

# A Fast Non-Gaussian Bayesian Matching Pursuit Method for Sparse Reconstruction

Mudassir Masood, and Tareq Y. Al-Naffouri\*, *Member, IEEE*,

## Abstract

A fast matching pursuit method using a Bayesian approach is introduced for sparse signal recovery. This method, referred to as nGpFBMP, performs Bayesian estimates of sparse signals even when the signal prior is non-Gaussian or unknown. It is agnostic on signal statistics and utilizes *a priori* statistics of additive noise and the sparsity rate of the signal, which are shown to be easily estimated from data if not available. nGpFBMP utilizes a greedy approach and order-recursive updates of its metrics to find the most dominant sparse supports to determine the approximate minimum mean square error (MMSE) estimate of the sparse signal. Simulation results demonstrate the power and robustness of our proposed estimator.

## Index Terms

sparse reconstruction, compressed sensing, Bayesian, linear regression, matching pursuit, basis selection, minimum mean-square error (MMSE) estimate, greedy algorithm

## I. INTRODUCTION

Sparsity is a feature that is abundant in both natural and man-made signals. Some examples of sparse signals include those from speech, images, videos, sensor arrays (e.g., temperature and light sensors), seismic activity, galactic activities, biometric activity, radiology, and frequency hopping. Given the vast existence of signals, their sparsity is an attractive property because the exploitation of this sparsity may be useful in the development of simple signal processing systems. Some examples of systems in which *a*

Authors are with the Department of Electrical Engineering, King Abdullah University of Science & Technology, Thuwal 23955-6900, Kingdom of Saudi Arabia, e-mail: mudassir.masood@kaust.edu.sa, tareq.alnaffouri@kaust.edu.sa

Tareq Y. Al-Naffouri is also associated with the Department of Electrical Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Kingdom of Saudi Arabia.

EDICS: MLR-BAYL, SPC-DETC, SSP-NGAU, SSP-PARE, SSP-SSAN

*priori* knowledge of signal sparsity is utilized include motion estimation [1], magnetic resonance imaging (MRI) [2], impulse noise estimation and cancellation in DSL [3], network tomography [4], and peak-to-average-power ratio reduction in OFDM [5]. All of these systems are based on sparsity-aware estimators such as Lasso [6], basis pursuit [7], structure-based estimator [8], fast Bayesian matching pursuit [9], and estimators related to the relatively new area of compressed sensing [10]–[12].

Compressed sensing (CS), otherwise known as compressive sampling, has found many applications in the fields of communications, image processing, medical imaging, and networking. CS algorithms have been shown to recover sparse signals from underdetermined systems of equations that take the form

$$\mathbf{y} = \Phi \mathbf{x} + \mathbf{n} \quad (1)$$

where  $\mathbf{x} \in \mathbb{C}^N$ , and  $\mathbf{y} \in \mathbb{C}^M$  are the unknown sparse signal and the observed signal, respectively. Furthermore,  $\Phi \in \mathbb{C}^{M \times N}$  is the measurement matrix and  $\mathbf{n} \in \mathbb{C}^M$  is the additive Gaussian noise vector. Here, the number of unknown elements,  $N$ , is much larger than the number of observations,  $M$ . CS uses linear projections of sparse signals that preserve structure of signals; furthermore, these projections are used to reconstruct the sparse signal using  $l_1$ -optimization with high probability.

$$\hat{\mathbf{x}} = \mathbf{argmin} \|\mathbf{x}\|_1 \text{ such that } \Phi \mathbf{x} = \mathbf{y} \quad (2)$$

$l_1$ -optimization is a convex optimization problem that conveniently reduces to a linear program known as basis pursuit, which has the computational complexity of  $\mathcal{O}(N^3)$ . Since, usually,  $N$  is large, such an approach rapidly becomes unrealistic. To tackle this problem, some efficient alternatives such as orthogonal matching pursuit (OMP) [13] and the algorithm proposed by Haupt et al. [14] have been proposed. These algorithms fall into the category of greedy algorithms that are relatively faster than basis pursuit. However, an inherent problem in these systems is that the only *a priori* information utilized is the sparsity information.

Another category of methods based on the Bayesian approach considers complete *a priori* statistical information of sparse signals. A method called fast Bayesian matching pursuit (FBMP) [9], adopts such an approach and assumes a Gaussian prior on the unknown sparse vector,  $\mathbf{x}$ . This method performs sparse signal estimation via model selection and model averaging. The sparse vector is described as a mixture of several components, the selection of which is based on successive model expansion. FBMP obtains an approximate MMSE estimate of the unknown signal with high accuracy and low complexity. It was shown to outperform several sparse recovery algorithms, including OMP [13], StOMP [15], GPSR-Basic

[16], Sparse Bayes [17], BCS [18] and a variational-Bayes implementation of BCS [19]. However, there are several drawbacks associated with this method. The assumption that the signal prior is Gaussian is not realistic, because, in most real-world scenarios, the signal distribution is not Gaussian, or it is unknown. In addition, its performance is dependent on the knowledge of signal statistics, which are usually difficult to compute. Although, a signal statistics estimation process is proposed, it is dependent on knowledge of the initial estimates of these signal parameters. The estimation process, in turn, has a negative impact on the complexity of the method.

Another popular Bayesian method proposed by Larsson and Selén [20] computes the MMSE estimate of the unknown vector,  $\mathbf{x}$ . Its approach is similar to that of FBMP in the sense that the sparse vector is described as a mixture of several components that are selected based on successive model reduction. It also requires knowledge of the noise and signal statistics. However, it was found that the MMSE estimate is insensitive to the *a priori* parameters and therefore an empirical-Bayesian variant that does not require any *a priori* knowledge of the data was devised.

The Bayesian approaches mentioned above work successfully only for Gaussian priors. It is reasonable to assume that any additive noise, generated at the sensing end, is Gaussian. However, assuming the signal statistics to be Gaussian is inadequate, because the actual situation is not captured. Moreover, in situations where the assumption of a Gaussian prior is valid, the parameters of that prior (mean and covariance) need to be estimated, which is challenging, especially when the signal statistics are not i.i.d. In that respect, one can appreciate convex relaxation approaches that are agnostic with regard to signal statistics.

In this paper, we pursue a Bayesian approach for sparse signal reconstruction that combines the advantages of the two approaches. On one hand, the approach is Bayesian, acknowledging the noise statistics and the signal sparsity rate, while on the other hand, the approach is agnostic on the signal amplitude statistics. The approach can bootstrap itself and estimate the required statistics (sparsity rate and noise variance) when they are unknown. The algorithm is implemented in a greedy manner and pursues an order-recursive approach, helping it to enjoy low complexity. Specifically, the advantages of our approach are as follows

- 1) The Bayesian estimate of the sparse signal is performed even when the signal prior is non-Gaussian or unknown.
- 2) Signals whose statistics are unknown are dealt with. In fact, contrary to other methods, these statistics need not be estimated, which is particularly useful when the signal statistics are not i.i.d. Therefore, it is agnostic with regard to variations in signal statistics.
- 3) The *a priori* statistics of the additive noise and the sparsity rate of the signal, which can be easily

estimated from the data if not available, are utilized.

