LETTER Communicated by Ken Kreutz-Delgado

Computing Sparse Representations of Multidimensional Signals Using Kronecker Bases

Cesar F. Caiafa
ccaiafa@gmail.com
Instituto Argentino de Radioastronomía (IAR), CCT La Plata-CONICET, 1894, Villa Elisa C.C. No. 5, Argentina, and Facultad de Ingeniería, Universidad de Buenos Aires, Buenos Aires C1063ACV, Argentina

Andrzej Cichocki
cia@brain.riken.jp
LABSP, RIKEN Brain Science Institute, Wako, Saitama 351-01989, Japan, and Research Systems Institute, PAS, Warsaw 01-447, Poland

Recently there has been great interest in sparse representations of signals under the assumption that signals (data sets) can be well approximated by a linear combination of few elements of a known basis (dictionary). Many algorithms have been developed to find such representations for one-dimensional signals (vectors), which requires finding the sparsest solution of an underdetermined linear system of algebraic equations. In this letter, we generalize the theory of sparse representations of vectors to multiway arrays (tensors)—signals with a multidimensional structure—by using the Tucker model. Thus, the problem is reduced to solving a large-scale underdetermined linear system of equations possessing a Kronecker structure, for which we have developed a greedy algorithm, Kronecker-OMP, as a generalization of the classical orthogonal matching pursuit (OMP) algorithm for vectors. We also introduce the concept of multiway block-sparse representation of *N*-way arrays and develop a new greedy algorithm that exploits not only the Kronecker structure but also block sparsity. This allows us to derive a very fast and memory-efficient algorithm called N-BOMP (*N*-way block OMP). We theoretically demonstrate that under the block-sparsity assumption, our N-BOMP algorithm not only has a considerably lower complexity but is also more precise than the classic OMP algorithm. Moreover, our algorithms can be used for very large-scale problems, which are intractable using standard approaches. We provide several simulations illustrating our results and comparing our algorithms to classical algorithms such as OMP and BP (basis pursuit) algorithms. We also apply the N-BOMP algorithm as a fast solution for the compressed sensing (CS) problem with large-scale data sets, in particular, for 2D compressive imaging (CI) and 3D hyperspectral CI, and we show examples with real-world multidimensional signals.

1 Introduction

A concept that underlies some recent developments in many fields such as in image and signal processing is sparsity. Usually signals living in a vector space do not cover the entire space uniformly (Lu & Do, 2008). In particular, it was discovered that most signals of interest can be well approximated by a sparse representation over a known dictionary by using a linear combination of few dictionary elements (atoms) (Donoho, 2006; Candès, Romberg, & Tao, 2006). Moreover, sometimes it is useful to consider overcomplete dictionaries, where the number of atoms is larger than the signal size (Elad, Figueiredo, & Ma, 2010). In fact, sparse representations are found in the way that visual cortex codes natural images in the brain (Olshausen & Field, 1996). The implications of the sparsity assumption have recently driven the development of many exciting applications in signal processing. For example, it was proved that signals with a sparse representation can be reconstructed from a reduced number of measurements, the main objective of compressed sensing (CS) (Donoho, 2006; Candès et al., 2006); some algorithms for blind source separation are based on the sparsity assumption (Bobin, Starck, Fadili, & Moudden, 2007; Li et al, 2004; Caiafa & Cichocki, 2009); also techniques for denoising and inpainting of images have been developed using similar ideas (Elad, Starck, Querre, & Donoho, 2005; Elad & Aharon, 2006). In this framework, the main problem is how to solve an underdetermined system of linear algebraic equations constrained to the fact that among the infinite number of solutions, the sparsest solution may be unique. To this end, many algorithms were proposed, including greedy methods such as matching pursuit (MP) (Davis, Mallat, & Zhang, 1994), orthogonal matching pursuit (OMP) (Tropp, 2004), compressive sampling matching pursuit (CoSaMP) (Needell & Tropp, 2009), ℓ_1 norm minimization methods such as basis pursuit (Chen & Donoho, 2001), gradient projection sparse reconstruction (GPSR) (Figueiredo, Nowak, & Wright, 2007), and many others (see Tropp & Wright, 2010, for an up-to-date review of algorithms).

Another characteristic of signals found in modern applications is that they often have a multidimensional structure, where each dimension (mode) has a particular physical meaning (e.g., space, time, frequency, trials). For example, a 3D image produced by a computed tomography (CT) system or a magnetic resonance imaging (MRI) system corresponds to a sampled version of a 3D function $f(x_1, x_2, x_3)$. In this case, the multidimensional image is stored in memory as a multiway array (tensor) $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ whose elements are samples taken on a grid, that is, $y_{i_1 i_2 i_3} = f(i_1 h, i_2 h, i_3 h)$ ($i_n = 1, 2, \ldots, I_n$, $n = 1, 2, 3$), with h being the discretization step among all dimensions. Multiway arrays are the generalization of vectors and matrices to a higher number of dimensions and are usually referred in the literature as N-way arrays or tensors. They are very attractive mathematical tools possessing their own properties (Kolda & Bader, 2009; Cichocki, Zdunek, Phan, & Amari, 2009).

Due to the availability of massive data sets occurring in new applications and requirements in scientific computation, a great deal of effort has been devoted to the development of new algorithms for multiway structured data sets, sometimes with a high number of dimensions (Oseledets, Savostianov, & Tyrtyshnikov, 2008; Caiafa & Cichocki, 2010; Oseledets, 2011). The curse of dimensionality problem makes the task of finding sparse representations of *N*-way arrays very expensive in terms of memory storage resources and computation load since the number of entries grows exponentially with the number of dimensions. To solve this problem, one may try to find some structure in data sets and build approximate models using fewer parameters than the number of entries, as is the case of the Tucker model (Lathauwer, Moor, & Vandewalle, 2000b; Kolda & Bader, 2009). In fact, multidimensional signals often reveal a structure in each mode that allows one to adopt good approximate representations based on the Kronecker product of dictionaries associated with each of the modes (Rivenson & Stern, 2009a). For example, most popular transforms applied to two-dimensional signals (2D images) are based on the application of a transformation of rows followed by a transformation of columns known as separable transforms. In this case, the dictionary associated with the vectorized image can be written as the Kronecker product of dictionaries associated with rows and columns. The Kronecker structure has been extensively used in the image processing community (Nagy & Kilmer, 2006; Rivenson & Stern, 2009a). Also the Kronecker structure has been been proposed in the matrix analysis community as a good preconditioner to solve linear systems (Loan & Pitsianis, 1992) and to approximate function-related linear systems (Ford & Tyrtyshnikov, 2003; Tyrtyshnikov, 2004).

This letter is organized as follows. In section 2, the basic notation is introduced, and some important previous results are presented; in section 3, we generalize the theory of sparse representations of vectors to multiway arrays and introduce the concept of block sparsity; in section 4, we introduce the multidimensional compressed sensing problem; in section 5, our new greedy algorithms are developed for the case of multidimensional sparsity (Kronecker-OMP algorithm) and multiway block sparsity (N-BOMP), including a detailed analysis of their computational complexities; in section 6, we present a new theoretical result about the performance guarantee of N-BOMP algorithm, which we show to be much less restrictive than the case of the classical OMP algorithm; in section 7, extensive simulations are presented using synthetically generated *N*-way arrays, as well as the application to CS of real-world multidimensional signals. In section 8, we outline the main conclusions of this work.

2 Notation and Preliminaries

In this letter, N-way arrays (multidimensional signals) are denoted by underlined boldface capital letters; matrices (two-way arrays) are denoted by bold uppercase letters and vectors by boldface lowercase letters, that is, $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$; and $\mathbf{D} \in \mathbb{R}^{I \times M}$ and $\mathbf{y} \in \mathbb{R}^{I}$ are examples of an N-way array, a matrix, and a vector, respectively. The ith entry of a vector $\mathbf{y}$ is denoted by y_i, and the element (i, j) of a matrix $\mathbf{Y}$ is denoted by either of the following ways $\mathbf{Y}(i, j) = y_{ij}$. The same notation is used for N-way arrays by referring to the element $(i_1, i_2, \ldots, i_N)$ as $\underline{\mathbf{Y}}(i_1, i_2, \ldots, i_N) = y_{i_1 i_2 \cdots i_N}$. As a natural generalization, we define the Frobenius norm of an N-way array by $||\underline{\mathbf{Y}}||_F = \sqrt{\sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} y^2_{i_1 i_2 \cdots i_N}}$. Sometimes we use MATLAB notation to indicate the full range of indices in certain modes; for example, $\mathbf{Y}(:, j)$ means a vector composed of all the entries of column j. Additionally, we refer to a subarray (or block) by restricting the indices to belonging to certain subsets of indices, for example, given the following subsets of S_n indices $\mathcal{I}_n = \{i_n^1, i_n^2, \ldots, i_n^{S_n}\}$ in each mode $n = 1, 2, \ldots, N$, the subarray $\underline{\mathbf{Y}}(\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_N) \in \mathbb{R}^{S_1 \times S_2 \times \cdots \times S_N}$ is obtained by keeping the entries of the original N-way array $\underline{\mathbf{Y}}$ at the selected index subsets $\mathcal{I}_n$ $(n = 1, 2, \ldots, N)$. We denote the cardinality of a subset of indices $\mathcal{I}$ by $|\mathcal{I}|$.

2.1 Sparse Solutions of Underdetermined Linear Systems. We say that a one-dimensional signal (vector) has a sparse representation in a known dictionary if it can be recovered exactly by a linear combination of few selected elements of the dictionary (atoms) as the following definition states:

Definition 1 *(sparse representation of vectors). A signal $\boldsymbol{y} \in \mathbb{R}^I$ has a K-sparse representation with respect to the dictionary $\boldsymbol{D} \in \mathbb{R}^{I \times M}$ if the following relation holds:*

$$\boldsymbol{y} = \boldsymbol{D}\boldsymbol{x}, \ \text{with} \ ||\boldsymbol{x}||_0 \leq K \tag{2.1}$$

where $||\boldsymbol{x}||_0$ is the ℓ_0 quasi-norm[1] *of the vector $\mathbf{x} \in \mathbb{R}^M$ obtained by counting the number of nonzero entries and typically $M \geq I$, $K \ll M$. In other words, there is a small set of K indices $\mathcal{I} = [i^1, i^2, \ldots, i^K]$ such that $x_i = 0$ if $i \notin \mathcal{I}$.*

Since the dictionary $\mathbf{D}$ has more columns than rows, the solution $\mathbf{x}$ cannot be uniquely recovered; however, the sparsest solution may be unique. For instance, it is known that if no $2K$ columns of $\mathbf{D}$ are linearly dependent, then the solution of equation 2.1, if it exists, is unique (Donoho & Elad, 2003). Many other conditions on matrix $\mathbf{D}$ have been provided in order to guarantee the uniqueness of a sparse representation (Donoho & Elad, 2003;

[1] Note that ℓ_0 norm is not a norm formally since $||\alpha\mathbf{x}||_0 \neq \alpha||\mathbf{x}||_0$.

Tropp, 2004). In particular, for a matrix with unit-norm columns ($||\mathbf{d}_i||_2^2 = 1$, $i = 1, 2, \ldots, M$), its "coherence" $\mu(\mathbf{D})$, which is defined as the maximum absolute value of the correlation between two columns,

$$\mu(\mathbf{D}) = \max_{i \neq j} |\mathbf{d}_i^T \mathbf{d}_j|, \tag{2.2}$$

gives us a characterization of the system. More specifically, if

$$K < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right), \tag{2.3}$$

then the solution of equation 2.1 is unique (Tropp, 2004).

Several algorithms have been proposed to find the sparsest representation of a signal for a given dictionary by solving the corresponding underdetermined linear system of algebraic equations under the sparsity constraint (Tropp & Wright, 2010). These methods can be divided in two main groups: basis pursuit (BP) and matching pursuit (MP).

In BP, the combinatorial problem is replaced with a convex optimization problem; basically, the ℓ_1 norm $||\mathbf{x}||_1$ is minimized subject to the constraint $\mathbf{y} = \mathbf{D}\mathbf{x}$, which can be proved to solve our original problem if the vector is sparse enough and the matrix $\mathbf{D}$ has sufficiently low coherence (Donoho & Elad, 2003; Tropp, 2004). However, it is well known that BP is computationally expensive and unsuitable for large-scale problems (Tropp, 2004). On the other side, MP methods, which are also known as greedy algorithms (Davis et al., 1994; Tropp, 2004), have complexity $\mathcal{O}(KMI)$, which is significantly smaller compared to BP, especially when the signal sparsity level K is low (Tropp & Gilbert, 2007). A powerful standard greedy algorithm is the orthogonal matching pursuit (OMP) which was adapted and studied in Tropp (2004), Needell, Tropp, and Vershynin (2008), and Needell and Tropp (2009). The OMP pseudocode is reproduced in algorithm 1. OMP iteratively refines a sparse solution by successively identifying one component at a time that yields the greatest improvement in quality until a desired sparsity level K is reached or the approximation error is below some predetermined level ϵ. An optimized implementation of this algorithm is obtained by using the Cholesky factorization for the computation of STEP 5 (Rubinstein, Zibulevsky, & Elad, 2008).

Besides its simplicity, a nice property of the OMP algorithm is that if the dictionary $\mathbf{D}$ satisfies condition 2.3 and a K-sparse solution exists, then the OMP algorithm will obtain it after exactly K iterations (Tropp, 2004).

