that is corrupted by additive white Gaussian noise (AWGN) $n \in \mathbb{R}^N$, yielding the observed noisy signal $y = x + n \in \mathbb{R}^N$. Our goal is to extract the signal component x in y, i.e., denoise y.

We assume that x is k-sparse over a known dictionary D, that is, it can be expressed as a linear combination of k elements (referred to as *atoms*) in D. We will represent D as a matrix $D \in \mathbb{R}^{N \times M}$ whose M column vectors, denoted d_j, $j = 1, ..., M$, represent the dictionary atoms and are assumed to be normalized, i.e., $\|d_j\|^2 = 1$. We also assume that D is *over-complete*, i.e., $M > N$, so that any signal in $\mathbb{R}^N$ can be represented by more than one linear combination of atoms in D. The assumption that x is k-sparse over D translates to the existence of a vector $\alpha \in \mathbb{R}^M$ containing k non-zero elements, such that $x = D\alpha$.

[1] Stages here refer to successive algorithm runs; this differs from some cases in the literature, e.g., StOMP [6], where stages replace iterations.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

2.2. Background: RandOMP-based denoising

RandOMP [12] is an iterative procedure for the expansion of y as a linear combination of atoms, described as follows. Let the initial residual be $r^{(0)} = y$, and suppose that at the $(i-1)$th iteration we have the following decomposition of y

$$y = D^{(i-1)}\alpha^{(i-1)} + r^{(i-1)}, \tag{1}$$

where $D^{(i-1)}$ is the sub-matrix of D containing the atoms so far selected, and $\alpha^{(i-1)}$ is the vector containing the corresponding coefficients. The residual $r^{(i-1)}$ is then used to select the next atom to be included in the decomposition of y. Instead of selecting the atom that maximizes $|\langle r^{(i-1)}, d^{(i)}\rangle|$ as in OMP [2], the selection is made randomly, so that, the larger the projection on an atom, the higher the probability of choosing that atom. More specifically, the probability of selecting the jth atom, $j = 1, ..., M$, is set to be exponentially proportional to its projection magnitude, that is

$$p_{\text{exp}}(j) \propto \exp\left(c\left|d_j^T r^{(i-1)}\right|^2\right), \quad c = \frac{\sigma_x^2}{2\sigma^2(\sigma_x^2+\sigma)^2}, \tag{2}$$

where σ_x^2 and σ^2 are the signal and noise variances, respectively, and $\left|d_j^T r^{(i-1)}\right|$ are the projection magnitudes.

Finally, as in OMP, y is projected on the set of selected atoms by solving the minimization problem

$$\alpha^{(i)} = \arg\min_{\alpha\in\mathbb{R}^i} \|D^{(i)}\alpha - y\|_2^2, \tag{3}$$

thus producing a new decomposition and residual as in (1). The solution in (3) makes $r^{(i)}$ orthogonal to all atoms in $D^{(i)}$, thus producing zero-coefficient projections on all atoms in $D^{(i)}$ in the next iteration $(i+1)$. Consequently, atoms can only be chosen once, and RandOMP guarantees perfect decomposition of y (that is, zero residual) after N iterations (the signal's dimension) [2]. We note that $D^{(i)}\alpha^{(i)}$ can be generalized to $D\alpha$; the coefficients in α at indices that correspond to non-selected atoms are set at zero, thus cancelling the respective columns (atoms) in D. Then, the solution α produces the estimate $\hat{x} = D\alpha$.

RandOMP exploits multiple runs (each decomposing the same observation y), referred to here as *paths*, to produce different solutions. Let α_ℓ be the solution vector of the ℓth path for $\ell = 1, ..., L$ paths. It has been shown in [12] that averaging the solutions as

$$\hat{\alpha} = \frac{1}{L}\sum_{\ell=1}^{L}\alpha_\ell. \tag{4}$$

can produce a better estimate $\hat{x} = D\hat{\alpha}$ of x. That is, the estimate error to x from the RandOMP solution $\hat{\alpha}$, $\|D\hat{\alpha} - x\|$, is smaller than the error from the OMP solution α, $\|D\alpha - x\|$, produced from (locally) optimal atom selections[2]. We note that individual path solutions are constrained to be sparse similar to the OMP solution, but the average solution may not be sparse.