4) The greedy nature of the approach and the order-recursive update of its metrics, make it simple.

The remainder of this paper is organized as follows. In Section II, we formulate the problem and present the MMSE setup in the non-Gaussian/unknown statistics case. In Section III, we describe our greedy algorithm that is able to obtain the approximate MMSE estimate of the sparse vector. Section IV demonstrates how the greedy algorithm can be made faster by calculating various metrics in a recursive manner. This is followed by Section V, which describes our hyperparameter estimation process. In Section VI, we present our simulation results and in Section VII, we conclude the paper.

### A. Notation

We denote scalars with small-case letters (e.g.,  $x$ ), vectors with small-case, bold-face letters (e.g.,  $\mathbf{x}$ ), matrices with upper-case, bold-face letters (e.g.,  $\mathbf{X}$ ), and we reserve calligraphic notation (e.g.,  $\mathcal{S}$ ) for sets. We use  $\mathbf{x}_i$  to denote the  $i^{\text{th}}$  column of matrix  $\mathbf{X}$ ,  $x(j)$  to denote the  $j^{\text{th}}$  entry of vector  $\mathbf{x}$ , and  $\mathcal{S}_i$  to denote a subset of a set  $\mathcal{S}$ . We also use  $\mathbf{X}_{\mathcal{S}}$  to denote the sub-matrix formed by the columns  $\{\mathbf{x}_i : i \in \mathcal{S}\}$ , indexed by set  $\mathcal{S}$ . Finally, we use  $\hat{\mathbf{x}}$ ,  $\mathbf{x}^*$ ,  $\mathbf{x}^{\text{T}}$ , and  $\mathbf{x}^{\text{H}}$  to respectively denote the estimate, conjugate, transpose, and conjugate transpose of the vector  $\mathbf{x}$ .

## II. PROBLEM FORMULATION AND MMSE SETUP

### A. The Signal Model

The analysis in this paper considers obtaining an  $N \times 1$  sparse vector,  $\mathbf{x}$ , from an  $M \times 1$  observations vector,  $\mathbf{y}$ . These observations obey the linear regression model

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} + \mathbf{n} \quad (3)$$

where  $\mathbf{\Phi}$  is a known  $M \times N$  regression matrix and  $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \mathbf{K}_{\mathbf{n}})$  is the additive Gaussian noise vector.

We shall assume that  $\mathbf{x}$  has a sparse structure and is modeled as  $\mathbf{x} = \mathbf{x}_A \circ \mathbf{x}_B$  where  $\circ$  indicates Hadamard (element-by-element) multiplication. The vector  $\mathbf{x}_A$  consists of elements that are drawn from some unknown distribution and the entries of  $\mathbf{x}_B$  are drawn i.i.d. from a Bernoulli distribution with success probability  $p$ . We observe that the sparsity of vector  $\mathbf{x}$  is controlled by  $p$  and, therefore, we call it the sparsity parameter/rate. Typically, in Bayesian estimation, the signal entries are assumed to be drawn from a Gaussian distribution but here we would like to emphasize that the distribution and color

of the entries of  $\mathbf{x}_A$  do not matter.<sup>1</sup>

### B. MMSE Estimation of $\mathbf{x}$

To determine  $\mathbf{x}$ , we compute the MMSE estimate of  $\mathbf{x}$  given observation  $\mathbf{y}$ . This estimate is formally defined by

$$\hat{\mathbf{x}}_{mmse} \triangleq \mathbb{E}[\mathbf{x}|\mathbf{y}] = \sum_{\mathcal{S}} p(\mathcal{S}|\mathbf{y}) \mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}] \quad (4)$$

where the sum is executed over all possible  $2^N$  support sets of  $\mathbf{x}$ . In the following, we explain how the expectation  $\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}]$ , the posterior  $p(\mathcal{S}|\mathbf{y})$  and the sum in (4) can be evaluated.

Given the support  $\mathcal{S}$ , (3) becomes

$$\mathbf{y} = \mathbf{\Phi}_{\mathcal{S}} \mathbf{x}_{\mathcal{S}} + \mathbf{n} \quad (5)$$

where  $\mathbf{\Phi}_{\mathcal{S}}$  is a matrix formed by selecting columns of  $\mathbf{\Phi}$  indexed by support  $\mathcal{S}$ . Similarly,  $\mathbf{x}_{\mathcal{S}}$  is formed by selecting entries of  $\mathbf{x}$  indexed by  $\mathcal{S}$ . Since the distribution of  $\mathbf{x}$  is unknown, the best we can do is to use the best linear unbiased estimate (BLUE) as an estimate of  $\mathbf{x}$ , i.e.,

$$\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}] = \left( \mathbf{\Phi}_{\mathcal{S}}^H \mathbf{\Phi}_{\mathcal{S}} \right)^{-1} \mathbf{\Phi}_{\mathcal{S}}^H \mathbf{y} \quad (6)$$

The posterior in (4) can be written using the Bayes rule as

$$p(\mathcal{S}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathcal{S})p(\mathcal{S})}{p(\mathbf{y})} \quad (7)$$

The probability,  $p(\mathbf{y})$ , is a factor common to all posterior probabilities that appear in 3 and hence can be ignored. Since the elements of  $\mathbf{x}$  are activated according to the Bernoulli distribution with success probability  $p$ , we have

$$p(\mathcal{S}) = p^{|\mathcal{S}|} (1-p)^{N-|\mathcal{S}|} \quad (8)$$

It remains to evaluate the likelihood  $p(\mathbf{y}|\mathcal{S})$ . If  $\mathbf{x}_{\mathcal{S}}$  is Gaussian,  $p(\mathbf{y}|\mathcal{S})$  would also be Gaussian and that is easy to evaluate. On the other hand, when the distribution of  $\mathbf{x}$  is unknown or even when it is known but non-Gaussian, determining  $p(\mathbf{y}|\mathcal{S})$  is in general very difficult. To go around this, we note that  $\mathbf{y}$  is formed by a vector in the subspace spanned by the columns of  $\mathbf{\Phi}_{\mathcal{S}}$  plus a Gaussian noise vector,  $\mathbf{n}$ . This

<sup>1</sup>The distribution may be unknown or known with unknown parameters or even Gaussian. Our developments are agnostic with regard to signal statistics.