2.2 Kronecker Dictionaries. The Kronecker product is defined as follows:

Algorithm 1: Orthogonal Matching Pursuit (Tropp, 2004)

```
Require: Dictionary $\mathbf{D} \in \mathbb{R}^{I \times M}$, signal $\mathbf{y} \in \mathbb{R}^{I}$, sparsity $K$, tolerance $\epsilon$
Ensure: Sparse representation $\mathbf{y} \approx \mathbf{D}\mathbf{x}$, $\mathbf{x} \in \mathbb{R}^{M}$, with $||\mathbf{x}||_0 \leq K$ (nonzero
    entries are given by $\mathbf{x}(\mathcal{I}) = \mathbf{a}$)
$\mathcal{I} = [\emptyset]$, $\mathbf{r} = \mathbf{y}$, $\mathbf{x} = \mathbf{0}$, $k = 1$;
while $k \leq K$ and $||\mathbf{r}|| > \epsilon$ do
    $i^k = \arg\max_i |\mathbf{d}_i^T \mathbf{r}|$; (to find the maximum correlated atom with
    respect to the residual)
    $\mathcal{I} = [\mathcal{I}, i^k]$; (to incorporate the selected index)
    $\mathbf{a} = \arg\min_u ||\mathbf{D}(:, \mathcal{I})\mathbf{u} - \mathbf{y}||_2^2$; (nonzero entries are obtained by
    solving an LS problem)
    $\mathbf{r} = \mathbf{y} - \mathbf{D}(:, \mathcal{I})\mathbf{a}$; (to update the residual)
    $k = k + 1$;
end while
return $\mathcal{I}$,$\mathbf{a}$;
```

Definition 2 *(Kronecker product). Given two matrices $\boldsymbol{A} \in \mathbb{R}^{I \times M}$ and $\boldsymbol{B} \in \mathbb{R}^{J \times N}$ their Kronecker product $\boldsymbol{A} \otimes \boldsymbol{B} \in \mathbb{R}^{IJ \times MN}$ is defined by*

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{pmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} & \cdots & a_{1M}\boldsymbol{B} \\ a_{21}\boldsymbol{B} & a_{22}\boldsymbol{B} & \cdots & a_{2M}\boldsymbol{B} \\ \cdot & \cdot & \cdot & \cdot \\ a_{I1}\boldsymbol{B} & a_{I2}\boldsymbol{B} & \cdots & a_{IM}\boldsymbol{B} \end{pmatrix}. \tag{2.4}$$

Kronecker dictionaries—those that can be written as a Kronecker product of elementary matrices—play a key role in higher-dimensional signal processing (Nagy & Kilmer, 2006; Duarte & Baraniuk, 2011; Rivenson & Stern, 2009a) and other fields (Loan & Pitsianis, 1992; Ford & Tyrtyshnikov, 2003). In order to introduce them here, let us consider the simplest case of a 2D image $\mathbf{Y} \in \mathbb{R}^{I_1 \times I_2}$ for which a separable transform can be applied as follows:[2]

$$\mathbf{X} = \mathbf{T}_1^T \mathbf{Y} \mathbf{T}_2, \tag{2.5}$$

with $\mathbf{T}_1 \in \mathbb{R}^{I_1 \times I_1}$ and $\mathbf{T}_2 \in \mathbb{R}^{I_2 \times I_2}$ being nonsingular matrices associated with the transforms of columns and rows, respectively, and $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$ is the matrix of coefficients. In other words, the original image $\mathbf{Y}$ can be recovered by applying the inverse transform,

$$\mathbf{Y} = (\mathbf{T}_1^{-1})^T \mathbf{X} \mathbf{T}_2^{-1}. \tag{2.6}$$

A basic property of the Kronecker product is that (Loan, 2000)

$$vec(\mathbf{B}\mathbf{X}\mathbf{A}^T) = (\mathbf{A} \otimes \mathbf{B}) vec(\mathbf{X}), \tag{2.7}$$

where the operation $\mathbf{x} = vec(\mathbf{X})$ converts the matrix $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$ into a vector $\mathbf{x} \in \mathbb{R}^{I_1 I_2}$ by stacking the columns of matrix $\mathbf{X}$. If we use property 2.7 in equation 2.6 and by defining $\mathbf{D}_1^T = \mathbf{T}_1^{-1}$, $\mathbf{D}_2^T = \mathbf{T}_2^{-1}$, $\mathbf{x} = vec(\mathbf{X})$ and $\mathbf{y} = vec(\mathbf{Y})$ we easily obtain

$$\mathbf{y} = (\mathbf{D}_2 \otimes \mathbf{D}_1)\mathbf{x}, \tag{2.8}$$

where we explicitly express the vector $\mathbf{y}$ as a linear combination of elements of a dictionary with Kronecker structure $\mathbf{D} = \mathbf{D}_2 \otimes \mathbf{D}_1$. If the coefficient matrix $\mathbf{X}$ has few nonzero representative entries, then the signal $\mathbf{Y}$ has an approximated sparse representation. We can generalize the definition of a Kronecker dictionary for any arbitrary number N of matrices, that is, $\mathbf{D} = \mathbf{D}_N \otimes \mathbf{D}_{N-1} \otimes \cdots \otimes \mathbf{D}_1$. It is known that the coherence of $\mathbf{D}$ (global coherence) satisfies the following equation (Jokar & Mehrmann, 2009):

$$\mu(\mathbf{D}) = \max\{\mu_1, \mu_2, \ldots, \mu_N\}, \tag{2.9}$$

where $\mu_n = \mu(\mathbf{D}_n)$, $(n = 1, 2, \ldots, N)$, that is, if one of the dictionaries $\mathbf{D}_n$ has a large coherence μ_n, then it will dominate the coherence of $\mathbf{D}$.

2.3 Multiway Arrays (N-way arrays) and the Tucker Model. Given a multiway array (tensor) $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$, its mode-$n$ vectors are obtained by fixing every index but the one in mode n. The mode-n unfolding matrix $\mathbf{Y}_{(n)} \in \mathbb{R}^{I_n \times I_1 I_2 \cdots I_{n-1} I_{n+1} \cdots I_N}$ is defined by arranging all the mode-n vectors as columns of a matrix. Note that for the 2D case, mode-1 and mode-2 vectors

[2] Classical examples of such separable transforms in the area of image processing are the discrete Fourier transform (DFT), the discrete cosine transform (DCT), discrete wavelet transform (DWT), and others.

are columns and rows, respectively, and the mode-2 unfolding matrix is the transposed matrix.

Given a multidimensional signal (N-way array) $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ and a matrix $\mathbf{A} \in \mathbb{R}^{J \times I_n}$ the mode-n tensor-by-matrix product $\underline{\mathbf{Z}} = \underline{\mathbf{Y}} \times_n \mathbf{A} \in \mathbb{R}^{I_1 \times I_2 \cdots I_{n-1} \times J \times I_{n+1} \cdots I_N}$ is defined by

$$z_{i_1 i_2 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} y_{i_1 i_2 \cdots i_N} a_{j i_n}, \tag{2.10}$$

with $i_k = 1, 2, \ldots, I_k$ ($k \neq n$) and $j = 1, 2, \ldots, J$.

The Tucker decomposition (Tucker, 1963; Lathauwer et al., 2000a) is a powerful compressed format that exploits the linear structure of the unfolding matrices of an N-way array. More specifically, when the ranks of these matrices are bounded by $\mathrm{rank}(\mathbf{Y}_{(n)}) \leq R_n \leq I_n$ ($n = 1, 2, \ldots, N$), then the following multilinear expression holds:

$$\underline{\mathbf{Y}} = \underline{\mathbf{G}} \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_2 \cdots \times_N \mathbf{A}_N, \tag{2.11}$$

with a core tensor $\underline{\mathbf{G}} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_n}$ and factor matrices $\mathbf{A}_n \in \mathbb{R}^{I_n \times R_n}$. It is easy to see that mode-n vectors of an N-way array with a Tucker representation belong to the span of the columns of matrix $\mathbf{A}_n$. In fact, it can be shown that equation 2.11 implies (Kolda & Bader, 2009)[3]

$$\mathbf{Y}_{(n)} = \mathbf{A}_n \mathbf{G}_{(n)} (\mathbf{A}_N \cdots \otimes \mathbf{A}_{n+1} \otimes \mathbf{A}_{n-1} \cdots \otimes \mathbf{A}_1)^T. \tag{2.12}$$

Let us define the vectorization operator on N-way arrays as $vec(\underline{\mathbf{Y}}) \equiv vec(\mathbf{Y}_{(1)}) \in \mathbb{R}^{I_1 I_2 \cdots I_N}$, that is, by stacking all the mode-1 vectors. There is a connection between the Tucker model and a Kronecker representation for multiway arrays, as the following proposition stablishes.

Proposition 1 *(relationship between the Tucker model and a Kronecker representation of an N-way array). Given* $\underline{\boldsymbol{Y}} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$, $\underline{\boldsymbol{X}} \in \mathbb{R}^{M_1 \times M_2 \cdots \times M_N}$, $\boldsymbol{D}_n \in \mathbb{R}^{I_n \times M_n}$ *(*$n = 1, 2, \ldots, N$*),* $\boldsymbol{x} = vec(\underline{\boldsymbol{X}})$ *and* $\boldsymbol{y} = vec(\underline{\boldsymbol{Y}})$*, the following two representations are equivalent:*

$$\underline{\boldsymbol{Y}} = \underline{\boldsymbol{X}} \times_1 \boldsymbol{D}_1 \times_2 \boldsymbol{D}_2 \cdots \times_N \boldsymbol{D}_N, \tag{2.13}$$

$$\boldsymbol{y} = (\boldsymbol{D}_N \otimes \boldsymbol{D}_{N-1} \otimes \cdots \otimes \boldsymbol{D}_1)\boldsymbol{x}. \tag{2.14}$$

Proof. Using equation 2.12 for mode-1 in equation 2.13, we have

$$\mathbf{Y}_{(1)} = \mathbf{D}_1 \mathbf{X}_{(1)} (\mathbf{D}_N \otimes \mathbf{D}_{N-1} \otimes \cdots \otimes \mathbf{D}_2)^T, \tag{2.15}$$

[3] It is worth mentioning that equation 2.12 is also valid for factor matrices with more columns than rows, that is, with $R_n > I_n$.

and by applying the property of equation 2.7, we finally obtain the desired result, equation 2.14.

The result of proposition 1 tells us that the multilinear Tucker model structure for a multidimensional signal is equivalent to a linear representation (matrix equation) of the vectorized signal where the dictionary has a Kronecker structure. In the standard Tucker model, a core tensor $\underline{\mathbf{G}}$ usually has a much smaller size than $\underline{\mathbf{Y}}$ with $R_n \ll I_n$ and the main objective is to find such compressed decomposition, that is, to compute $\underline{\mathbf{G}}$ and factor matrices $\mathbf{A}_n$, usually with additional constraints. In contrast, in our approach, the data set (measurements) $\underline{\mathbf{Y}}$ and dictionaries $\mathbf{D}_n$ are assumed to be known, and our objective is to compute the core tensor $\underline{\mathbf{X}}$, which is assumed to be very sparse and is larger than $\underline{\mathbf{Y}}$ ($M_n \geq I_n$).

3 Sparse Representations of *N*-Way Arrays

Here we generalize the concept of sparse representations of vectors to N-way arrays through the equivalence between the Tucker model and the Kronecker structure. Formally, definition 1 can be applied to a vectorized version of an N-way array with respect to a Kronecker dictionary arriving at the following obvious generalization:

Definition 3 *(multiway sparsity). A multidimensional signal (N-way array) $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ is K-sparse with respect to the factors $\boldsymbol{D}_n \in \mathbb{R}^{I_n \times M_n}$ ($M_n \geq I_n$, $n = 1, 2, \ldots, N$) if its vectorized version $\boldsymbol{y}$ admits a K-sparse representation over the Kronecker dictionary $\boldsymbol{D} = \boldsymbol{D}_N \otimes \boldsymbol{D}_{N-1} \otimes \cdots \otimes \boldsymbol{D}_1$ according to representation 2.14.*

The columns of factor matrices $\mathbf{D_n}$ are interpreted as dictionary elements or atoms associated with each mode. For many problems, these atoms can be selected to resemble the coherent structures that appear in each mode of the input N-way array. In other applications, we may want to design such dictionaries in order to favor a sparse representation. In this work, we assume that the dictionaries $\mathbf{D}_n$ for $n = 1, 2, \ldots, N$ are known (mode-n dictionaries).

We see that an N-way array having a K-sparse representation with respect to a Kronecker dictionary has an equivalent Tucker representation (see equation 2.13) with a sparse core tensor $\underline{\mathbf{X}}$, that is, with only K nonzero entries. If we define the location of the nonzero entries in $\underline{\mathbf{X}}$ by $(i_1^k, i_2^k, \ldots, i_N^k)$ with $k = 1, 2, \ldots, K$, then we can express the N-way array $\underline{\mathbf{Y}}$ as a weighted sum of K rank-1 N-way arrays as

$$\underline{\mathbf{Y}} = \sum_{k=1}^{S} x_{i_1^k i_2^k \cdots i_N^k} \mathbf{D}_1(\,:, i_1^k) \circ \mathbf{D}_2(\,:, i_2^k) \circ \cdots \circ \mathbf{D}_N(\,:, i_N^k), \tag{3.1}$$

where the symbol $\circ$ stands for the outer product of vectors: $[\mathbf{a}_1 \circ \mathbf{a}_2 \circ \cdots \circ \mathbf{a}_N]_{i_1 i_2 \cdots i_N} = \mathbf{a}_1(i_1)\mathbf{a}_2(i_2)\cdots\mathbf{a}_N(i_N)$.

Without losing generality, we will assume throughout the letter that mode-n dictionaries have unit-norm columns ($\|\mathbf{D}_n(:,i)\|^2 = 1,\ i = 1, 2, \ldots M_n, n = 1, 2, \ldots, N$).