The algorithm runs for a pre-specified number of iterations per path. Assuming that x is k-sparse on D, we may stop each path after k iterations to obtain an estimate that is k-sparse over D. We expect that the first k selected atoms are sufficient to represent x, while we also expect that further iterations would incorporate more noise into the estimate. This behaviour, shown in Figure 1, implies that both OMP and RandOMP approach x on the path towards y, illustrated from the shape of a 'dip'. Evidently, RandOMP's replacing of OMP's optimal deterministic selections by suboptimal random selections allows some deviation and a slower progression on the path towards y, which produces a wider 'dip', thus remaining closer to x than OMP for a larger number of iterations[3].

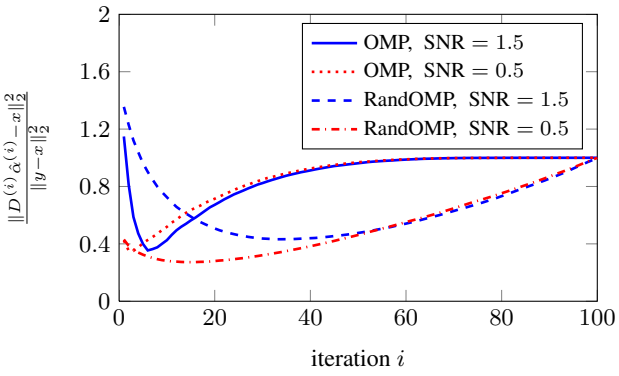

Fig. 1. Normalized RandOMP estimate error to x per iteration for two SNRs (see parameters in the simulations section).

3. THE MULTI-STAGE RANDOMIZED OMP

Random atom selections in RandOMP can improve denoising performance by allowing different paths towards y, whereby some paths may approach x more than OMP's single deterministic path. In noisy environments where x is far from y, the RandOMP estimate may remain closer to x for a larger number of iterations than the OMP estimate. The denoising performance depends on the bias towards larger projections (in this case, RandOMP's exponential bias in (2) vs. OMP's maximum projection), which, in turn, must be chosen relative to the noise level. In low-noise situations, representing x is similar to representing y, so larger projections should be preferred. Conversely, in high-noise situations, we should allow suboptimal atom selections to limit overfitting.

Although randomizing reduces this bias and widens the range of iterations wherein a good estimate can be obtained, the iteration that produces the best estimate of x within the decomposition path depends on the sparsity and noise levels, which are unknown. In this case, it is difficult to determine how many iterations are sufficient to obtain a good estimate $\hat{x}$. This translates into a stopping problem: on one hand, the number of iterations must be large enough to obtain a sufficient number of atoms that match the sparsity level and can well represent x; on the other hand, additional iterations beyond the sparsity level should be limited to minimize overfitting.

In this section, we present a novel approach to relax the stopping problem. First, we adopt the randomized approach from [12] with the exponential selection bias in (2) and the averaging of (4). However, we also consider further reducing the bias towards larger

[2]OMP selections are only optimal within each iteration since they are greedily selected instead of collaboratively (see Optimized OMP [3]).

[3]Convergence is not affected; it always occurs after N iterations.

projections, whereby we allow paths to further deviate from y in order to avoid n and possibly further improve the estimate to x. Hence, we replace the exponential bias with a linear bias so that

$$p_{\text{lin}}(j) \propto c \left| d_j^T r^{(i)} \right|^2, \quad c = \frac{\sigma_x^2}{2\sigma^2(\sigma_x^2 + \sigma)^2}. \tag{5}$$

In the remainder of this paper, we refer to ROMP, a more general version of RandOMP that can potentially employ any type of bias on random atom selections, summarized in Algorithm 1.

Algorithm 1 ROMP algorithm

procedure ROMP(D, y, T_i) ▷ inputs: dictionary D, signal y, iteration threshold T_i
- $\alpha_\ell^{(0)} \leftarrow 0$, for $\ell = 1, ..., L$ ▷ initialize path solutions
- $r_\ell^{(0)} \leftarrow y$, for $\ell = 1, ..., L$ ▷ initialize path residuals
- $i \leftarrow 1$ ▷ initialize iteration counter
- **while** $i < T_i$ **do** ▷ loop for T_i iterations
  - $d_\ell^{(i)} \leftarrow \text{rand}(d_j \in D, j = 1, ..., M)$ ▷ select atoms
  - $\alpha_\ell^{(i)} \leftarrow \arg\min_{\alpha \in \mathbb{R}^i} \|D_\ell^{(i)} \alpha - y\|_2^2$ ▷ update solutions
  - $r_\ell^{(i)} \leftarrow y - D_\ell^{(i)} \alpha_\ell^{(i)}$ ▷ update residuals
  - $i \leftarrow i + 1$ ▷ increment counter
- **end while**
- $\hat{\alpha} \leftarrow \frac{1}{L} \sum_{\ell=1}^{L} \alpha_\ell^{(i)}$ ▷ average solution
- $\hat{x} \leftarrow D\hat{\alpha}$ ▷ average estimate