motivates us to eliminate the non-Gaussian component by projecting  $\mathbf{y}$  onto the orthogonal complement space of  $\Phi_{\mathcal{S}}$ . This is done by multiplying  $\mathbf{y}$  by the projection matrix  $\mathbf{P}_{\mathcal{S}}^{\perp} = \mathbf{I} - \Phi_{\mathcal{S}} (\Phi_{\mathcal{S}}^H \Phi_{\mathcal{S}})^{-1} \Phi_{\mathcal{S}}^H$ . This leaves us with  $\mathbf{P}_{\mathcal{S}}^{\perp} \mathbf{y} = \mathbf{P}_{\mathcal{S}}^{\perp} \mathbf{n}$ , which is Gaussian with a zero mean and covariance

$$\begin{aligned} \mathbf{K} &= \mathbb{E}[(\mathbf{P}_{\mathcal{S}}^{\perp} \mathbf{n})(\mathbf{P}_{\mathcal{S}}^{\perp} \mathbf{n})^H] \\ &= \mathbf{P}_{\mathcal{S}}^{\perp} \mathbb{E}[\mathbf{n}\mathbf{n}^H] \mathbf{P}_{\mathcal{S}}^{\perp H} \\ &= \mathbf{P}_{\mathcal{S}}^{\perp} \mathbf{K}_{\mathbf{n}} \mathbf{P}_{\mathcal{S}}^{\perp H} \end{aligned} \quad (9)$$

where  $\mathbf{K}_{\mathbf{n}}$  is the noise covariance matrix. Thus,

$$p(\mathbf{y}|\mathcal{S}) \simeq \frac{1}{\sqrt{(2\pi)^M \det \mathbf{K}}} \exp\left(-\frac{1}{2} (\mathbf{P}_{\mathcal{S}}^{\perp} \mathbf{y})^H \mathbf{K}^{-1} (\mathbf{P}_{\mathcal{S}}^{\perp} \mathbf{y})\right) \quad (10)$$

Dropping the pre-exponential factor yields

$$p(\mathbf{y}|\mathcal{S}) \propto \exp\left(-\left\|\mathbf{P}_{\mathcal{S}}^{\perp} \mathbf{y}\right\|_{\mathbf{K}^{-1}}^2\right) \quad (11)$$

In the white noise case,  $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma_{\mathbf{n}}^2 \mathbf{I})$ , which we will focus on for the remainder of the paper, we have

$$p(\mathbf{y}|\mathcal{S}) \simeq \exp\left(-\frac{1}{\sigma_{\mathbf{n}}^2} \left\|\mathbf{P}_{\mathcal{S}}^{\perp} \mathbf{y}\right\|^2\right) \quad (12)$$

Substituting (8) and (12) into (7) finally yields an expression for the posterior probability. In this way, we have all the ingredients to compute the sum in (4). Computing this sum is a challenging task when  $N$  is large because the number of support sets can be extremely large and the computational complexity can become unrealistic. To have a computationally feasible solution, this sum can be computed over a few support sets corresponding to significant posteriors. Let  $\mathcal{S}^d$  be the set of supports for which the posteriors are significant. Hence, we arrive at the following approximation to the MMSE estimate

$$\hat{\mathbf{x}}_{ammse} = \sum_{\mathcal{S}^d} p(\mathcal{S}^d|\mathbf{y}) \mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}^d] \quad (13)$$

In the next section, we propose a greedy algorithm to find  $\mathcal{S}^d$ . Before proceeding, for ease of representation and convenience, we represent the posteriors in the log domain. In this regard, we define a dominant

support selection metric,  $\nu(\mathcal{S})$ , to be used by the greedy algorithm as follows:

$$\begin{aligned}
\nu(\mathcal{S}) &\triangleq \ln p(\mathbf{y}|\mathcal{S})p(\mathcal{S}) \\
&= \ln \exp\left(-\frac{1}{\sigma_{\mathbf{n}}^2} \left\| \mathbf{P}_{\mathcal{S}}^{\perp} \mathbf{y} \right\|^2\right) + \ln \left( p^{|\mathcal{S}|} (1-p)^{N-|\mathcal{S}|} \right) \\
&= \frac{1}{\sigma_{\mathbf{n}}^2} \left\| \Phi_{\mathcal{S}} (\Phi_{\mathcal{S}}^H \Phi_{\mathcal{S}})^{-1} \Phi_{\mathcal{S}}^H \mathbf{y} \right\|^2 - \frac{1}{\sigma_{\mathbf{n}}^2} \|\mathbf{y}\|^2 + |\mathcal{S}| \ln p + (N - |\mathcal{S}|) \ln(1-p) \quad (14)
\end{aligned}$$

### III. A GREEDY ALGORITHM

We now present a greedy algorithm to determine the set of dominant supports,  $\mathcal{S}^d$ , required to evaluate  $\hat{\mathbf{x}}_{ammse}$  (13). We search for the optimal support in a greedy manner. We first start by finding the best support of size 1, which involves evaluating  $\nu(\mathcal{S})$  for  $\mathcal{S} = \{1\}, \dots, \{N\}$ , i.e., a total of  $\binom{N}{1}$  search points. Let  $\mathcal{S}_1 = \{i_1^*\}$  be the optimal support. Now, we look for the optimal support of size 2, which involves a search of size  $\binom{N}{2}$ . To reduce the search space, we pursue a greedy approach and look for the point  $i_2^* \neq i_1^*$  such that  $\mathcal{S}_2 = \{i_1^*, i_2^*\}$  maximizes  $\nu(\mathcal{S}_2)$ . This involves  $\binom{N-1}{1}$  search points (as opposed to the optimal search over  $\binom{N}{2}$  points). We continue in this manner by forming  $\mathcal{S}_3 = \{i_1^*, i_2^*, i_3^*\}$  and searching for  $i_3^*$  in the remaining  $N - 2$  points and so on until we reach  $\mathcal{S}_P = \{i_1^*, \dots, i_P^*\}$ . The value of  $P$  is selected to be slightly larger than the expected number of nonzero elements in the constructed signal such that  $\Pr(|\mathcal{S}| > P)$  is sufficiently small<sup>2</sup>. A formal algorithmic description is presented in Table I and an example run of this algorithm for  $N = 7$  and  $P = 4$  is presented in Fig. 1.

One point to note here is that in our greedy move from  $\mathcal{S}_j$  to  $\mathcal{S}_{j+1}$ , we need to evaluate  $\nu(\mathcal{S}_j \cup \{i_{j+1}^*\})$  around  $N$  times, which can be done in an order-recursive manner starting from that of  $\nu(\mathcal{S}_j)$ . Specifically, we note that every expansion,  $\mathcal{S}_j \cup \{i_{j+1}^*\}$ , from  $\mathcal{S}_j$  requires a calculation of  $\nu(\mathcal{S}_j \cup \{i_{j+1}^*\})$  from (14). This translates to appending a column,  $\phi_{j+1}$ , to  $\Phi_{\mathcal{S}_j}$  in the calculations of (14), which can be done in an order-recursive manner. We summarize these calculations in Section IV. This order-recursive approach reduces the calculation in each search step to an order of  $\mathcal{O}(MN)$  operations down from  $\mathcal{O}(MN^2)$  in the direct approach. Therefore, the complexity we incur is of the order  $\mathcal{O}(PMN)$  in our greedy search for the best  $P$  support.

#### A. A Repeated Greedy Search

It is obvious that to achieve the best signal estimate,  $\mathcal{S}^d$  should contain all  $2^N$  possibilities of supports. However, our greedy algorithm would result in an approximation of the signal estimate because it selects

<sup>2</sup> $|\mathcal{S}|$ , i.e., support of the constructed signal, follows the binomial distribution  $\mathcal{B}(N, p)$ , which can be approximated by the Gaussian distribution  $\mathcal{N}(Np, Np(1-p))$  if  $Np > 5$ . For this case,  $\Pr(|\mathcal{S}| > P) = \frac{1}{2} \operatorname{erfc} \frac{P - Np}{\sqrt{2Np(1-p)}}$ .