In the multiway sparsity definition, definition 3, the nonzero entries can be located anywhere within the core tensor $\underline{\mathbf{X}}$. Sparsity is a simple assumption that naturally arises in real life, but it leads to mathematical problems that are not easy to solve. Therefore, additional assumptions are needed in order to simplify the problems and develop more efficient algorithms, especially for high-dimensional data sets. Here, we propose block sparsity for tensor data sets motivated by the fact that in the real world, the nonzero coefficients are not evenly distributed and are likely to be grouped in blocks (in Figure 2, two motivating examples of natural images revealing block-sparse representations, are shown). In other words, block sparsity is a natural and realistic assumption that incorporates valuable prior information about signals in nature, because block-sparse signals are more likely to occur than totally random sparse representations.[4]

Definition 4 *(multiway block sparsity). A multidimensional signal (N-way array) $\underline{\boldsymbol{Y}} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ is $(S_1, S_2, \ldots, S_N)$-block sparse with respect to the factors $\boldsymbol{D}_n \in \mathbb{R}^{I_n \times M_n}$ $(n = 1, 2, \ldots, N)$ if it admits a Tucker representation based only on few S_n selected columns of each factor $(S_n \leq M_n)$, that is, if $\mathcal{I}_n = [i_n^1, i_n^2, \ldots, i_n^{S_n}]$ denote a subset of indices for mode n $(n = 1, 2, \ldots, N)$, then*

$$\underline{\boldsymbol{Y}} = \underline{\boldsymbol{X}} \times_1 \boldsymbol{D}_1 \times_2 \boldsymbol{D}_2 \times_3 \cdots \times_N \boldsymbol{D}_N, \tag{3.2}$$

with $x_{i_1 i_2 \cdots i_N} = 0\ \forall (i_1, i_2, \ldots, i_N) \notin \mathcal{I}_1 \times \mathcal{I}_2 \times \cdots \times \mathcal{I}_N$.

We typically assume that $S_n \ll M_n$ and $M_n \geq I_n$. In other words, multiway block sparsity assumes that the nonzero entries of the core tensor $\underline{\mathbf{X}}$ are located within a subarray (block) defined by $\underline{\mathbf{X}}(\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_N)$. In Figure 1, a comparison of the sparsity types used in this work is shown.

The following basic results are easily derived from the previous definitions:

Proposition 2 *(multiway block sparsity implies sparsity of the vectorized version of the signal). If an N-way array $\underline{\boldsymbol{Y}} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ is $(S_1, S_2, \ldots, S_N)$-block sparse with respect to factor matrices $\boldsymbol{D}_n \in \mathbb{R}^{I_n \times M_n}$ $(n = 1, 2, \ldots, N)$, then its vectorized version $\boldsymbol{y} = vec(\underline{\boldsymbol{Y}}) \in \mathbb{R}^{I_1 I_2 \cdots I_N}$ is K-sparse $(K = S_1 S_2 \cdots S_N)$ with respect to the Kronecker dictionary $\boldsymbol{D} = \boldsymbol{D}_N \otimes \boldsymbol{D}_{N-1} \otimes \cdots \otimes \boldsymbol{D}_1$.*

[4] This multiway block-sparsity structure was recently proposed in Caiafa and Cichocki (2012)

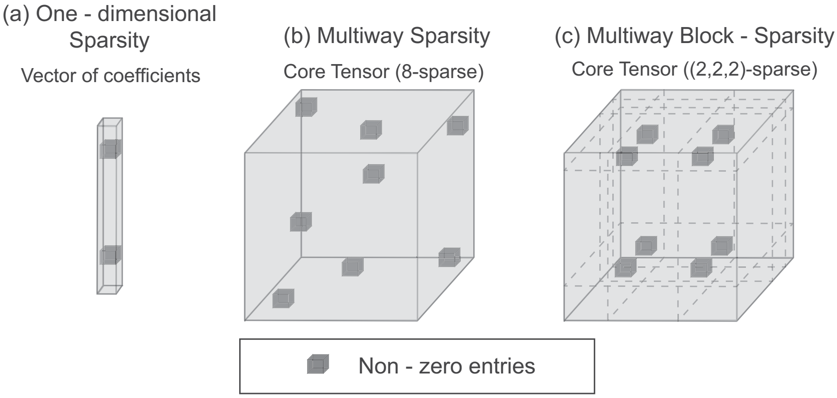
(a) One - dimensional Sparsity
Vector of coefficients
(b) Multiway Sparsity
Core Tensor (8-sparse)
(c) Multiway Block - Sparsity
Core Tensor ((2,2,2)-sparse)
Non - zero entries

Figure 1: (a) Distribution of nonzero entries within a vector of coefficients $\mathbf{x}$ (one-dimensional sparsity) and distribution of nonzero entries within the core tensor $\underline{\mathbf{X}}$ for (b) multiway sparsity (randomly distributed) and (c) multiway block sparsity.

Proof. If we use the equivalence of equations 2.13 and 2.14 and the definition of multiway block sparsity, we conclude that the vector of coefficients $\mathbf{x} = vec(\underline{\mathbf{X}})$ has at most $K = S_1 S_2 \cdots S_N$ nonzero entries, which means that $\mathbf{y}$ has a K-sparse representation on the dictionary $\mathbf{D}_N \otimes \mathbf{D}_{N-1} \otimes \cdots \otimes \mathbf{D}_1$.

Proposition 3 *(multiway block sparsity implies sparsity of mode-n vectors). If an N-way array $\underline{\boldsymbol{Y}} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ is $(S_1, S_2, \ldots, S_N)$-block sparse, then its mode-n vectors have an S_n-sparse representation with respect to $\boldsymbol{D}_n$ for each $n = 1, 2, \ldots, N$.*

Proof. *The N-way array $\underline{\boldsymbol{Y}}$ has a Tucker representation with a core tensor $\underline{\boldsymbol{X}}$ and factors $\boldsymbol{D}_n$ $(n = 1, 2, \ldots, N)$. By using equation 2.12, we obtain*

$$\boldsymbol{Y}_{(n)} = \boldsymbol{D}_n \boldsymbol{X}_{(n)} (\boldsymbol{D}_N \otimes \boldsymbol{D}_{N-1} \cdots \boldsymbol{D}_{n+1} \otimes \boldsymbol{D}_{n-1} \cdots \boldsymbol{D}_1)^T, \tag{3.3}$$

where we have written the mode-n vectors as linear combinations of elements in the mode-n dictionary $\boldsymbol{D}_n$, that is, $\boldsymbol{Y}_{(n)} = \boldsymbol{D}_n \boldsymbol{Z}_{(n)}$ with $\boldsymbol{Z}_{(n)} = \boldsymbol{X}_{(n)} (\boldsymbol{D}_N \otimes \boldsymbol{D}_{N-1} \cdots \boldsymbol{D}_{n+1} \otimes \boldsymbol{D}_{n-1} \cdots \boldsymbol{D}_1)^T$. Now we note that the mode-n unfolding matrix $\boldsymbol{X}_{(n)}$ has at least S_n rows with all-zero entries because $\boldsymbol{X}_{(n)}(i_n, j) = 0 \; \forall i_n \notin \mathcal{I}_n$ and therefore $\boldsymbol{Z}_{(n)}$ also has at least S_n rows with all-zero entries, which let us conclude that mode-n vectors are S_n-sparse.

Note that all n-vectors in each mode n belong to the same subspace with dimension at most S_n; that is, not only all these vectors can be sparsely represented over $\mathbf{D}_n$ but also they all use the same reduced set of dictionary elements, which means that the reciprocal of proposition 3 is not true in general; that is, sparsity of mode-n vectors does not implies multiway block sparsity.

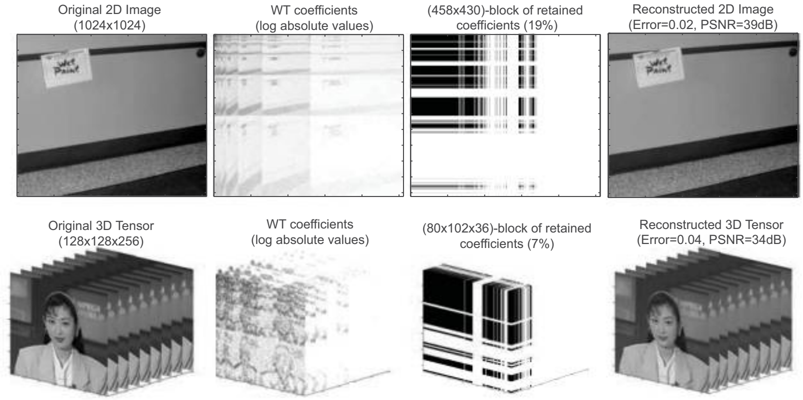

Figure 2: Examples of multiway block-sparsity approximation of multidimensional signals. (Top) "Wet paint" image (1024 × 1024, taken by Mike Wakin's research group in Duncan Hall at Rice University and available at http://www.ece.rice.edu/wakin/images/) is approximated by a selection of a 458 × 430 block in its Daubechies 8 (db8) wavelet transform (WT) domain (19% of coefficients). (Bottom row) Akiyo video sequence (128 × 128 × 256) is approximated by a selection of a 80 × 102 × 36 block in its WT domain (7% of coefficients). The selection of blocks was done by applying our N-BOMP algorithm (see section 5.2) to the signals.

It is important to mention that our definition of multiway block sparsity is different from the common definition of block-sparse representation introduced for 1D signals in Eldar, Kuppinger, and Bolcskei (2010). For 1D signals, block sparsity of order K usually refers to the fact that the vector of coefficients $\mathbf{x}$ is composed of a concatenation of blocks of length d, that is, $\mathbf{x} = [x_1, \ldots, x_d, x_{d+1}, \ldots, x_{2d}, \ldots, x_{N-d+1}, \ldots, x_N]^T$, with no more than K nonzero blocks.

4 Multidimensional Compressed Sensing

Compressed sensing (CS) (Donoho, 2006; Candès et al., 2006) proposes the reconstruction of compressible signals from a number of measurements significantly lower than the size of the signal. Let us consider a one-dimensional signal $\mathbf{z} \in \mathbb{R}^M$, which is assumed to have a K-sparse representation on the basis of $\mathbf{W} \in \mathbb{R}^{M \times M}$, that is, $\mathbf{z} = \mathbf{W}\mathbf{x}$ with $||\mathbf{x}||_0 \leq K$. Suppose that we cannot access the whole signal $\mathbf{z}$ but instead have available a set of I linear random projections obtained as follows:

$$\mathbf{y} = \mathbf{\Phi}\mathbf{z} = \mathbf{\Phi}\mathbf{W}\mathbf{x} = \mathbf{D}\mathbf{x}, \tag{4.1}$$

where $\mathbf{\Phi} \in \mathbb{R}^{I \times M}$ is the sensing matrix and we have defined $\mathbf{D} = \mathbf{\Phi}\mathbf{W}$. From equation 4.1, we see that we could reconstruct the signal $\mathbf{z}$ by identifying the proper sparse vector $\mathbf{x}$; in other words, the CS problem is reduced to solve an undetermined system of linear equations with the sparsity constraint as discussed in section 2.1.

The generalization of CS to higher dimensions is straightforward by assuming the Kronecker structure for the dictionary and the sensing matrix. This was recently proposed by Rivenson and Stern (2009a) for 2D signals and Duarte and Baraniuk (2011) for the general N-dimensional signals case. Formally, the vectorization of an N-way array $\underline{\mathbf{Z}} \in \mathbb{R}^{M_1 \times M_2 \times \cdots \times M_N}$ has a K-sparse representation on a Kronecker basis, that is, $\mathbf{z} = (\mathbf{W}_N \otimes \mathbf{W}_{N-1} \cdots \otimes \mathbf{W}_1)\mathbf{x}$ with $||\mathbf{x}||_0 \leq K$. Then, by using a global sensing matrix with Kronecker structure $\mathbf{\Phi} = \mathbf{\Phi}_N \otimes \mathbf{\Phi}_{N-1} \otimes \cdots \otimes \mathbf{\Phi}_1$, we obtain a large-scale linear system of equations with Kronecker structure as given by equation 2.14, where $\mathbf{D}_n = \mathbf{\Phi}_n \mathbf{W}_n$ $(n = 1, 2, \ldots, N)$.

A natural branch of CS is compressive imaging (CI) where, instead of collecting a large set of pixels and then compressing them to store in memory, CI seeks to minimize the collection of redundant data in the acquisition step. CI has been successfully implemented for 2D images in the celebrated single-pixel camera (Duarte et al., 2008). Besides, Rivenson and Stern (2009a, 2009b) showed that the Kronecker structure can be easily implemented for CI by separable imaging operators in optics, providing a practical implementation. Another application of CS using the Kronecker structure is hyperspectral compressive imaging (HCI), which consists of the acquisition of 2D images at several spectral bands providing a 3D signal where each slice corresponds to a different channel (Duarte & Baraniuk, 2011; Duarte et al., 2008). Here, a 2D-separable operator is applied to the hyperspectral light field, which means that each spectral band's image is multiplexed by the same sensing operator simultaneously. The resulting measurement matrix applied to the hyperspectral data cube can be represented as the following Kronecker product:

$$\mathbf{\Phi} = \mathbf{I} \otimes \mathbf{\Phi}_2 \otimes \mathbf{\Phi}_1, \tag{4.2}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{\Phi}_2 \otimes \mathbf{\Phi}_1$ is the separable sensing matrix for 2D images. Thus, HCI consists of computing the sparsest N-way array $\underline{\mathbf{X}}$ such as constrained to the available measurements:

$$\underline{\mathbf{Y}} = \underline{\mathbf{X}} \times_1 \mathbf{\Phi}_1 \mathbf{W}_1 \times_2 \mathbf{\Phi}_2 \mathbf{W}_2 \times_3 \mathbf{W}_3. \tag{4.3}$$

In the following section, we develop algorithms to solve this problem.

Table 1: Computing Sparse Representations of Multidimensional Signals.