end procedure

Second, we propose successive ROMP runs, referred to as *stages*, using the obtained estimate from each stage as the decomposition target for the next stage. By repeatedly stopping before N iterations through several stages, we seek to successively discard noise components from the estimate and, hence, the target. Consequently, we relax the stopping problem by minimizing the level of overfitting from additional iterations beyond the sparsity level.

Furthermore, we expect that the decomposition path may change over subsequent stages, due to altering the target. If the discarded components mainly contain noise, the algorithm may identify a better set of atoms to match x, in which case we expect further denoising improvement by repeatedly moving the decomposition target closer to x. This behaviour will be investigated in Section 4.

The proposed multi-stage ROMP (MS-ROMP) algorithm is described as follows. The 1st stage target $y^{(1)}$ is set equal to the original observation vector y. From (1), the decomposition of $y^{(1)}$ after i iterations (and after averaging over all paths) can be written as

$$y^{(1)} = D^{(i,1)} \hat{\alpha}^{(i,1)} + r^{(i,1)}, \tag{6}$$

where $r^{(i,1)}$ is the resulting residual, and the estimate is $\hat{x}^{(1)} = D^{(i,1)} \hat{\alpha}^{(i,1)}$. At stage 2, ROMP is repeated with $y^{(2)} = \hat{x}^{(1)}$ as the target, and a new estimate $\hat{x}^{(2)}$ and residual $r^{(i,2)}$ are produced.

Generally, if $\hat{\alpha}^{(i,n-1)}$ is the ROMP solution after i iterations (that is, the average solution over all paths) at the $(n-1)$th stage, then the target for the nth stage is set as

$$y^{(n)} = D^{(i,n-1)} \hat{\alpha}^{(i,n-1)} = \hat{x}^{(n-1)}, \tag{7}$$

where $\hat{x}^{(n-1)}$ is the estimate of x obtained after $n-1$ stages. At stage n, MS-ROMP approximates the minimization problem

$$\alpha_\ell^{(i,n)} = \arg\min_{\alpha \in \mathbb{R}^i} \|D_\ell^{(i,n)} \alpha - y^{(n)}\|_2^2 \tag{8}$$

for each ROMP path $\ell = 1, ..., L$, and the average solution $\hat{\alpha}^{(i,n)}$ is obtained from (4). The dictionary D is the same for all stages, and the solutions $\hat{\alpha}$ and α_ℓ, $\ell = 1, ..., L$, are reset to zero at the beginning of each stage. Finally, based on (6) and generalizing, the nth-stage estimate can be written either in terms of the previous estimate, as in

$$\hat{x}^{(n)} = \hat{x}^{(n-1)} - r^{(i,n)}, \tag{9}$$

or in terms of $y^{(1)} = y$ and the sum of the residuals, as in

$$\hat{x}^{(n)} = y - \sum_{k=1}^{n} r^{(i,k)}. \tag{10}$$

The complete MS-ROMP algorithm is summarized in Algorithm 2.

Algorithm 2 MS-ROMP algorithm

procedure MS-ROMP(D, y, T_i, T_s) ▷ inputs: dictionary D, signal y, iteration threshold T_i, stage threshold T_s
- $n \leftarrow 1$ ▷ initialize stage counter
- $y^{(n)} \leftarrow y$ ▷ initialize target
- **while** $n \leq T_s$ **do** ▷ loop for T_s stages
  - $\hat{x}^{(n)} \leftarrow \text{ROMP}(D, y^{(n)}, T_i)$ ▷ ROMP procedure
  - $n \leftarrow n + 1$ ▷ increment stage counter
  - $y^{(n)} \leftarrow \hat{x}^{(n-1)}$ ▷ set new target
- **end while**
- $\hat{x} \leftarrow \hat{x}^{(n)}$ ▷ final estimate