TABLE I: The Greedy Algorithm

1) Initialize $L = \{1, 2, \dots, N\}$ , $\mathcal{S}_{max} = \{\}$ , $\mathcal{S}^d = \{\}$ , $i = 1$ , $L_i = L$ .
2) If $i > P$ , then stop.
3) Generate $\Omega = \{\mathcal{S}_{max} \cup \{\alpha_1\}, \mathcal{S}_{max} \cup \{\alpha_2\}, \dots, \mathcal{S}_{max} \cup \{\alpha_{ L_i }\} \mid \alpha_k \in L_i\}$
4) Compute $\{\nu(\mathcal{S}_k) \mid \mathcal{S}_k \in \Omega\}$ .
5) Find $\mathcal{S}^* \in \Omega$ , such that $\nu(\mathcal{S}^*) \geq \max_j \nu(\mathcal{S}_j)$ .
6) Update, $\mathcal{S}^d = \{\mathcal{S}^d, \mathcal{S}^*\}$ , $\mathcal{S}_{max} = \mathcal{S}^*$ , $L_{i+1} = L \setminus \mathcal{S}^*$ .
7) Set $i \leftarrow i + 1$ and repeat steps 2 - 7.

$S_1$	$\nu(S_1)$	$S_2$	$\nu(S_2)$	$S_3$	$\nu(S_3)$	$S_4$	$\nu(S_4)$
{1}	12.22						
{2}	12.29	{5, 1}	89.76	{5, 1, 2}	70.2		
{3}	12.11	{5, 2}	89.23	{5, 1, 3}	70.99	{5, 1, 3, 2}	110.76
{4}	12.71	{5, 3}	89.54	{5, 1, 4}	70.89	{5, 1, 3, 4}	111.09
{5}	12.89	{5, 4}	89.52	{5, 1, 6}	70.01	{5, 1, 3, 6}	110.89
{6}	12.32	{5, 6}	89.01	{5, 1, 7}	70.7	{5, 1, 3, 7}	110.24
{7}	12.81	{5, 7}	89.61				
$S_{max}$	{5}	{5, 1}		{5, 1, 3}		{5, 1, 3, 6}	
$S_d$	{{5}}	{{5}, {5, 1}}		{{5}, {5, 1}, {5, 1, 3}}		{{5}, {5, 1}, {5, 1, 3}, {5, 1, 3, 6}}	
$L_i$	$L_2 = \{1, 2, 3, 4, 6, 7\}$	$L_3 = \{2, 3, 4, 6, 7\}$		$L_4 = \{2, 4, 6, 7\}$		$L_5 = \{2, 4, 7\}$	

Fig. 1: An example run of the greedy algorithm for  $N = 7$  and  $P = 4$ 

only a handful of supports. The accuracy of the reconstructed signal is dependent on the number of support vectors in  $\mathcal{S}^d$  and may be increased by repeating the greedy algorithm a number of times (e.g.,  $D$ ). This would result in  $\mathcal{S}^d$  with  $D$  number of support vectors for each of the  $P$  sparsity levels (i.e., a total of  $PD$  supports). The selection of supports in subsequent repetitions of the algorithm is performed by making sure not to select an element at a particular sparsity level that has been selected at the same sparsity level in any of the previous repetitions. We note that a repeated greedy search in this manner would incur a complexity of order  $\mathcal{O}(DPMN)$ . For a detailed description of the steps followed by the method, the pseudocode is provided in the Appendix and the nGpFBMP code is provided on the author's website<sup>3</sup>.

<sup>3</sup>The MATLAB code of the nGpFBMP algorithm and the results from various experiments discussed in this paper are provided at <http://faculty.kfupm.edu.sa/ee/naffouri/publications.html>



*Remark*

Let  $\mathcal{S}_j^{(1)}$  and  $\mathcal{S}_j^{(2)}$  be two different support vectors at the  $j$ th sparsity level. Note from (14) that if  $\nu(\mathcal{S}_j^{(1)}) > \nu(\mathcal{S}_j^{(2)})$  for some  $\sigma_n^2$  and  $p$ , then that inequality remains valid regardless of how  $\sigma_n^2$  and  $p$  change. In other words, the selection of dominant supports at each sparsity level is independent of these quantities. This observation helps the algorithm to bootstrap itself and to estimate the unknown sparsity rate and noise variance, as demonstrated ahead.

#### IV. EFFICIENT COMPUTATION OF THE DOMINANT SUPPORT SELECTION METRIC

As explained in Section III,  $\nu(\mathcal{S})$  requires extensive computation to determine the dominant supports. The computational complexity of the proposed algorithm is therefore largely dependent upon the way  $\nu(\mathcal{S})$  is computed. In this section, a computationally efficient procedure to calculate this quantity is presented.

We note that the efficient computation of  $\nu(\mathcal{S})$  depends mainly on the efficient computation of the term  $\xi_{\mathcal{S}} = \|(\Phi_{\mathcal{S}}(\Phi_{\mathcal{S}}^H \Phi_{\mathcal{S}})^{-1} \Phi_{\mathcal{S}}^H \mathbf{y})\|^2 = \|\Phi_{\mathcal{S}} \mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}]\|^2$ . Our focus is therefore on computing  $\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}]$  efficiently.

Consider the general support  $\mathcal{S} = \{s_1, s_2, s_3, \dots, s_k\}$  with  $s_1 < s_2 < \dots < s_k$  and let  $\underline{\mathcal{S}}$  and  $\overline{\mathcal{S}}$  denote the following subset and superset, respectively:

$$\underline{\mathcal{S}} = \{s_1, s_2, s_3, \dots, s_{k-1}\} \quad \overline{\mathcal{S}} = \{s_1, s_2, s_3, \dots, s_{k+1}\} \quad (15)$$

where  $s_k < s_{k+1}$ . In the following, we demonstrate how to update  $\mathbf{e}_{\mathbf{y}, k-1}(\underline{\mathcal{S}}) \triangleq \mathbb{E}[\mathbf{x}_{\underline{\mathcal{S}}}|\mathbf{y}]$  to obtain<sup>4</sup>  $\mathbf{e}_{\mathbf{y}, k}(\mathcal{S}) = \mathbb{E}[\mathbf{x}_{\mathcal{S}}|\mathbf{y}]$ . Here, we use  $\mathcal{S}$  to designate the supports and thus  $\mathbb{E}[\mathbf{x}_{\mathcal{S}}|\mathbf{y}]$  refers to  $\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}]$ . We note that

$$\mathbf{e}_{\mathbf{y}, k}(\mathcal{S}) = (\Phi_{\mathcal{S}}^H \Phi_{\mathcal{S}})^{-1} \Phi_{\mathcal{S}}^H \mathbf{y} = \left( \begin{array}{c} \left[ \Phi_{\underline{\mathcal{S}}}^H \right] \\ \left[ \Phi_{\underline{\mathcal{S}}} \phi_{s_k} \right] \\ \left[ \phi_{s_k}^H \right] \end{array} \right)^{-1} \begin{array}{c} \left[ \Phi_{\underline{\mathcal{S}}}^H \mathbf{y} \right] \\ \left[ \phi_{s_k}^H \mathbf{y} \right] \end{array} \quad (16)$$