Dictionary **D**	Sparsity Type	Algorithms
Nonstructured	Nonstructured	Classical OMP/BP
Kronecker	Nonstructured	Kronecker-OMP/Kronecker-BP
Kronecker	Multiway block sparsity	N-BOMP

5 Solving Large, Underdetermined Linear Systems with Kronecker Structure

It is clear that if the dictionary does not have a predefined structure, we can use any of the available algorithms such as OMP or BP. With a Kronecker dictionary, we can exploit its structure in order to reduce the computational complexity and memory requirements of a greedy algorithm as presented in section 5.1. It is worth mentioning that a BP algorithm that exploits the Kronecker structure has been proposed by Rivenson and Stern (2009a, 2009b), which we refer to here as Kronecker-BP (or Kronecker-SPGL1) algorithm. But we can do it better: if the multiway block-sparsity model is valid, we can use this structure to develop a very fast algorithm called N-BOMP, which we introduce in section 5.2. In Table 1 we summarize the algorithms and their applicability conditions that are covered in this letter.

In the following sections, we consider algorithms to compute a sparse representation of a multidimensional signal (N-way array) $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ given a fixed dictionary $\mathbf{D} = \mathbf{D}_N \otimes \mathbf{D}_{N-1} \otimes \cdots \otimes \mathbf{D}_1$, with $\mathbf{D}_n \in \mathbb{R}^{I_n \times M_n}$. To simplify the notation and facilitate the analysis of the associated complexity, we consider N-way arrays with the same size in each mode and dictionaries with the same number of atoms in each mode: $I_n = I$ and $M_n = M$ ($n = 1, 2, \ldots, N$).

5.1 The Kronecker-OMP Algorithm for Multiway Sparsity.

We exploit here the Kronecker structure of the dictionary to avoid the explicit storage of it and save memory resources. In other words, we can compute the equivalent multiway product, $\mathbf{R} \times_1 \mathbf{D}_1^T \times_2 \mathbf{D}_2^T \times_3 \cdots \times_N \mathbf{D}_N^T$ in algorithm 1, STEP 3, which also gives us much lower complexity. Note that M is related to the number of dictionary elements associated with each mode, which is usually in the range $M \approx 2I \sim 5I$ (underdetermined system of equations). For example, if $M = 2I$, with this simple technique we achieve a complexity of order $\mathcal{O}((2I)^{N+1})$, which is smaller compared to the complexity of the OMP algorithm applied to the vectorized signal (approximately $\mathcal{O}(2^{N+1}I^{2N})$).

Another expensive task in algorithm 1 is STEP 5—the least squares (LS) problem—because it refers to a very large explicit dictionary. We note that at iteration k, the approximation of the N-way array can be written in the following convenient vector form:

$$\mathbf{y} = (\mathbf{W}_N \odot \mathbf{W}_{N-1} \odot \cdots \odot \mathbf{W}_1)\mathbf{a}, \tag{5.1}$$

where matrices $\mathbf{W}_n \in \mathbb{R}^{I_n \times k}$ $(n = 1, 2, \ldots, N)$ contain in their columns the corresponding mode atoms selected in iterations $m = 1, 2, \ldots, k$; more precisely, $\mathbf{W}_n(:, m) = \mathbf{D}_n(:, i_n^m)$, the $\odot$ symbol stands for the Khatri-Rao product;[5] and the vector $\mathbf{a}$ contains the nonzero entries of the N-way array $\underline{\mathbf{X}}$, that is, $a_m = x_{i_1^m i_2^m \cdots i_N^m}$. Then the vector $\mathbf{a}$ in equation 5.1 is formally given by

$$\mathbf{a} = (\mathbf{W}_N \odot \mathbf{W}_{N-1} \odot \cdots \odot \mathbf{W}_1)^\dagger \mathbf{y}, \tag{5.2}$$

where † stands for the Moore-Penrose pseudoinverse. Since the pseudoinverse is rather expensive to compute, we apply the following property of the Khatri-Rao product (Kolda & Bader, 2009): $(\mathbf{A} \odot \mathbf{B})^\dagger = [(\mathbf{A}^T\mathbf{A}) * (\mathbf{B}^T\mathbf{B})]^\dagger (\mathbf{A} \odot \mathbf{B})^T$, with $*$ being the element-wise product of matrices (Hadamard product); we finally obtain

$$\mathbf{a} = \mathbf{Z}^\dagger (\mathbf{W}_N \odot \mathbf{W}_{N-1} \odot \cdots \odot \mathbf{W}_1)^T \mathbf{y}, \tag{5.3}$$

where $\mathbf{Z} = (\mathbf{W}_N^T\mathbf{W}_N) * (\mathbf{W}_{N-1}^T\mathbf{W}_{N-1}) * \cdots * (\mathbf{W}_1^T\mathbf{W}_1)$. If we assume that $\mathbf{Z}$ is non-singular, then $\mathbf{Z}^\dagger = \mathbf{Z}^{-1}$, and we can compute this inverse adaptively as follows. The matrix at iteration k, denoted by $\mathbf{Z}^{(k)}$, can be expressed in terms of the matrix used at step $(k-1)$,

$$\mathbf{Z}^{(k)} = \begin{pmatrix} \mathbf{Z}^{(k-1)} & \mathbf{b} \\ \mathbf{b}^T & 1 \end{pmatrix}, \tag{5.4}$$

with

$$\mathbf{b} = \left(\overline{\mathbf{W}}_N^T \mathbf{w}_N\right) * \left(\overline{\mathbf{W}}_{N-1}^T \mathbf{w}_{N-1}\right) * \cdots * \left(\overline{\mathbf{W}}_1^T \mathbf{w}_1\right), \tag{5.5}$$

where we defined the partition $\mathbf{W}_n = \left(\overline{\mathbf{W}}_n \; \mathbf{w}_n\right)$, that is, $\overline{\mathbf{W}}_n = \mathbf{W}_n(:, 1:k-1)$ and $\mathbf{w}_n = \mathbf{W}_n(:, k)$. Then the inverse at step k can be quickly computed using the Schur complement inversion formula for a block matrix:

[5] The Khatri-Rao product is obtained by applying the Kronecker product of columns, that is, given matrices $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2 \cdots \mathbf{a}_M]$ and $\mathbf{B} = [\mathbf{b}_1\ \mathbf{b}_2 \cdots \mathbf{b}_M]$, then $\mathbf{A} \odot \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1\ \mathbf{a}_2 \otimes \mathbf{b}_2 \cdots \mathbf{a}_M \otimes \mathbf{b}_M]$.

Algorithm 2: Kronecker-OMP

```
Require: mode-$n$ dictionaries $\{\mathbf{D}_1, \mathbf{D}_2, \cdots, \mathbf{D}_N\}$ with $\mathbf{D}_n \in \mathbb{R}^{I_n \times M_n}$, signal $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, sparsity $K$, tolerance $\epsilon$
Ensure: Sparse representation $vec(\underline{\mathbf{Y}}) \approx (\mathbf{D}_N \otimes \mathbf{D}_{N-1} \otimes \cdots \otimes \mathbf{D}_1)vec(\underline{\mathbf{X}})$ with $||vec(\underline{\mathbf{X}})||_0 \leq K$ (with nonzero entries given by $a_k = x_{i_1^k, i_2^k, ..., i_N^k}$, $k = 1, 2, ..., K$ )
$\mathcal{I}_n = [\emptyset]$ $(n = 1, 2, ..., N)$, $\mathbf{W}_n = \mathbf{0} \in \mathbb{R}^{I_n \times K}$, $\underline{\mathbf{R}} = \underline{\mathbf{Y}}$, $\underline{\mathbf{X}} = \underline{\mathbf{0}}$, $k = 1$;
while $k \leq K$ and $||\underline{\mathbf{R}}||_F > \epsilon$ do
    $[i_1^k, i_2^k, ..., i_N^k] = \arg\max_{[i_1 i_2 \cdots i_N]} |\underline{\mathbf{R}} \times_1 \mathbf{D}_1^T(:, i_1) \times_2 \mathbf{D}_2^T(:, i_2) \cdots \times_N \mathbf{D}_N^T(:, i_N)|$;
    $\mathcal{I}_n = [\mathcal{I}_n, i_n^k]$ $(n = 1, 2, ..., N)$; $\mathbf{W}_n(:, k) = \mathbf{D}_n(:, i_n^k)$
    $\mathbf{a} = \arg\min_{\mathbf{u}} ||(\mathbf{W}_N \odot \mathbf{W}_{N-1} \odot \cdots \odot \mathbf{W}_1)\mathbf{u} - vec(\underline{\mathbf{Y}})||_F^2$;
    $vec(\underline{\mathbf{R}}) = vec(\underline{\mathbf{Y}}) - (\mathbf{W}_N \odot \mathbf{W}_{N-1} \odot \cdots \odot \mathbf{W}_1)\mathbf{a}$;
    $k = k + 1$;
end while
return $\{\mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_N\}$, $\mathbf{a}$;
```

$$\left(\mathbf{Z}^{(k)}\right)^{-1} = \frac{1}{c}\begin{pmatrix} c\left(\mathbf{Z}^{(k-1)}\right)^{-1} + \mathbf{d}\mathbf{d}^T & \mathbf{d} \\ \mathbf{d}^T & 1 \end{pmatrix}, \tag{5.6}$$

where $c = 1 - \mathbf{b}^T\mathbf{b}$ and $\mathbf{d} = -(\mathbf{Z}^{(k-1)})^{-1}\mathbf{b}$.

Finally, at step k, the nonzero coefficients are computed by using equation 5.3, which can be written in the following equivalent form:

$$\mathbf{a} = \left(\mathbf{Z}^{(k)}\right)^{-1}\begin{pmatrix} \left(\overline{\mathbf{W}}_N \odot \overline{\mathbf{W}}_{N-1} \odot \cdots \odot \overline{\mathbf{W}}_1\right)^T \mathbf{y} \\ \left(\mathbf{w}_N \otimes \mathbf{w}_{N-1} \otimes \cdots \otimes \mathbf{w}_1\right)^T \mathbf{y} \end{pmatrix}. \tag{5.7}$$

As a result, the complexity of the update of the nonzero entries vector $\mathbf{a}$ is dominated by the computation of $\left(\mathbf{w}_N \otimes \mathbf{w}_{N-1} \otimes \cdots \otimes \mathbf{w}_1\right)^T \mathbf{y}$, which is only $\mathcal{O}(I^N)$, and the saving of memory storage is huge compared to the classical OMP with an explicit dictionary. The implementation of OMP for N-way arrays using a Kronecker dictionary (Kronecker-OMP) is given in algorithm 2.

Algorithm 3: N-BOMP

```
Require: mode-$n$ dictionaries $\{\mathbf{D}_1, \mathbf{D}_2, \cdots, \mathbf{D}_N\}$ with $\mathbf{D}_n \in \mathbb{R}^{I_n \times M_n}$, signal $\underline{\mathbf{Y}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, maximum number of nonzero entries $K_{\max}$ , tolerance
Ensure: Sparse representation $\underline{\mathbf{Y}} \approx \underline{\mathbf{X}} \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2 \times_3 \cdots \times_N \mathbf{D}_N$ with $x_{i_1 i_2 \cdots i_N} = 0 \ \forall (i_1, i_2, ..., i_N) \notin \mathcal{I}_1 \times \mathcal{I}_2 \times \cdots \times \mathcal{I}_N$ (with nonzero entries given by $\underline{\mathbf{X}}(\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_N) = \underline{\mathbf{A}}$).
1: $\mathcal{I}_n = [\emptyset]$ $(n = 1, 2, ..., N)$, $\underline{\mathbf{R}} = \underline{\mathbf{Y}}$, $\underline{\mathbf{X}} = \underline{\mathbf{0}}$, $k = 1$;
2: while $|\mathcal{I}_1||\mathcal{I}_2| \cdots |\mathcal{I}_N| \leq K_{\max}$ and $||\underline{\mathbf{R}}||_F >$ do
3:     $[i_1^k i_2^k \cdots i_N^k] = \arg\max_{[i_1 i_2 \cdots i_N]} |\underline{\mathbf{R}} \times_1 \mathbf{D}_1^T(:, i_1) \times_2 \cdots \times_N \mathbf{D}_N^T(:, i_N)|$;
4:     $\mathcal{I}_n = \mathcal{I}_n \cup [i_n^k]$ $(n = 1, 2, ..., N)$, $\mathbf{B}_n = \mathbf{D}_n(:, \mathcal{I}_n)$;
5:     $\mathbf{a} = \arg\min_{\mathbf{u}} ||(\mathbf{B}_N \otimes \mathbf{B}_{N-1} \otimes \cdots \otimes \mathbf{B}_1)\mathbf{u} - \mathbf{y})||_2^2$;
6:     $\underline{\mathbf{R}} = \underline{\mathbf{Y}} - \underline{\mathbf{A}} \times_1 \mathbf{B}_1 \times_2 \mathbf{B}_2 \cdots \times_N \mathbf{B}_N$;
7:     $k = k + 1$;
8: end while
9: return $\{\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_N\}, \underline{\mathbf{A}}$;
```

5.2 N-BOMP Algorithm: An Efficient Algorithm to Find Multiway Block-Sparse Representations. In this section we introduce algorithm 3, N-way Block OMP (N-BOMP), to find an $(S_1, S_2, \ldots, S_N)$-block sparse representation of an N-way array (see definition 4) with respect to the factors $\mathbf{D}_n \in \mathbb{R}^{I_n \times M_n}$ $(n = 1, 2, \ldots, N)$. We show that since the nonzero entries are restricted to being located within a subarray (block) of size $S_1 \times S_2 \times \cdots \times S_N$, they can be identified very quickly and in many fewer iterations compared to the Kronecker-OMP presented in the previous section (see algorithm 2).