end procedure

In each MS-ROMP stage n, the decomposition removes some noise from the previous estimate $\hat{x}^{(n-1)}$. Setting this as the new target for the next stage permanently discards the noise in the previous residual, while the new decomposition produces a new residual. Therefore, each stage subtracts a residual from the original observation y, as seen from (10). The error $\|\hat{x} - x\|_2^2$ can be iteratively reduced if noisy components are removed from $\hat{x}^{(n)}$, thus updating the target to an estimate that is closer to x. After several stages, we expect that MS-ROMP will converge to a good estimate $\hat{x}$, hence allowing us to stop at any iteration beyond the sparsity level. Critical to MS-ROMP's performance is whether the same number of atoms is sufficient to represent x in subsequent stages. This number may change if the signal component is distorted due to repeatedly altering the target, thus requiring a different sparsity level to achieve a good estimate. If stable, then MS-ROMP can repeat until the estimate converges; otherwise, the x component in the target will be distorted through the updates, and the estimate will diverge from x.

4. SIMULATION RESULTS

In this section, we evaluate MS-ROMP's denoising performance by comparing the estimate $\hat{x} = D\hat{\alpha}$ to x through the estimate error $\|D\hat{\alpha} - x\|$. We use the following set of parameters[4]: signal dimension $N = 100$, dictionary size $M = 200$, sparsity level $k = 10$,

[4]Parameter values chosen are based on the analysis in [14].

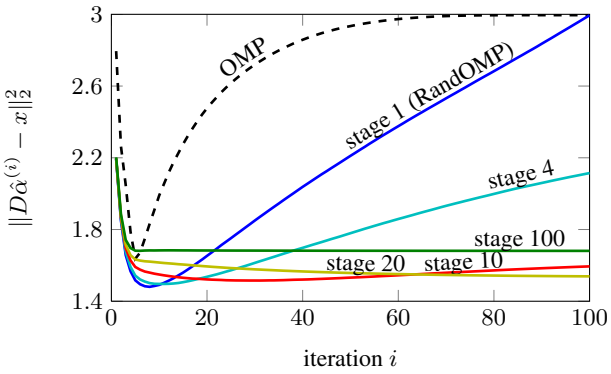

Fig. 2. MS-ROMP estimate error to x per iteration with atoms chosen exponentially at random.

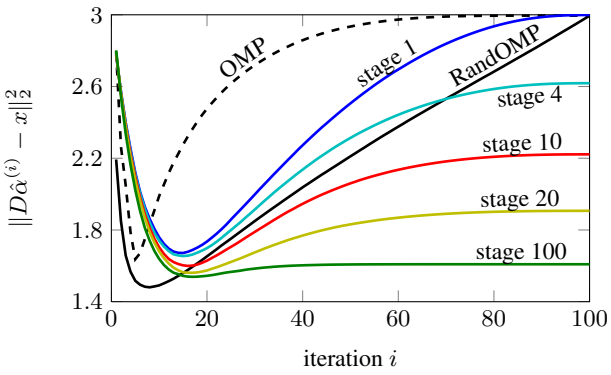

Fig. 3. MS-ROMP estimate error to x per iteration with atoms chosen linearly at random.

noise standard deviation $\sigma = 0.3$, and number of paths $L = 20$. The dictionary elements are generated using a standard normal distribution, and columns (atoms) are normalized. The signal x is formed as a k-sparse linear combination of randomly selected atoms in D, which implies an SNR of $\frac{k\sigma_x^2}{N\sigma^2} = {}^{10}/_{9}$. We use exponential and linear distributions to randomly select atoms as defined in (2) and (5), respectively. ROMP paths are averaged according to (4), while the overall MS-ROMP performance is averaged over 20 independent trials. Targets (previous stage estimates) are obtained randomly from the iteration range $i \in \{50, ..., N\}$, significantly higher than the sparsity level to ensure a good representation of x. Simulation results are presented by displaying MS-ROMP's decomposition path for several stages.

Fig. 2 shows MS-ROMP's decomposition path over multiple stages when using an exponential distribution, along with the decomposition path of OMP and RandOMP, the latter being MS-ROMP's stage 1 in this case. Initially, each additional stage lowers the estimate error for the range of iterations after the 'dip' (i.e., the iterations that are overfitting) by removing noise components from the new target when discarding the residual. However, after several stages, the altered target causes MS-ROMP to select different atoms, which causes the best-case (minimum error) estimate to deteriorate.