By using the block inversion formula to express the inverse of the above and simplifying, we get

<sup>4</sup>We explicitly indicate the size  $k$  of  $\mathcal{S}$  in this notation as it elucidates the recursive nature of the developed algorithms.

$$\mathbf{e}_{\mathbf{y},k}(\mathcal{S}) = \begin{bmatrix} \frac{1}{f_{\mathcal{S}}} \left( \mathbf{q}_{\phi,k}^{\mathbf{H}}(\mathcal{S}) \mathbf{e}_{\mathbf{y},k-1}(\underline{\mathcal{S}}) - \mathbf{e}_{\mathbf{y},1}(s_k) \right) \mathbf{e}_{\phi,k}(\mathcal{S}) + \mathbf{e}_{\mathbf{y},k-1}(\underline{\mathcal{S}}) \\ \frac{-1}{f_{\mathcal{S}}} \left( \mathbf{q}_{\phi,k}^{\mathbf{H}}(\mathcal{S}) \mathbf{e}_{\mathbf{y},k-1}(\underline{\mathcal{S}}) - \mathbf{e}_{\mathbf{y},1}(s_k) \right) \end{bmatrix} \quad (17)$$

This recursion is initialized by  $\mathbf{e}_{\mathbf{y},1}(i) = (\phi_s^{\mathbf{H}} \phi_s)^{-1} \phi_s^{\mathbf{H}} \mathbf{y}$  for  $i = 1, 2, \dots, N$ . The recursion also depends on  $\mathbf{q}_{\phi,k}(\mathcal{S}) \triangleq \Phi_{\underline{\mathcal{S}}}^{\mathbf{H}} \phi_{s_k}$ ,  $\mathbf{e}_{\phi,k}(\mathcal{S}) \triangleq (\Phi_{\underline{\mathcal{S}}}^{\mathbf{H}} \Phi_{\underline{\mathcal{S}}})^{-1} \Phi_{\underline{\mathcal{S}}}^{\mathbf{H}} \phi_{s_k}$  and  $f_{\mathcal{S}} \triangleq 1 - \mathbf{q}_{\phi,k}^{\mathbf{H}}(\mathcal{S}) \mathbf{e}_{\phi,k}(\mathcal{S})$ . The recursions for  $\mathbf{q}_{\phi,k}(\mathcal{S})$ , and  $\mathbf{e}_{\phi,k}(\mathcal{S})$  may be determined in a similar fashion as

$$\mathbf{e}_{\phi,k+1}(\bar{\mathcal{S}}) = \begin{bmatrix} \frac{1}{f_{\bar{\mathcal{S}}}} \left( \mathbf{q}_{\phi,k}^{\mathbf{H}}(\mathcal{S}) \mathbf{e}_{\phi,k}(\underline{\mathcal{S}}; s_{k+1}) - \mathbf{e}_{\phi,2}(s_k; s_{k+1}) \right) \mathbf{e}_{\phi,k}(\mathcal{S}) + \mathbf{e}_{\phi,k}(\underline{\mathcal{S}}; s_{k+1}) \\ \frac{-1}{f_{\bar{\mathcal{S}}}} \left( \mathbf{q}_{\phi,k}^{\mathbf{H}}(\mathcal{S}) \mathbf{e}_{\phi,k}(\underline{\mathcal{S}}; s_{k+1}) - \mathbf{e}_{\phi,2}(s_k; s_{k+1}) \right) \end{bmatrix} \quad (18)$$

and<sup>5</sup>

$$\mathbf{q}_{\phi,k+1}(\bar{\mathcal{S}}) = \begin{bmatrix} \Phi_{\underline{\mathcal{S}}}^{\mathbf{H}} \\ \phi_{s_k}^{\mathbf{H}} \end{bmatrix} \phi_{s_{k+1}} = \begin{bmatrix} \mathbf{q}_{\phi,k}(\underline{\mathcal{S}}; s_{k+1}) \\ \mathbf{q}_{\phi,2}(s_k; s_{k+1}) \end{bmatrix} \quad (19)$$

The two recursions (18) and (19) start at  $k = 2$  and are thus initialized by  $\mathbf{e}_{\phi,2}(s_1; s_2)$  and  $\mathbf{q}_{\phi,2}(s_1; s_2)$  for  $s_1, s_2 = 1, 2, \dots, N$ . This completes the recursion of  $\mathbf{e}_{\mathbf{y},k}(\mathcal{S})$  which we utilize for recursive evaluation of  $\nu(\mathcal{S})$  as

$$\nu_k(\mathcal{S}) = \frac{1}{\sigma_{\mathbf{n}}^2} \|\Phi_{\mathcal{S}} \mathbf{e}_{\mathbf{y},k}(\mathcal{S})\|^2 - \frac{1}{\sigma_{\mathbf{n}}^2} \|\mathbf{y}\|^2 + |\mathcal{S}| \ln p + (N - |\mathcal{S}|) \ln(1 - p) \quad (20)$$

## V. ESTIMATION OF THE HYPERPARAMETERS $p$ AND $\sigma_{\mathbf{n}}^2$

One of the advantages of the proposed nGpFBMP is that it is agnostic with regard to signal statistics; the only parameters required are the noise variance,  $\sigma_{\mathbf{n}}^2$ , and the sparsity rate,  $p$ . In Section III, we pointed out that the dominant support selection process is independent of parameters  $p$  and  $\sigma_{\mathbf{n}}^2$ . We are therefore able to determine the supports irrespective of the initial estimates of these parameters. The independence of the dominant support selection process from  $p$  and  $\sigma_{\mathbf{n}}^2$  allows accurate and rapid estimation of these parameters. We note that although  $p$  and  $\sigma_{\mathbf{n}}^2$  are not needed for support calculations, they are required for computing the posteriors used in the calculation of  $\hat{\mathbf{x}}_{ammse}$  in (13). To determine these estimates, we might opt for finding the maximum-likelihood (ML) or maximum *a posteriori* (MAP) estimates using the

<sup>5</sup>Notation such as  $\mathbf{e}_{\phi,k}(\underline{\mathcal{S}}; s_{k+1})$  is a short hand for  $\mathbf{e}_{\phi,k}(\underline{\mathcal{S}} \cup \{s_{k+1}\})$ .

expectation maximization (EM) algorithm. However, this will add to the computational complexity and is unnecessary as a fairly accurate estimation could be performed in a very simple manner as follows.