If we denote by $\mathbf{B}_n \in \mathbb{R}^{I_n \times S_n}$ the submatrices obtained by restricting the mode-n dictionaries to the columns indicated by indices $\mathcal{I}_n$, that is, $\mathbf{B}_n = \mathbf{D}_n(:, \mathcal{I}_n)$, then the approximation of the signal is given by a Tucker model using these matrices as factors that can be written in vector form as

$$\hat{\mathbf{y}} = (\mathbf{B}_N \otimes \mathbf{B}_{N-1} \otimes \cdots \otimes \mathbf{B}_1)\mathbf{a}, \tag{5.8}$$

where $\mathbf{a} \in \mathbb{R}^K$ $(K = \prod_{n=1}^N S_n)$ is the vectorized version of the N-way array consisting of only nonzero entries. Accordingly, step 5 in algorithm 1 corresponds to the following general minimization problem:

$$\mathbf{a} = arg \min_{\mathbf{u}} ||(\mathbf{B}_N \otimes \mathbf{B}_{N-1} \otimes \cdots \otimes \mathbf{B}_1)\mathbf{u} - \mathbf{y})||_2^2, \tag{5.9}$$

where $\mathbf{y} \in \mathbb{R}^{I_1 I_2 \cdots I_N}$ is the vectorized version of the N-way array $\underline{\mathbf{Y}}$. By defining $\mathbf{B} = \mathbf{B}_N \otimes \mathbf{B}_{N-1} \otimes \cdots \otimes \mathbf{B}_1$, we see that the solution of this problem is given by $\mathbf{a} = [\mathbf{B}^T\mathbf{B}]^{-1}\mathbf{B}\mathbf{y}$, which means that $[\mathbf{B}^T\mathbf{B}]\mathbf{a} = \mathbf{B}\mathbf{y}$. This allows us to write

$$\mathbf{B}_1^T\mathbf{B}_1\mathbf{A}_{(1)}\left(\mathbf{B}_N^T\mathbf{B}_N \otimes \cdots \otimes \mathbf{B}_2^T\mathbf{B}_2\right) = \mathbf{B}_1^T\mathbf{Y}_{(1)}\left(\mathbf{B}_N^T \otimes \cdots \otimes \mathbf{B}_2^T\right). \tag{5.10}$$

By denoting $\underline{\mathbf{Z}}^{(1)} = \underline{\mathbf{A}} \times_1 \mathbf{I} \times_2 \mathbf{B}_2^T\mathbf{B}_2 \cdots \times_N \mathbf{B}_N^T\mathbf{B}_N$ and $\underline{\mathbf{P}} = \underline{\mathbf{Y}} \times_1 \mathbf{B}_1^T \times_2 \cdots \times_N \mathbf{B}_N^T$, we have

$$\mathbf{B}_1^T\mathbf{B}_1\left(\mathbf{Z}^{(1)}\right)_{(1)} = \mathbf{P}_{(1)}, \tag{5.11}$$

which can be solved for $(\mathbf{Z}^{(1)})_{(1)}$ efficiently by using a Cholesky factorization of the Hermitian matrix $\mathbf{B}_1^T\mathbf{B}_1$. Note that this is a relatively small problem because the size of the matrix $\mathbf{B}_1^T\mathbf{B}_1$ is only $|\mathcal{I}_1| \times |\mathcal{I}_1|$. Now, we can use the solution $\underline{\mathbf{Z}}^{(1)}$ of the subproblem, equation 5.11, and write its mode-2 unfolded version as

$$\mathbf{B}_2^T\mathbf{B}_2\mathbf{A}_{(2)}\left(\mathbf{B}_N^T\mathbf{B}_N \otimes \cdots \otimes \mathbf{B}_3^T\mathbf{B}_3 \otimes \mathbf{I}\right) = \left(\mathbf{Z}^{(1)}\right)_{(2)}, \tag{5.12}$$

where defining $\underline{\mathbf{Z}}^{(2)} = \underline{\mathbf{A}} \times_1 \mathbf{I} \times_2 \mathbf{I} \times_3 \mathbf{B}_3^T\mathbf{B}_3 \cdots \times_N \mathbf{B}_N^T\mathbf{B}_N$ leads us to the following simple subproblem also solved efficiently using the Cholesky factorization of the Hermitian matrix $\mathbf{B}_2^T\mathbf{B}_2$:

$$\mathbf{B}_2^T\mathbf{B}_2\left(\mathbf{Z}^{(2)}\right)_{(2)} = \left(\mathbf{Z}^{(1)}\right)_{(2)}. \tag{5.13}$$

By subsequently applying this procedure, after N steps, we finally arrive at the desired matrix $\mathbf{A}_{(N)}$, which corresponds to the coefficients in a mode-N matrix format for selected indices in the current iteration.

The N-BOMP algorithm not only optimizes the memory storage but also requires far fewer iterations compared to the classic OMP algorithm because the maximum number of iterations is $k_{\max} \ll K = S_1S_2\cdots S_N$, with K being the number of nonzero entries within the core tensor $\underline{\mathbf{X}}$ (see a complexity analysis below).

5.3 Complexity Analysis. Here we analyze the computational complexity associated with each of the algorithms discussed in this letter. For this analysis, we assume an N-way array $\underline{\mathbf{Y}} \in \mathbb{R}^{I\times I\times\cdots\times I}$ and mode-n dictionaries given by matrices $\mathbf{D}_n \in \mathbb{R}^{I\times M}$. We also assume that an

$(S, S, \ldots, S)$-block sparse representation of $\underline{\mathbf{Y}}$ with factors $\mathbf{D}_n$ ($n = 1, 2, \ldots, N$), which means that there are S^N nonzero coefficients. We consider the arithmetic operations required by step 3 (maximum correlated atom detection), step 5 (least squares estimation of nonzero coefficients), and step 6 (residual update) in all algorithms at iteration number k in terms of I (mode size), M (number of atoms per mode), and N (number of dimensions). The comparative results are summarized in Table 2 (see the details in the appendix) where the advantage of N-BOMP over Kronecker-OMP and OMP is evident. From this table, we observe that for very sparse representations with $S \ll I < M$, the complexity is dominated by step 3, which is exactly the same for Kronecker-OMP and N-BOMP. The key advantage of N-BOMP over the other algorithms is that it requires many fewer iterations ($\mathcal{O}(S)$ against $\mathcal{O}(S^N)$ iterations in OMP/Kronecker-OMP). In addition, in step 5, the N-BOMP algorithm complexity in terms of the number of entries I^N is sublinear compared to a linear dependence of the standard OMP and the Kronecker-OMP algorithms. In section 7, we show several numerical results with comparisons of the computation times required by different algorithms applied to multidimensional signals.

6 Algorithms Performance Guarantees

It is important to analyze under which conditions the proposed algorithms can obtain the true expected solution. We assume that an N-way array $\underline{\mathbf{Y}} \in \mathbb{R}^{I \times I \times \cdots \times I}$ has a $(S, S, \ldots, S)$-block sparse representation according to the model of definition 4 and the coherence of each mode-n dictionary is bounded by $\mu_n \leq \mu$ ($n = 1, 2, \ldots, N$). It is easy to see that if

$$S^N < \frac{1}{2}\left(1 + \frac{1}{\mu}\right), \tag{6.1}$$

then algorithm 1 (standard OMP) applied to $\mathbf{y} \in \mathbb{R}^{I^N}$ (the vectorized version of an N-way array) with a global dictionary $\mathbf{D} = \mathbf{D}_N \otimes \mathbf{D}_{N-1} \otimes \cdots \otimes \mathbf{D}_1$, will obtain the true sparse representation after S^N iterations as predicted by condition 2.3. It is also straightforward to see that this is true for algorithm 2 (Kronecker-OMP) applied to the N-way array $\underline{\mathbf{Y}} \in \mathbb{R}^{I \times I \times \cdots \times I}$ with mode-n dictionaries $\mathbf{D}_n \in \mathbb{R}^{I \times M}$ ($n = 1, 2, \ldots, N$).

The following theorem shows that for an N-way array generated as described before, the N-BOMP algorithm is able to obtain the true expected sparse representation in fewer iterations and under a less restrictive condition compared to equation 6.1.

Theorem 1 *(N-BOMP performance guarantee). Given the multiway decomposition* $\underline{\mathbf{Y}} = \underline{\mathbf{X}} \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2 \cdots \times_N \mathbf{D}_N$, *with a fixed N-way array* $\underline{\mathbf{Y}} \in \mathbb{R}^{I \times I \times \cdots \times I}$

Table 2: Complexity Analysis.

	OMP	Kronecker-OMP	N-BOMP
Number of Iterations	S^N	S^N	$\leq NS$
Step 3	$2(MI)^N + 2M^N$	$2M^N I \left(\frac{1-(I/M)^N}{1-I/M}\right) + 2M^N$	$2M^N I \left(\frac{1-(I/M)^N}{1-I/M}\right) + 2M^N$
Step 5	$(kI)^N + 3k^{2N}$	$2I^N + 2Nk^N I + (N+4)k^N + 7k^{2N}$	$\leq 2Nk(I+1) + Nk^2 + 2Nk^{N+1}$
Step 6	$2(Ik)^N + I^N$	$N(N-1)I^N + I^N$	$2kI^N + I^N$

Note: Iterations and operations required at iteration k for standard OMP, Kronecker-OMP and N-BOMP algorithms for an N-way array $\underline{\mathbf{Y}} \in \mathbb{R}^{I\times I\times I\cdots\times I}$ with dictionaries $\mathbf{D}_n \in \mathbb{R}^{I\times M}$ and the block sparsity parameter S (usually $S \ll I$).

and known dictionaries $\mathbf{D}_n \in \mathbb{R}^{I\times M}$ having coherences μ_n ($n = 1, 2, \ldots, N$), if a $(S, S, \ldots, S)$-block sparse solution exists satisfying

$$(S\mu)^N < 2 - (1 + (S-1)\mu)^N, \tag{6.2}$$

with $\mu = max\{\mu_1, \mu_2, \ldots, \mu_N\}$. Then algorithm 3 (N-BOMP) is guaranteed to find this sparse representation in K iterations with $S \leq K \leq NS$.

Proof. Suppose, without loss of generality, that the desired sparse core tensor $\underline{\mathbf{X}}$ is such that its nonzero entries are located at the subarray defined by the indices in the range $i_n = 1, 2, \ldots, S$ and the maximum absolute value is $|x_{11\cdots 1}|$. Thus,

$$\underline{\mathbf{Y}} = \sum_{i_1=1}^{S}\sum_{i_2=1}^{S}\cdots\sum_{i_N=1}^{S} x_{i_1 i_2\cdots i_N}\mathbf{D}(:, i_1)\circ\mathbf{D}(:, i_2)\circ\cdots\circ\mathbf{D}(:, i_N). \tag{6.3}$$

At the first step of the algorithm $k = 1$, the residual is set to $\underline{\mathbf{R}} = \underline{\mathbf{Y}}$. Thus, for the first step of choosing the largest absolute value $|x_{11\cdots 1}|$, we must require that

$$\begin{aligned}&|\underline{\mathbf{Y}} \times_1 \mathbf{D}_1^T(:, 1) \times_2 \mathbf{D}_2^T(:, 1)\cdots\times_N \mathbf{D}_N^T(:, 1)|\\ &\quad > |\underline{\mathbf{Y}} \times_1 \mathbf{D}_1^T(:, j_1) \times_2 \mathbf{D}_2^T(:, j_2)\cdots\times_N \mathbf{D}_N^T(:, j_N)|,\end{aligned} \tag{6.4}$$

for $(j_1, j_2, \ldots, j_N) \neq (1, 1, \ldots, 1)$. Now we construct a lower bound for the left-hand side (LHS) and an upper bound for the right-hand side (RHS), and then pose the above requirement. Using equation 6.3, the LHS can be written as equal to

$$\left|\sum_{i_1=1}^{S}\sum_{i_2=1}^{S}\cdots\sum_{i_N=1}^{S} x_{i_1 i_2\cdots i_N} g_1(1, i_1) g_2(1, i_2)\cdots g_N(1, i_N)\right|, \tag{6.5}$$

where $g_n(i_n, j_n)$ are the elements of the Gram matrices $\mathbf{G}_n = \mathbf{D}_n^T\mathbf{D}_n$. By expanding this equation, we obtain

$$\begin{aligned}&\left|\underline{\mathbf{Y}} \times_1 \mathbf{D}_1^T(:, 1) \times_2 \mathbf{D}_2^T(:, 1)\cdots\times_N \mathbf{D}_N^T(:, 1)\right|\\ &\quad = \left|x_{11\cdots 1} + A_1 + A_2 + \cdots + A_N\right|,\end{aligned} \tag{6.6}$$

where

$$A_1 = \sum_{n=1}^{N}\sum_{i_n=2}^{S} x_{11\cdots i_n\cdots 1} g_n(1, i_n),$$

$$A_2 = \sum_{n \neq m} \sum_{i_n=2}^{S} \sum_{i_m=2}^{S} x_{11\cdots i_n \cdots i_m \cdots 1} g_n(1, i_n) g_m(1, i_m),$$

$$A_N = \sum_{i_1=2}^{S} \sum_{i_2=2}^{S} \cdots \sum_{i_N=2}^{S} x_{i_1 i_2 \cdots i_N} g_1(1, i_1) g_2(1, i_2) \cdots g_N(1, i_N).$$

Using the definition of coherence and the fact that the maximum absolute nonzero entry is $x_{11\cdots 1}$, we can write the bounds $|A_n| \leq \binom{N}{n}(S-1)^n |x_{11\cdots 1}| \mu^n$, which allows us to write the following lower bound for the LHS of equation 6.4:

$$|x_{11\cdots 1}| \left[1 - \sum_{n=1}^{N} \binom{N}{n} (S-1)^n \mu^n \right] = |x_{11\cdots 1}| \left[2 - (1 + (S-1)\mu)^N \right]. \tag{6.7}$$

Similarly, for the RHS of equation 6.4, we can write it equal to

$$\left| \sum_{i_1=1}^{S} \sum_{i_2=1}^{S} \cdots \sum_{i_N=1}^{S} x_{i_1 i_2 \cdots i_N} g_1(i_1, j_1) g_2(i_2, j_2) \cdots g_N(i_N, j_N) \right|, \tag{6.8}$$

which allows us to to write the following upper bound for the RHS of equation 6.4:

$$\sum_{i_1=1}^{S} \sum_{i_2=1}^{S} \cdots \sum_{i_N=1}^{S} |x_{11\cdots 1}| \mu^N = |x_{11\cdots 1}| (S\mu)^N. \tag{6.9}$$

Finally, using equations 6.7 and 6.8 and by requiring

$$(S\mu)^N < 2 - (1 + (S-1)\mu)^N, \tag{6.10}$$

we guarantee that the first chosen entry is the largest in absolute value. After that, the algorithm updates the residual, which again can be written in terms of the same dictionary atoms in each mode. Then, by using the same steps, we obtain that condition 6.2 guarantees that the algorithm again finds an index from the support of the expected solution. The maximum number of iterations is NS, which corresponds to the worst scenario in which only one new index is added to each mode-n dictionary at every iteration. The minimum number of iterations is S, which corresponds to the best scenario, where a new index is added to each mode at every iteration.