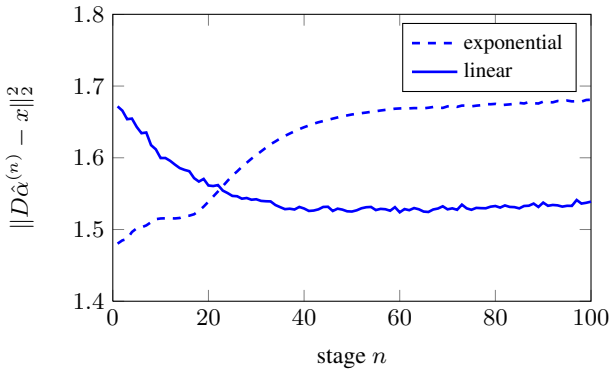

Fig. 4. MS-ROMP minimum estimate error to x per stage.

We conjecture that the discarded residual also contains some components of x, which are permanently lost; consequently, the estimate error converges to a level higher than the original minimum error.

Conversely, MS-ROMP selects a similar set of atoms over multiple stages when using a linear distribution. Fig. 3 shows that the estimate error is repeatedly improved, implying that stages here are more effective in concentrating noise within the residual[5]. Since the minimum error is further lowered, thus improving the best attainable estimate, MS-ROMP converges to an estimate error that is lower than the original minimum. This is also shown in Fig. 4 when comparing the exponential and linear distributions by displaying the minimum estimate error achieved by each stage. Although the linear case does not match RandOMP's minimum, MS-ROMP's convergence allows us to obtain a good estimate over any number of iterations beyond the 'dip' (after several stages). Therefore, the linear MS-ROMP algorithm alleviates the stopping problem as the denoising performance becomes insensitive to the number of iterations executed.

5. CONCLUSIONS

Randomized OMP-based algorithms can improve signal denoising by greedily approximating a sparse signal decomposition using random atom selections. In this paper, we described exponential and linear distributions to randomly select atoms in high-noise situations. To limit overfitting caused by additional iterations beyond the sparsity level, we proposed a novel multi-stage ROMP algorithm that uses the estimate obtained from each stage as the decomposition target for the subsequent stage, iteratively discarding noise components from the estimate. Using a linear distribution, the estimate error converges towards the minimum error attained in the decomposition path, thus allowing us to stop after any number of iterations beyond the sparsity level to obtain a good estimate.

Further theoretical analysis is necessary to quantify the underlying differences between atom selection distributions and their relation to the decomposition behaviour. Finally, mathematically relating ROMP's behaviour to the noise statistics can provide valuable information to improve algorithm design.

[5]Performance degrades for higher SNR, as the decomposition path is too slow and requires a higher projection bias, but also for fewer iterations per stage, where the signal representation is not sufficient, thus losing the 'dip'.

6. REFERENCES

[1] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[2] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *1993 Conference Record of the 27th Asilomar Conference on Signals, Systems and Computers*, vol. 1, Nov. 1993, pp. 40–44.

[3] L. Rebollo-Neira and D. Lowe, "Optimized orthogonal matching pursuit approach," *IEEE Signal Processing Letters*, vol. 9, no. 4, pp. 137–140, Apr. 2002.

[4] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *Foundations of Computational Mathematics*, vol. 9, no. 3, pp. 317–334, Apr. 2009.

[5] D. Needell and J. A. Tropp, "CoSaMP: iterative signal recovery from incomplete and inaccurate samples," *Communications of the ACM*, vol. 53, no. 12, pp. 93–100, Dec. 2010.

[6] D. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1094–1121, Feb. 2012.

[7] E. J. Candès, "Compressive sampling," in *Proceedings of the International Congress of Mathematicians*, 2006.

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

[9] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, Feb. 2009.

[10] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer, 2010.

[11] S. E. Ferrando, E. J. Doolittle, A. J. Bernal, and L. J. Bernal, "Probabilistic matching pursuit with Gabor dictionaries," *Signal Processing*, vol. 80, no. 10, pp. 2099–2120, Oct. 2000.

[12] M. Elad and I. Yavneh, "A plurality of sparse representations is better than the sparsest one alone," *IEEE Transactions on Information Theory*, vol. 55, no. 10, pp. 4701–4714, Oct. 2009.

[13] S. Li and L. Fang, "Signal denoising with random refined orthogonal matching pursuit," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 1, pp. 26–34, Jan. 2012.

[14] S. Koskinas, "Denoising using Randomized Matching Pursuit: algorithmic improvements and GPU implementation," Master's thesis, McGill University, April 2013.