The maximum *a posteriori* (MAP) estimate of support  $\mathcal{S}$  is given by  $\hat{\mathcal{S}}_{map} = \arg \max_{\mathcal{S}} p(\mathcal{S}|\mathbf{y})$ . We use this to get the MAP estimate of  $\mathbf{x}$ , i.e.,  $\hat{\mathbf{x}}_{map} = \mathbb{E}[\mathbf{x}|\mathbf{y}, \hat{\mathcal{S}}_{map}]$ . This  $\hat{\mathbf{x}}_{map}$  is in turn used to estimate  $p$ , iteratively, as follows:

$$\hat{p}^{(i+1)} = \left\| \hat{\mathbf{x}}_{map}^{(i)} \right\|_0 / N \quad (21)$$

Here, the superscripts refer to a particular iteration. The estimate is computed iteratively where in the first iteration of nGpFBMP,  $\hat{p}^{(1)}$  is initialized by  $p_{init}$ , the given initial estimate, to compute  $\hat{\mathbf{x}}_{map}^{(1)}$ . This is used to find the new estimate,  $\hat{p}^{(2)}$ , using (21) which is then supplied to nGpFBMP in the next iteration to compute  $\hat{\mathbf{x}}_{map}^{(2)}$ . This process is repeated until the estimate of  $p$  changes by less than 2% or until a predetermined maximum number of iterations has been performed. Simulation results show that, in most cases,  $\hat{p}$  converges rapidly. At this stage, the estimate of the noise variance can be computed as follows:

$$\hat{\sigma}_n^2 = \text{var}(\mathbf{y} - \Phi \hat{\mathbf{x}}_{map}) \quad (22)$$

We note that we do not need any iteration for estimating the noise variance. We use the  $\hat{\mathbf{x}}_{map}$  corresponding to the final estimate,  $\hat{p}$ , to find  $\hat{\sigma}_n^2$ .

## VI. RESULTS

To demonstrate the performance of the proposed nGpFBMP, we compare it here with Fast Bayesian Matching Pursuit (FBMP) [9] and the convex relaxation-based ( $l_1$ ) approach. The reason FBMP was selected is that it was shown to outperform a number of the contemporary sparse signal recovery algorithms, including OMP [13], StOMP [15], GPSR-Basic [16], and BCS [18]. Comparison with FBMP shows that nGpFBMP performs where FBMP fails for various signal settings which are discussed in detail in the following.

### *Signal Setup*

Experiments were conducted for signals drawn from Gaussian as well as non-Gaussian distributions. The following signal configurations were used for the experiments:

- 1) Gaussian (i.i.d. ( $\mu_{\mathbf{x}} = 10$ ,  $\sigma_{\mathbf{x}}^2 = 2$ )).
- 2) Non-Gaussian (Uniform, non-i.i.d. ( $5 \leq \mu_{\mathbf{x}} \leq 10$ ,  $1 \leq \sigma_{\mathbf{x}}^2 \leq 2$ )).
- 3) Unknown distribution (for this example, different images with unknown statistics were used),

where  $\mu_{\mathbf{x}}$  and  $\sigma_{\mathbf{x}}^2$  refer to the mean and variance of the corresponding distributions, respectively.

Entries of  $M \times N$  sensing/measurement matrix  $\Phi$  were i.i.d., with zero means and complex Gaussian distribution where the columns were normalized to the unit norm. The size of  $\Phi$  selected for the experiments was  $M = 256, N = 1024$ . The noise had a zero mean and was white and Gaussian,  $\mathcal{CN}(\mathbf{0}, \sigma_{\mathbf{n}}^2 \mathbf{I}_M)$ , with  $\sigma_{\mathbf{n}}^2$  determined according to the desired signal-to-noise ratio (SNR). Initial estimates of the hyperparameters used for the simulations were  $\mu_{\mathbf{x} \text{ est}} = 0$ ,  $\sigma_{\mathbf{x} \text{ est}}^2 = \frac{1}{10} \times \sigma_{\mathbf{x}}^2$ ,  $\sigma_{\mathbf{n} \text{ est}}^2 = 10 \times \sigma_{\mathbf{n}}^2$ , and  $p_{\text{est}} = 0.003$ , where estimates of the signal mean and variance were needed for FBMP.

In all of the experiments, parameter refinement was performed for both nGpFBMP and FBMP. For FBMP, the surrogate EM method proposed by its authors was used to refine the hyperparameters. The refinements were allowed to perform for a maximum of  $E_{max} = 10$  iterations. For fairness, support and amplitude refinement [3] procedures were performed on the results of the CS algorithm<sup>6</sup>. Finally, the normalized mean-squared error (NMSE) between the original signal,  $\mathbf{x}$ , and its MMSE estimate,  $\hat{\mathbf{x}}_{ammse}$ , was used as the performance measure:

$$\text{NMSE} = 10 \log_{10} \left( \frac{1}{K} \sum_{k=1}^K \frac{\|\hat{\mathbf{x}}_k - \mathbf{x}_k\|^2}{\|\mathbf{x}_k\|^2} \right) \quad (23)$$

where  $K$  was the number of trials performed to compute NMSE between  $\mathbf{x}$  and its estimate.

#### *Experiment 1 (Signal estimation performance comparison for varying SNR)*

In the first set of experiments, NMSEs were measured for values of SNR between 0 dB and 30 dB and plotted to compare the performance of nGpFBMP with FBMP and the CS algorithm. The signal sparsity rate selected for these experiments was  $p = 0.005$ .

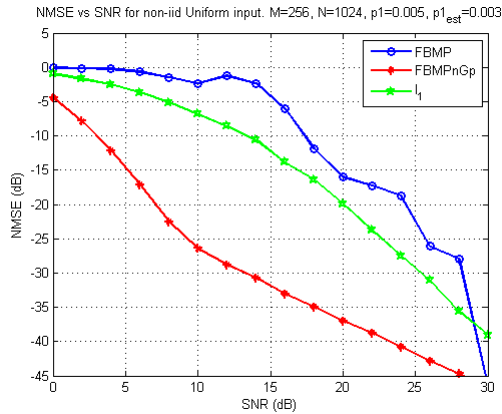
Figs. 2a and 2b show that the proposed method has better NMSE performance than both FBMP and CS for all considered signals. Only at very high values of SNR does the NMSE performance of FBMP approach that of nGpFBMP.

#### *Experiment 2 (Signal estimation performance comparison for varying sparsity parameter $p$ )*

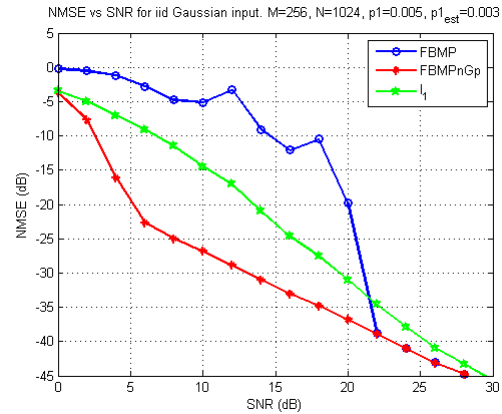
In a similar set of experiments, NMSE and mean runtime were measured for different values of sparsity parameter  $p$ . The value of SNR selected for these experiments was 20 dB.

Figs. 3 and 4 demonstrate the superiority of nGpFBMP over FBMP and CS. Runtime graphs of Figs. 3 and 4 depict that the runtime of nGpFBMP increases for higher values of  $p$ . This occurs because the

<sup>6</sup>Actual parameter values were provided to the CS algorithm instead of estimates; furthermore, support and amplitude refinement was also performed to demonstrate that, despite these measures, its performance was inferior to that of nGpFBMP.