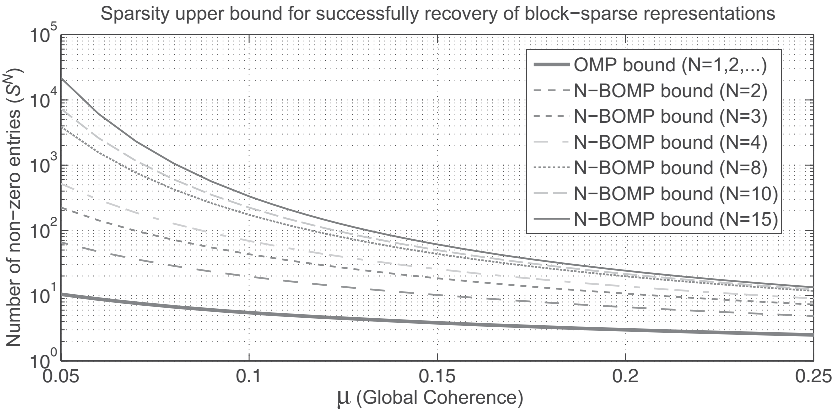

Figure 3: Comparison of successfull recovery bounds for N-BOMP and vectorized OMP (also Kronecker-OMP) algorithms.

Condition 6.2 establishes an upper bound for the number of nonzero entries S^N in the multiway block-sparse representation, which is less restrictive than condition 6.1, as illustrated in Figure 3, where this bound was computed numerically. We can see that the N-BOMP bound is always much larger than the classic OMP bound, and the situation improves for a higher number of dimensions N (at a fixed global coherence value μ). It is interesting to note that in practice, even beyond this upper bound, the algorithms are able to recover the correct representation with high probability, and N-BOMP always performs better than OMP for multiway block-sparse representations as our simulations in section 7 demonstrate.

7 Experimental Evaluation

In this section, we present several simulation results on synthetically generated signals and real-world signals in order to compare the performance of our algorithms against state-of-the-art methods like the classic OMP and BP. In particular, we evaluated an optimized version of a BP algorithm called SPGL1 (a solver for large-scale sparse reconstruction) (Berg & Friedlander, 2007), referred to here as Kronecker-SPGL1, which is optimized for large data sets and takes into account the Kronecker structure of the linear operator as used in Rivenson and Stern (2009a, 2009b) in the CI context. All the experiments were performed on a 2.65 GHz Intel Core i5 with 8GB RAM with Matlab using the Tensor Toolbox (Bader & Kolda, 2007) for efficient handling of multiway arrays. (A downloadable Matlab Demo package is available at http://web.fi.uba.ar/∼ccaiafa/Cesar/N-BOMPdemo.html, which reproduces our results for 2D and 3D CS as reported in this letter.)

7.1 Recovery of Exact Multiway Block-Sparse Representations. Here we analyze the capability of our algorithms to recover the correct set of nonzero coefficients and compare them against the classic OMP and BP (SPGL1). We have generated N-way arrays having exact multiway block-sparse representations according to definition 4 with $M_n = M$, $I_n = I$, and $S_n = S$ for $n = 1, 2, \ldots, N$. The indices that define the subarray of nonzero coefficients were randomly chosen, and the values of the nonzero entries were generated with independent and identically distributed (i.i.d.) gaussian numbers. We decide that a sparse representation was correctly recovered if the relative error based in the Frobenius norm is less than a threshold of 10^{-2}, that is, if $e = ||\mathbf{X} - \hat{\mathbf{X}}||_F / ||\mathbf{X}||_F < 10^{-2}$.

In experiment 1 (deterministic dictionary), we generated matrices $\mathbf{D}_n \in \mathbb{R}^{I\times 2I}$ by concatenating two orthogonal bases: the discrete cosine transform (DCT) basis and the canonical basis (spikes). More specifically, the dictionaries are generated as follows: $\mathbf{D}_n = [\mathbf{C}|\mathbf{I}]$ where the entries of the DCT matrix $\mathbf{C} \in \mathbb{R}^{I\times I}$ are defined by $c_{ij} = \sqrt{\frac{2}{I}} \cos(\frac{\pi(2j-1)(i-1)}{2I})$ for $1 < i \leq I$ and $c_{1j} = \sqrt{\frac{1}{I}}$ and the matrix $\mathbf{I} \in \mathbb{R}^{I\times I}$ is the identity matrix. It is known that sines and spikes provide highly incoherent matrices (Tropp, 2008), which is desirable in order to guarantee the recovery of sparse representations. In fact, the coherence of these matrices is $\mu_n = \mu(\mathbf{D}_n) = \sqrt{2/I}$, and therefore the global coherence of matrix $\mathbf{D} = \mathbf{D}_N \otimes \mathbf{D}_{N-1} \otimes \cdots \otimes \mathbf{D}_1$ is $\mu = \mu(\mathbf{D}) = \sqrt{2/I}$. These kinds of dictionaries (i.e., concatenation of orthogonal bases) have already been tested in the compressed sensing literature (Donoho & Elad, 2003; Gribonval, 2003). In Figure 4 we compare the percentage of correctly recovered representations over an ensemble of 100 simulations by using standard OMP/Kronecker-OMP, BP (implemented by SPGL1/Kronecker-SPGL1), and N-BOMP algorithms applied to generated signals for 1D, 2D, and 3D cases as a function of the total number of nonzero entries (S^N). It is important to note that the N-BOMP algorithm always performed better than standard OMP and BP. Furthermore, it is interesting to note that the maximum number of nonzero entries required by conditions 6.1 and 6.2 to guarantee the success of OMP and N-BOMP, respectively, is much lower than the actual required number of nonzero entries. For example, in Figure 4c, the global coherence is $\mu = 0.36$, which gives a maximum number of nonzero entries of $S^3_{\max} = 1.87$ (OMP) and $S^3_{\max} = 4.05$ (N-BOMP); nevertheless, our experimental results show that 100% of the cases were correctly recovered even with $S^3 = 27$ (OMP) and $S^3 = 64$ (N-BOMP), which reveals that the theoretical guarantees are very conservative (pesimistic) and the algorithms also perform well beyond those limits.

In experiment 2 (random dictionary), we consider that matrices $\mathbf{D}_n \in \mathbb{R}^{I\times M}$ are random as in the case of CS, because they are the result of multiplying a random-sensing matrix by an orthogonal matrix. Dictionaries have been generated using i.i.d. gaussian variables and applying normalization

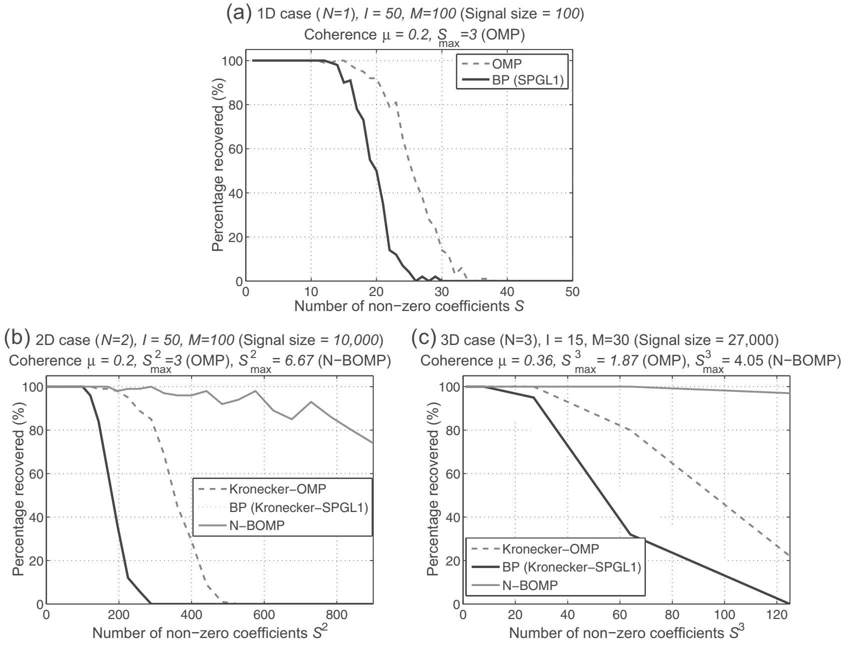

Figure 4: Experiment 1 (deterministic dictionary). Percentage of correctly recovered representations (relative error $e < 10^{-2}$) versus number of nonzero entries S^N by using Kronecker-OMP, Kronecker-BP (SPGL1), and N-BOMP algorithms for 1D (a), 2D (b), and 3D (c) signals. The coherence value μ is shown together with the theoretical guarantee bounds by using equations 6.1 and 6.2 for OMP and N-BOMP algorithms, respectively.

to the columns. It is well known that gaussian matrices have a high probability of low coherence (Tropp & Gilbert, 2007). In Figure 5 we compare the percentage of correctly recovered representations over an ensemble of 100 simulations by using standard OMP/Kronecker-OMP, BP (implemented by SPGL1/Kronecker-SPGL1), and N-BOMP algorithms applied to generated signals for 1D, 2D and 3D cases as a function of the total number of measurements (I^N). N-BOMP provides a higher percentage of correctly recovered signals for a few measurements. For example, in the 3D case for approximately 3000 measurements, which represents $3000/24^3 = 21.7\%$ of the whole signal, N-BOMP provides almost 90% of successfully recovered representations against 65% and 35% obtained by BP and Kronecker-OMP, respectively.

In Figure 6, the computation times required by Kronecker-OMP, Kronecker-BP, and N-BOMP algorithms are shown for experiment 1 (time versus the number of nonzero coefficients S^N) and experiment 2 (time versus

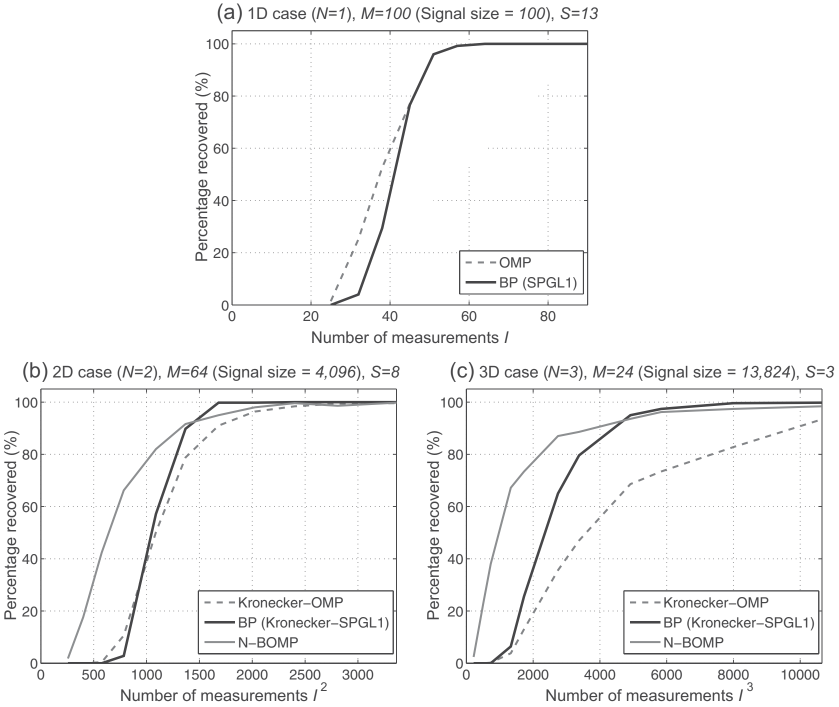

Figure 5: Experiment 2 (random dictionary). Percentage of correctly recovered representations (relative error $e < 10^{-2}$) versus number of measurements I^N by using Kronecker-OMP, Kronecker-BP (SPGL1), and N-BOMP algorithms for 1D (a), 2D (b), and 3D (c) signals.

the number of measurements I^N). It is important to highlight that we used the SPGL1 algorithm for the Kronecker-BP, which is optimally implemented by Berg and Friedlander (2007) in Matlab, including fast C implementations. Our implementation of Kronecker-OMP and N-BOMP does not include any C implementation. For experiment 1 (see Figures 6a and 6b), the complexity of Kronecker-OMP, as a function of the number of nonzero entries, grows faster than the cases of Kronecker-BP and N-BOMP. This is because the number of iterations required by Kronecker-OMP is S^N and for for N-BOMP is between S and NS (see section 5.3). On the other side, the number of iterations of Kronecker-BP depends on the convergence of the algorithm, not on the number of nonzero entries directly. For experiment 2 (see Figures 6c and 6d), we observe that Kronecker-OMP and N-BOMP show approximately flat plots because the number of nonzero entries is kept fixed to $S^2 = 8^2 = 64$ in Figure 6c and $S^3 = 3^3 = 27$ in Figure 6d. On the other side, as more measurements become available, the Kronecker-BP algorithm

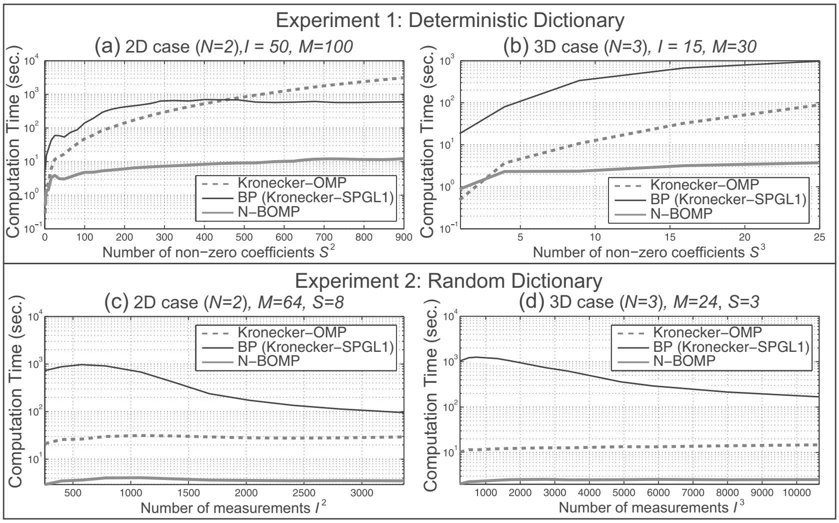

Figure 6: Computation times required by Kronecker-OMP, Kronecker-BP (SPGL1), and N-BOMP algorithms for 2D (left) and 3D (right) signals with multiway block-sparse representations generated according to experiment 1 (deterministic dictionary) (top) and experiment 2 (random dictionary) (bottom).

runs faster because it requires fewer iterations to converge. From all these plots, the advantage of N-BOMP algorithm over the other algorithms is clear.