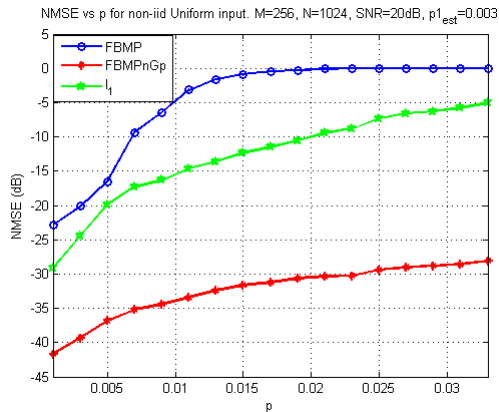
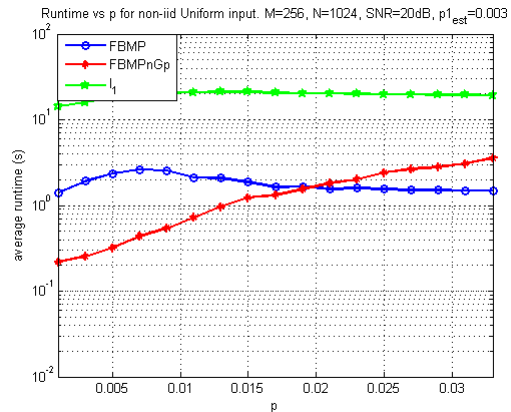
(a) NMSE vs SNR for uniform non-i.i.d. input



(b) NMSE vs SNR for Gaussian i.i.d. input

Fig. 2: NMSE vs SNR graphs for uniform non-i.i.d. and Gaussian i.i.d. inputs

initial estimate of  $p$  was 0.003, and as the sparsity rate of  $\mathbf{x}$  increased, more iterations were required to estimate the value of  $p$ . With higher values of  $p$ , the difference in runtime is insignificant given the excellent NMSE performance of our method. We also observe that performance of nGpFBMP is relatively insensitive to changes in  $p$  as the corresponding changes in NMSE are very small, thus demonstrating the strength of the proposed algorithm.

(a) NMSE vs  $p$ (b) Runtime vs  $p$ Fig. 3: NMSE and average runtime vs  $p$  graphs for uniform non-i.i.d. input

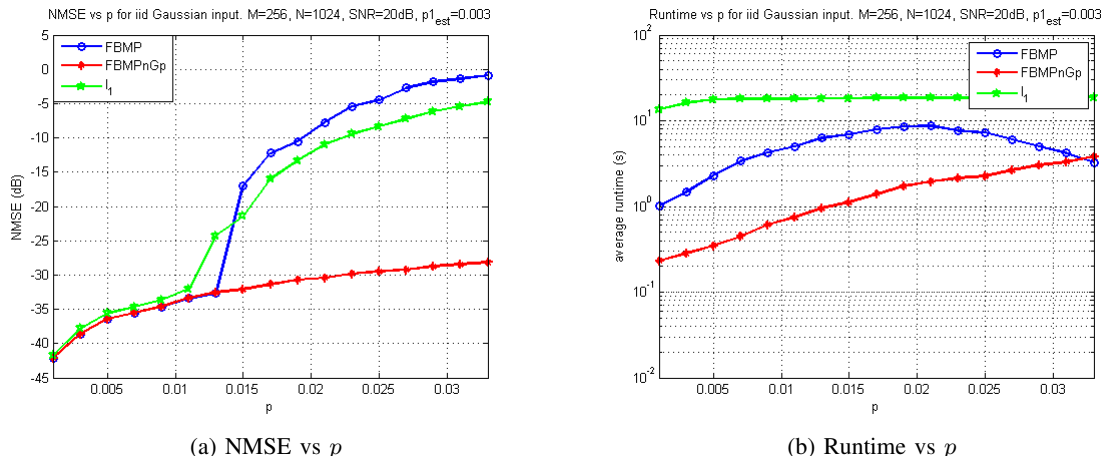


Fig. 4: NMSE and average runtime vs  $p$  graphs for Gaussian i.i.d. input

TABLE II: Average NMSE (dB) comparison between FBMP and nGpFBMP when the initial estimates of the hyperparameters are close to the actual values

Signal type	FBMP	nGpFBMP
Gaussian	-20.55	-31.103
Uniform (i.i.d.)	-24.2	-30.98
Uniform (non-i.i.d.)	-23.87	-30

*Experiment 3 (Comparison of signal estimates when the initial statistics of signal and noise are very close to the actual values)*

Table II compares the average NMSEs of FBMP and nGpFBMP for different types of signals when the initial estimates ( $\mu_x$ ,  $\sigma_x^2$ , and  $\sigma_n^2$ ) were chosen to be very near to their actual values. Since nGpFBMP is independent of these initial estimates its performance did not change. On the other hand, performance of FBMP improved, although it did not outperform nGpFBMP. The results show the robustness of our algorithm as it is not dependent on the initial estimates of  $\mu_x$ ,  $\sigma_x^2$ , and  $\sigma_n^2$ . We note that each value in Table II has been computed by averaging the results of 500 independent experiments.

*Experiment 4 (Comparison of multiscale image recovery performance)*

In another experiment, we carried out multiscale recovery of different images that were  $128 \times 128$  pixels. These images are shown in the first row of Fig. 5. One-level Haar wavelet decomposition of these images was performed, resulting in one ‘approximation’ (low-frequency) and three ‘detail’ (high-

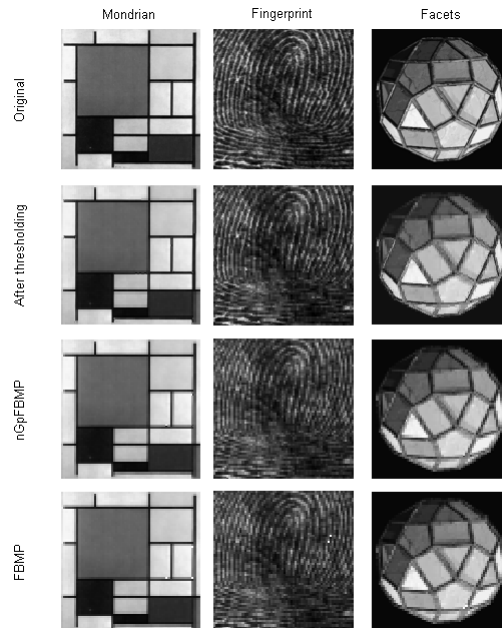


Fig. 5: Row 1: Original images used in the multiscale recovery experiment. Row 2: Reconstructed images (inverse wavelet transform) once the wavelet-decomposed detail components were subjected to the threshold. Rows 3 & 4: Reconstructed from random measurements using nGpFBMP and FBMP respectively.

frequency) images. Unlike the approximation component, the detail components are compressible in nature. We, therefore, generated their sparse versions by applying a suitable threshold; the noisy random measurements were acquired later from the threshold versions. The detail components were reconstructed from these measurements through nGpFBMP. Finally, inverse wavelet transform was applied to reconstruct the images from the recovered details and the original approximations. Reconstruction errors and times were recorded and, for comparison, recoveries were obtained using FBMP. The resulting reconstructed images are shown in Fig. 5. Numerical details of the results for these experiments are given in Table III, from which it is obvious that images reconstructed using nGpFBMP have lower NMSEs and require a significantly shorter reconstruction time than does FBMP.