7.2 Application to Multidimensional CS. Here we apply our N-BOMP algorithm to CS and compare it to a state-of-the-art method for the case of multidimensional images (large-scale data sets). More specifically we compare it to the optimized implementation of Kronecker-SPGL1 as proposed in Rivenson and Stern (2009a, 2009b), where the Kronecker structure is considered. We use the peak signal-to-noise Ratio (PSNR) to measure the quality of reconstruction of a multidimensional signal $\underline{\mathbf{Z}} \in \mathbb{R}^{M_1 \times M_2 \times \cdots \times M_N}$, which is defined as $\text{PSNR(dB)} = 20 \log(\max(z_{i_1 i_2 \cdots i_N})/||\underline{\mathbf{Z}} - \hat{\underline{\mathbf{Z}}}||_F)$. Our results demonstrate that by taking into account the multiway block sparsity, we are able to obtain much better results in terms of computation time and quality of reconstruction, as the following experiments clearly show.

7.2.1 Compressive Imaging. Our N-BOMP algorithm is very attractive for CI since the Kronecker structure can be forced in the implementation of sensors by separable optics demonstrated in Rivenson and Stern (2009a, 2009b).

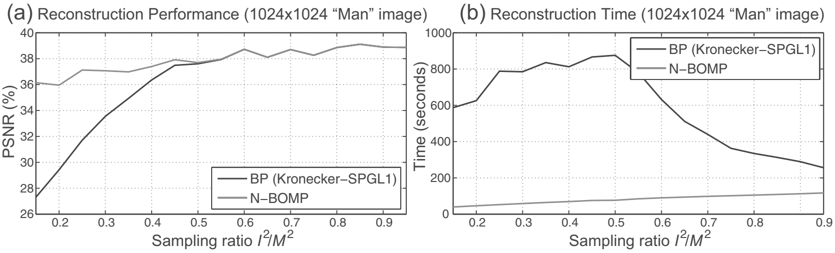

Figure 7: Compressive sensing of a 1-megapixel image. (a) Recovery performance in terms of the PSNR. (b) Computation times.

In order to put our algorithms in the context of the state-of-the-art methods for CI, in this section we consider the reconstruction of a 1-megapixel 2D image ($M \times M$ with $M = 1024$) benchmark already used in Candès and Romberg (2006) Candès and Wakin (2008), and Rivenson and Stern (2009a) where the original image was first processed by thresholding the largest 25,000 wavelet coefficients. Here, we kept 99,078 coefficients contained in a 294 × 337 2D block, which concentrates most of the signal energy in the Daubechies-separable wavelet transform domain (this block was selected by using the N-BOMP algorithm applied to the whole image with a target reconstruction relative error of 0.0874, the same error obtained by keeping the 25,000 largest coefficients). We have projected the input image by using gaussian sensing matrices $\mathbf{\Phi}_n \in \mathbb{R}^{I \times M}$ ($n = 1, 2$) for a wide range of the sampling ratio I^2/M^2. In Figure 7a, the obtained PSNRs for the Kronecker-SPGL1 and the N-BOMP algorithms as a function of the sampling ratio I^2/M^2 are shown. We see that for small sampling ratios, N-BOMP outperforms the BP strategy. For example, for a sampling ratio of 15%, we obtained PSNR values of 27 dB (Kronecker-SPGL1) and 35 dB (N-BOMP), as illustrated in Figure 8. In Figure 7b, the computation times required by these algorithms are shown. It is interesting to note that N-BOMP is always faster than the very optimized Kronecker-SPGL1 algorithm, which is extremely expensive for low levels of the sampling ratio due to the number of iterations required to converge to the solution.

7.2.2 Hyperspectral Compressive Imaging. Our sparse models for tensors are particularly relevant when dealing with large data sets with high dimensionality ($N \geq 2$) because existing vector techniques such as OMP or SPGL1 algorithms become prohibitively expensive or intractable. For example, large-scale problems with 3D signals cannot be processed with state-of-the-art algorithms (such as the Kronecker-BP based on SPGL1) without the aid of a super-computer. One example of such a problem is hyperspectral compressive imaging (HCI), introduced in section 4, which we illustrate in this section.

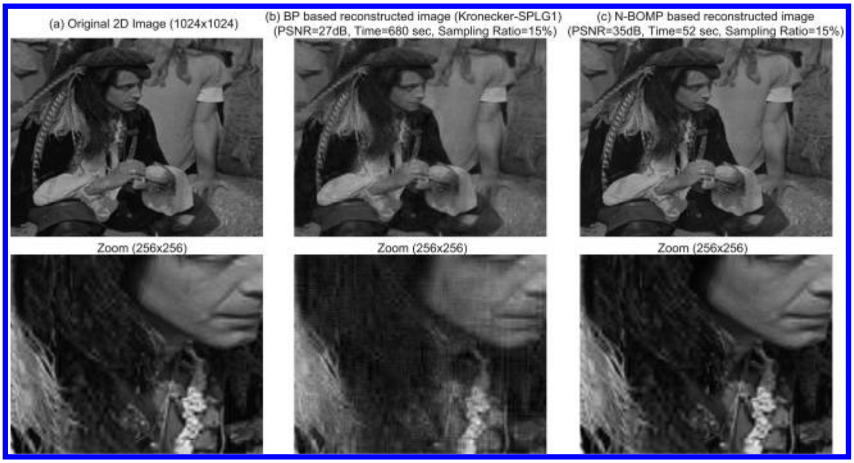
(a) Original 2D Image (1024x1024)
(b) BP based reconstructed image (Kronecker-SPLG1)
(PSNR=27dB, Time=680 sec, Sampling Ratio=15%)
(c) N-BOMP based reconstructed image
(PSNR=35dB, Time=52 sec, Sampling Ratio=15%)
Zoom (256x256)
Zoom (256x256)
Zoom (256x256)

Figure 8: Compressive imaging example of a 1-megapixel image using 397^2 samples (i.e., 15% of the original image size). (a) Original image approximated with 99,078 wavelet coefficients contained in a 294×337 2D block. (b) Kronecker-SPLG1 (Rivenson & Stern, 2009a) reconstruction. (c) N-BOMP reconstruction.

We show the results of applying our N-BOMP algorithm to two examples of natural scene hyperspectral images ($1024 \times 1024 \times 32$) corresponding to the signals extracted from scenes 7 and 8 in the Foster and Nascimento & Amano database (Foster, Nascimento, & Amano, 2004, available online at http://personalpages.manchester.ac.uk webpage). No preprocessing steps were applied to these data sets. These hyperspectral images contain scene reflectances measured at 32 different frequency channels acquired by a low-noise Peltier-cooled digital camera in the wave-length range of 400 to 720 nm (see the details in Foster et al., 2004).

As explained in section 4, for each channel, we apply a separable random-sensing matrix given by $\mathbf{\Phi} \times \mathbf{\Phi}$, where $\mathbf{\Phi} \in \mathbb{R}^{585\times1024}$ is a gaussian random matrix. We also assume that the data set has a multiway block-sparse representation on the separable Daubechies wavelet transform basis given by the matrix $\mathbf{W}_3 \otimes \mathbf{W}_2 \otimes \mathbf{W}_1$, with $\mathbf{W}_1 = \mathbf{W}_2 \in \mathbb{R}^{1024\times1024}$ and $\mathbf{W}_3 \in \mathbb{R}^{32\times32}$. The resulting sampling ratio is $\frac{585\times585\times32}{1024\times1024\times32} = 0.3263$. Results are shown in Figures 9 and 10. Quantitatively, the reconstructions perform very well with PSNR $= 35.26$ dB (global PSNR) for scene 7 and PSNR $= 42.52$ dB (global PSNR) for scene 8. This can be qualitatively verified by visual inspection of the images (the worst and best reconstructed slices are shown in Figures 9 and 10 with their corresponding PSNR values). It is important to highlight that these large data sets required only about 1 hour of computation time with our N-BOMP algorithm, which is very fast compared to the computation required by other state-of-the-art algorithms.

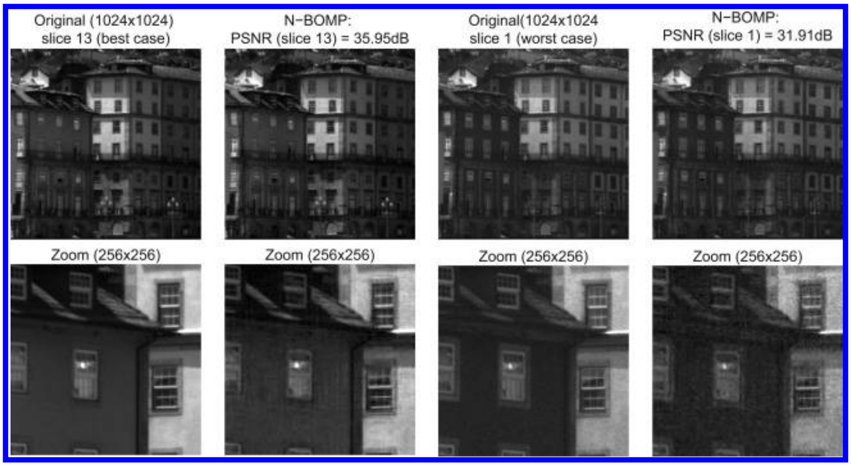

Figure 9: Hyperspectral compressive imaging (HCI) applied to the data set scene 07 (1024 × 1024 × 32) of the Foster et al. (2004) database using a sampling ratio of 33%. Original slices and their N-BOMP reconstructions are shown for the best case (slice 13, PSNR = 35.95 dB) and the worst case (slice 1, PSNR = 31.91 dB). The obtained global PSNR and computation times were (35.26 dB, 69 min).

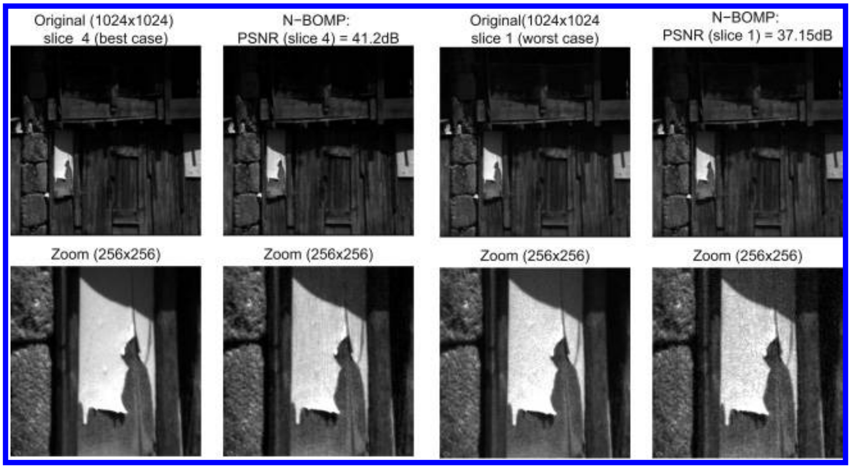

Figure 10: Hyperspectral compressive imaging (HCI) applied to the data set scene 08 (1024 × 1024 × 32) of the Foster et al. (2004) database using a sampling ratio of 33%. Original slices and their N-BOMP reconstructions are shown for the best case (slice 4, PSNR = 41.2 dB) and the worst case (slice 1, PSNR = 37.15 dB). The obtained global PSNR and computation times were (42.52 dB, 71 min).