## VII. CONCLUSION

In this paper, we presented a robust Bayesian matching pursuit algorithm based on a fast recursive method. Compared with other robust algorithms, our algorithm does not require signals to be derived from some known distribution. This is useful when we can not estimate the parameters of the signal distributions. Application of the proposed method on several different signal types demonstrated its

TABLE III: NMSE and Reconstruction time comparisons between FBMP and nGpFBMP for the test images shown in Fig. 5

Image	FBMP		nGpFBMP	
	NMSE (dB)	Time (s)	NMSE (dB)	Time (s)
Mondrian	-16.68	19.36	-18.84	6.8
Fingerprint	-9.97	19.24	-13.92	6.75
Facets	-14.75	19.56	-19.30	7.01

superiority and robustness.

#### REFERENCES

- [1] K. Jia, X. Wang, and X. Tang, "Optical flow estimation using learned sparse model," in *IEEE Int. Conf. on Comp. Vision (ICCV)*, Nov. 2011, pp. 2391–2398.
- [2] M. Lustig, D. Donoho, J. Santos, and J. Pauly, "Compressed sensing MRI," *IEEE Signal Processing Mag.*, vol. 25, no. 2, pp. 72–82, Mar. 2008.
- [3] T. Al-Naffouri, A. Quadeer, F. Al-Shaalan, and H. Hmida, "Impulsive noise estimation and cancellation in DSL using compressive sampling," in *IEEE Int. Symp. on Circuits and Syst. (ISCAS)*, May 2011, pp. 2133–2136.
- [4] M. Firooz and S. Roy, "Network tomography via compressed sensing," in *IEEE Global Telecommun. Conf. (GLOBECOM 2010)*, Dec. 2010, pp. 1–5.
- [5] T. Y. Al-Naffouri, E. B. Al-Safadi, and M. E. Eltayeb, "OFDM peak-to-average power ratio reduction method," U.S. Patent 2011/0 122 930, May, 2011.
- [6] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *J. of the Roy. Statistical Soc. Series B-Methodological*, vol. 58, no. 1, pp. 267–288, 1996.
- [7] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.
- [8] A. Quadeer, S. Ahmed, and T. Al-Naffouri, "Structure based bayesian sparse reconstruction using non-gaussian prior," in *49th Annu. Allerton Conf. on Commun., Control, and Computing (Allerton)*, Sep. 2011, pp. 277–283.
- [9] P. Schniter, L. C. Potter, and J. Ziniel, "Fast Bayesian Matching Pursuit: Model Uncertainty and Parameter Estimation for Sparse Linear Models," *IEEE Trans. Signal Process.*, to be published. [Online]. Available: [http://www.ece.osu.edu/~zinielj/FBMP\\_TransSP.pdf](http://www.ece.osu.edu/~zinielj/FBMP_TransSP.pdf)
- [10] E. Candes and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, vol. 23, no. 3, pp. 969–985, Jun. 2007.
- [11] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, pp. 489–509, Feb. 2006.
- [12] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, pp. 1289–1306, Apr. 2006.
- [13] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, pp. 4655–4666, Dec. 2007.



- [14] J. Haupt and R. Nowak, "Signal reconstruction from noisy random projections," *IEEE Trans. Inf. Theory*, vol. 52, pp. 4036–4048, Sep. 2006.
- [15] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit," *IEEE Trans. Inf. Theory*, vol. 58, pp. 1094–1121, Feb. 2012.
- [16] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems," *IEEE J. Sel. Topics Signal Process.*, vol. 1, pp. 586–597, Dec. 2007.
- [17] M. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. of Mach. Learning Research*, vol. 1, no. 3, pp. 211–244, SUM 2001.
- [18] S. Ji and L. Carin, "Bayesian compressive sensing and projection optimization," in *Proc. of the 24th Int. Conf. on Mach. Learning*, 2007, pp. 377–384.
- [19] C. M. Bishop and M. E. Tipping, "Variational relevance vector machines," in *Proc. of the 16th Conf. on Uncertainty in Artificial Intelligence*, 2000, pp. 46–53.
- [20] E. G. Larsson and Y. Selén, "Linear regression with a sparse parameter vector," *IEEE Trans. Signal Process.*, vol. 55, pp. 451–460, Feb 2007.

## APPENDIX

*Algorithm: Non-Gaussian prior fast Bayesian matching pursuit*

- 1) The following quantities are calculated one time and used in the following iterations
  - a) inner products of  $\mathbf{y}$  with columns of  $\Phi$ .
  - b) inner products of columns of  $\Phi$  with each other.
  - c)  $\mathbf{e}_{\mathbf{y},1}(i) = (\phi_i^H \phi_i)^{-1} \phi_i^H \mathbf{y}$  for  $i = 1, 2, \dots, N$  i.e.  $\mathbb{E}[\mathbf{x}|\mathbf{y}, \mathcal{S}]$  for each of the one-length support vector. Note that the components of these calculations are available from steps 1a and 1b.
- 2) Calculate  $N$  one-element metrics,  $\nu(\mathcal{S}_1)$ , using  $\mathbf{e}_{\mathbf{y},1}$  from the previous step for each of the possible support elements  $\alpha_i, i \in [1, N]$  as explained in Section III.
- 3) Perform the following search  $D$  times (each search has  $p = 1 : P$  stages)
  - a) At each  $p$ th stage:
    - i) Find the  $p$ -element metric,  $\nu(\mathcal{S}_p)$ , with the maximum value and note down the corresponding  $\alpha_i$ .
    - ii) If this  $\alpha_i$  has been selected in any of the previous  $p$ th stages, mark it and discard it and go to step 3(a)i again.
    - iii) else update  $\mathbf{e}_{\phi,p}$  using (18) for this newly selected position,  $\alpha_i$ , and add the corresponding  $\mathcal{S}_p$  as a new member of  $\mathcal{S}^d$ .
    - iv) Save  $\mathbf{e}_{\phi,p}$  to update its value in the next iteration.
    - v) Use  $\mathbf{e}_{\phi,p}$  and  $\mathbf{e}_{\mathbf{y},p}$  to compute the update  $\mathbf{e}_{\mathbf{y},p+1}$  (17) for all  $\mathcal{S}_{p+1} = [\mathcal{S}_p \ \alpha_i], \forall i \mid \alpha_i \notin \mathcal{S}_p$ .
    - vi) Compute the dominant support selection metrics for all these combinations  $\mathcal{S}_{p+1}$  using the computed  $\mathbf{e}_{\mathbf{y},p+1}$ . This yields  $(p+1)$ -element metrics  $\nu(\mathcal{S}_{p+1})$  to be used in the next iteration.
  - b) end  $p$
- 4)  $\mathcal{S}^d$  contains the dominant supports that will be used to find the sum in (13).