8 Conclusion

Sparsity is a simple assumption that naturally arises in real life, but it leads to mathematical problems that are not easy to solve. Therefore, additional assumptions are needed in order to simplify the problems and develop more efficient algorithms, especially for high-dimensional data sets. In this work, we demonstrated the advantage of keeping the multidimensional structure of a data set by applying N-way array algorithms instead of classical vector algorithms. Since multidimensional signals usually have a sparse representation over, for example, the cosine, fourier, or wavelet separable transforms, it is reasonable to consider that dictionaries of multidimensional signals can be modeled by a Kronecker product of mode dictionaries. Additionally, motivated by the fact that in the real world, the nonzero coefficients are not evenly distributed and are likely to be grouped in blocks, we have introduced block sparsity, which allowed us to develop N-BOMP, a very efficient algorithm. In other words, block sparsity is a natural and realistic assumption that incorporates valuable prior information about signals, because block-sparse signals are more likely to occur than totally random sparse representations. Moreover, block-sparse signals are better recovered; in fact, the theoretical bound for N-BOMP is better than the theoretical bound corresponding to the vector case (OMP). We have provided extensive experimental results demonstrating the tremendous advantages of using algorithm 3 (N-BOMP), especially for higher-dimensional signals. Examples of application of N-BOMP algorithm to compressed sensed (CS), especially to 2D compressive imaging (CI) and 3D hyperspectral CI, are presented showing the efficiency and usefulness of our algorithm compared to available state-of-the-art algorithms such as classical OMP and Kronecker-BP. Summarizing, our work showed that block-sparsity naturally arises in nature; incorporating block sparsity allows one to dramatically reduce the complexity of algorithms; and the block-sparsity assumption also allows one to obtain better-quality reconstructions of signals, as our experimental results demonstrated.

Appendix: Detailed Complexity Analysis

A.1 OMP Algorithm (Algorithm 1). We begin with analysis of algorithm 1 in its recent optimized version as explained in Rubinstein et al. (2008). This algorithm has as input the vector $\mathbf{y} \in \mathbf{R}^{I^N}$ and an explicit dictionary $\mathbf{D} \in \mathbb{R}^{I^N \times M^N}$ with $M > I$ (overcomplete case). Assuming that the algorithm is guaranteed to obtain the true sparse representation, it will need exactly S^N iterations:

Step 3: This step includes computing $\mathbf{D}^T\mathbf{r}$ and taking the absolute value maximum ($2(MI)^N + 2M^N$ operations).

Step 5: By using the Cholesky factorization method, this step requires computing $(kI)^N + 3k^{2N}$ operations (Rubinstein et al., 2008).

Step 6: This step includes computing $\mathbf{D}(:, \mathcal{I})\mathbf{a}$ and subtracting it from $\mathbf{y}$ $(2(Ik)^N + I^N$ operations)

A.2 Kronecker-OMP Algorithm (Algorithm 2). As in the previous case, assuming that the algorithm is guaranteed to obtain the true sparse representation, it will need exactly S^N iterations:

Step 3: This step includes computing $\underline{\mathbf{R}} \times_1 \mathbf{D}_1^T(:, i_1) \times_2 \mathbf{D}_2^T(:, i_2) \cdots \times_N \mathbf{D}_N^T(:, i_N)$ $(2M^N I + 2M^{N-1}I^2 + \cdots + 2MI^N = 2M^N I(\frac{1-(I/M)^N}{1-I/M})$ operations) and taking its absolute-value maximum $(2M^N$ operations).

Step 5: This step includes computing $(\mathbf{w}_N \otimes \mathbf{w}_{N-1} \otimes \cdots \otimes \mathbf{w}_1)^T \mathbf{y}$ $(2I^N$ operations), $\mathbf{b} = (\overline{\mathbf{W}}_N^T \mathbf{w}_N) * (\overline{\mathbf{W}}_{N-1}^T \mathbf{w}_{N-1}) * \cdots * (\overline{\mathbf{W}}_1^T \mathbf{w}_1)$ $(2Nk^N I + Nk^N$ operations), $\mathbf{d} = (\mathbf{Z}^{(k-1)})^{-1}\mathbf{b}$ $(2k^{2N}$ operations), $\mathbf{b}^T\mathbf{d}$ $(2k^N$ operations), updating the inverse matrix according to equation 5.4 $(2k^N + 3k^{2N})$, and computing the nonzero coefficients according to equation 5.7 $(2k^{2N})$. Thus, giving a total number of operations equals $2I^N + 2Nk^N I + (N+4)k^N + 7k^{2N}$.

Step 6: This step includes computing $(\mathbf{W}_N \odot \mathbf{W}_{N-1} \odot \cdots \odot \mathbf{W}_1)\mathbf{a}$ and subtracting it from $vec(\underline{\mathbf{Y}})$ $(N(N-1)I^N + I^N$ operations).

A.3 N-BOMP Algorithm (Algorithm 3). A distinctive characteristic of this algorithm compared to the previous ones is that, assuming that the algorithm is granted to obtain the true sparse representation, it will require much less iterations. More specifically, after the maximum correlated atom is detected in step 3, its position within the $(M \times M \times \cdots \times M)$ multiway array determines the indices $i_1^k, i_2^k, \ldots, i_N^k$ to be added to the current $\mathcal{I}_n$ $(n = 1, 2, \ldots, N)$ subsets. Some of these indices may already be included in the corresponding mode indices subsets. It is granted that at least one new index in one mode will be added. Thus, at every iteration, a situation between the following two extreme cases can happen: case 1: only one subset of indices $\mathcal{I}_n$ is incremented by 1 for some n; case 2: a new index is selected so $\mathcal{I}_n$ are incremented by 1 for every $n = 1, 2, \ldots, N$. Below, we analyze the complexity for the worst case (case 2). We note that the minimum number of iterations is S (case 2) and the maximum number of iterations is NS (case 1):

Step 3: The same as for the Kronecker-OMP algorithm $(2M^N I + 2M^{N-}I^2 + \cdots + 2MI^N = 2M^N I\left(\frac{1-(I/M)^N}{1-I/M}\right) + 2M^N$ operations).

Step 5: The update of the Cholesky factorization for each mode, similar to the case of the algorithm 1, requires $2kI + k^2 + 2k$ operations and solving a set of N equations of type 5.10 using again the

Cholesky factorization, requires $2Nk^{N+1}$ operations. Thus, in the worst case (case 2) where the Cholesky factorization update is needed for every mode, the total number of operations is $2NkI + Nk^2 + 2Nk + 2Nk^{N+1}$.

Step 6: This step includes computing $\underline{\mathbf{A}} \times_1 \mathbf{B}_1 \times_2 \mathbf{B}_2 \cdots \times_N \mathbf{B}_N$ ($2kI^N(\frac{1-(I/k)^N}{1-I/k}) \approx 2kI^N$ operations) and subtracting it from $\underline{\mathbf{Y}}$, giving us approximately $2kI^N + I^N$ operations where we assumed that $I \gg k$.

Acknowledgments

We are grateful to the anonymous reviewers for their comments, which helped us to improve this letter. This work has been supported in part by CONICET under grant PIP 2012-2014, number 11420110100021.

References

Bader, B. W., & Kolda, T. G. (2007). *MATLAB tensor toolbox version 2.2*. www.sandia.gov/~tgkolda/TensorToolbox/

Berg, E., & Friedlander, M. P. (2007). *SPGL1: A solver for large-scale sparse reconstruction.* http://www.cs.ubc.ca/labs/scl/spgl1

Bobin, J., Starck, J., Fadili, J., & Moudden, Y. (2007). Sparsity and morphological diversity in blind source separation. *IEEE Transactions on Image Processing*, *16*(11), 2662–2674.

Caiafa, C., & Cichocki, A. (2009). Estimation of sparse nonnegative sources from noisy overcomplete mixtures using map. *Neural Computation*, *21*(12), 3487–3518.

Caiafa, C., & Cichocki, A. (2010). Generalizing the column-row matrix decomposition to multi-way arrays. *Linear Algebra and Its Applications*, *433*(3), 557–573.

Caiafa, C. F., & Cichocki, A. (2012). Block sparse representations of tensors using Kronecker bases. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 2709–2712). Piscataway, NJ: IEEE.

Candès, E., & Romberg, J. (2006). Sparsity and incoherence in compressive sampling. *Inverse Problems*, *23*(3), 969–985.

Candès, E., Romberg, J., & Tao, T. (2006). Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, *52*(2), 489–509.

Candès, E., & Wakin, M. (2008). An introduction to compressive sampling. *IEEE Signal Processing Magazine*, *25*(2) 21–30.

Chen, S., & Donoho, D. (2001). Atomic decomposition by basis pursuit. *SIAM Review*, *43*(1) 129–159.

Cichocki, A., Zdunek, R., Phan, A. H., & Amari, S. I. (2009). *Nonnegative matrix and tensor factorizations*. Hoboken, NJ: Wiley.

Davis, G. M., Mallat, S. G., & Zhang, Z. (1994). Adaptive time-frequency decompositions. *Optical Engineering*, *33*, 2183–2191.

Donoho, D. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, *52*(4), 1289–1306.

Donoho, D. L., & Elad, M. (2003). Optimally sparse representation in general (nonorthogonal) dictionaries via minimization. *Proceedings of the National Academy of Science*, *100*, 2197–2202.

Duarte, M., & Baraniuk, R. (2011). Kronecker compressive sensing. *IEEE Transactions on Image Processing*, *21*(2), 494–504.

Duarte, M. F., Davenport, M. A., Takhar, D., Lakhar, J. N., Laska, J. N., Sun, T., et al. (2008). Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, *25*, 83–91.

Elad, M., & Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, *15*(12), 3736–3745.

Elad, M., Figueiredo, M., & Ma, Y. (2010). On the role of sparse and redundant representations in image processing. *Proceedings of the IEEE*, *98*(6), 972–982.

Elad, M., Starck, J., Querre, P., & Donoho, D. (2005). Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA) *Applied and Computational Harmonic Analysis*, *19*(3), 340–358.

Eldar, Y., Kuppinger, P., & Bolcskei, H. (2010). Block-sparse signals: Uncertainty relations and efficient recovery. *IEEE Transactions on Signal Processing*, *58*(6), 3042–3054.

Figueiredo, M., Nowak, R., & Wright, S. (2007). Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems *IEEE Journal of Selected Topics in Signal Processing*, *1*(4), 586–597.

Ford, J., & Tyrtyshnikov, E. (2003). Combining Kronecker product approximation with discrete wavelet transforms to solve dense, function-related linear systems. *SIAM Journal on Scientific Computing*, *25*, 961–981.

Foster, D. H., Nascimento, S. M. C., & Amano, K. (2004). Information limits on neural identification of colored surfaces in natural scenes. *Visual Neuroscience*, *21*, 331–336.

Gribonval, R. (2003). Sparse representation in union of bases. *IEEE Transactions on Information Theory*, *49*(12), 3320–3325.

Jokar, S., & Mehrmann, V. (2009). Sparse solutions to underdetermined Kronecker product systems. *Linear Algebra Appl.*, *431*(12), 2437–2447.

Kolda, T., & Bader, B. (2009). Tensor decompositions and applications. *SIAM Review*, *51*(3), 455–500.

Lathauwer, L. D., Moor, B. D., & Vandewalle, J. (2000a). A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, *21*, 1253–1278.

Lathauwer, L. D., Moor, B. D., & Vandewalle, J. (2000b). On the best rank-1 and rank- (r1,r2,..,rn) approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications*, *21*, 1324–1342.

Li, Y., Cichocki, A., Amari, S., Shishkin, S., Cao, J., & Gu, F. (2004). Sparse representation and its applications in blind source separation. In S. Thrün, L. K. Saul, & B. Schölkopf (Eds.), *Advances in neural information processing systems, 16* (pp. 241–248). Cambridge, MA: MIT Press.

Loan, C. V. (2000). The ubiquitous Kronecker product. *Journal of Computational and Applied Mathematics*, *123*(1–2), 85–100.

Loan, C., & Pitsianis, N. (1992). *Approximation with Kronecker products* (Tech. Rep.). Ithaca, NY: Cornell University.

Lu, Y., & Do, M. (2008). A theory for sampling signals from a union of subspaces. *IEEE Transactions on Signal Processing*, *56*(6), 2334–2345.

Nagy, J. G., & Kilmer, M. E. (2006). Kronecker product approximation for preconditioning in three-dimensional imaging applications. *IEEE Transactions on Image Processing*, *15*, 604–613.

Needell, D., & Tropp, J. (2009). CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, *26*(3). 301–321.

Needell, D., Tropp, J., & Vershynin, R. (2008). Greedy signal recovery review. In *Proceedings of the 42nd Asilomar Conference on Signals, Systems and Computers*, *2008* (pp. 1048–1050). Piscataway, NJ: IEEE.

Olshausen, B., & Field, D. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, *381*(6583), 607–609.

Oseledets, I. (2011). Tensor-train decomposition. *SIAM Journal on Scientific Computing*, *33*(5), 2295–2317.

Oseledets, I., Savostianov, D., & Tyrtyshnikov, E. (2008). Tucker dimensionality reduction of three-dimensional arrays in linear time. *SIAM J. Matrix Anal. Appl.*, *30*(3), 939–956.

Rivenson, Y., & Stern, A. (2009a). Compressed imaging with a separable sensing operator. *IEEE Signal Processing Letters*, *16*(6), 449–452.

Rivenson, Y., & Stern, A. (2009b). Practical compressive sensing of large images. In *16th International Conference on Digital Signal Processing* (pp. 1–8). Piscataway, NJ: IEEE.

Rubinstein, R., Zibulevsky, M., & Elad, M. (2008). *Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit* (Tech. Rep.). Haifa: Technion.

Tropp, J. (2004). Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, *50*(10), 2231–2242.

Tropp, J. A. (2008). On the linear independence of spikes and sines. *J. Fourier Anal. Appl.*, *14*(5–6), 838–858.

Tropp, J., & Gilbert, A. (2007). Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, *53*(12), 4655–4666.

Tropp, J., & Wright, S. (2010). Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, *98*(6), 948–958.

Tucker, L. R. (1963). Implications of factor analysis of three-way matrices for measurement of change. In C. W. Harris (Ed.), *Problems in measuring change* (pp. 122–137). Madison WI: University of Wisconsin Press.

Tyrtyshnikov, E. (2004). Kronecker-product approximations for some function-related matrices. *Linear Algebra and Its Applications*, *379*, 423–437.

Received January 26, 2012; accepted July 26, 2012.