

# HYBRID VECTORIAL AND TENSORIAL COMPRESSIVE SENSING FOR HYPERSPPECTRAL IMAGING

*Edgar A. Bernal and Qun Li*

PARC, A Xerox Company  
800 Phillips Rd.  
Webster, NY 14580

## ABSTRACT

Hyperspectral imaging has a wide range of applications; however, due to the high dimensionality of the data involved, the complexity and cost of hyperspectral imagers can be prohibitive. Exploiting redundancies along the spatial and spectral dimensions of a hyperspectral image of a scene has created new paradigms that do away with the limitations of traditional imaging systems. While Compressive Sensing (CS) approaches have been proposed and simulated with success on already acquired hyperspectral imagery, most of the existing work relies on the capability to simultaneously measure the spatial and spectral dimensions of the hyperspectral cube. Most real-life devices, however, are limited to sampling one or two dimensions at a time, which renders a significant portion of the existing work unfeasible. In this paper we propose a novel CS framework that is a hybrid between traditional vectorized approaches and recently proposed tensorial approaches, and that is compatible with real-life devices both in terms of the acquisition and reconstruction requirements.

*Index Terms*— Compressive sensing, hyperspectral imaging, high-order tensorial data representation, multilinear algebra

## 1. INTRODUCTION

Hyperspectral imaging is the process of using specialized sensors to collect image information across the electromagnetic spectrum, often beyond the visible electromagnetic wavelength range. A hyperspectral image can be represented as a three-dimensional data cube where the first and second dimensions correspond to spatial data and the third dimension corresponds to the spectral bands. Objects have their own respective fingerprints known as spectral signatures; consequently, there is a wide range of applications that rely on decomposing a two-dimensional image of a scene into its spectral bands in order to enable object identification within the scene. These applications include remote sensing, astronomy, mineralogy, agriculture, healthcare and surveillance, among others. The main disadvantages of hyperspectral imaging are related to the intrinsic high dimensionality of the

data, which imposes storage, computational, and sensitivity constraints. These factors drive up the complexity and cost of traditional hyperspectral imagers. Fortunately, there is significant redundancy in hyperspectral images along both the spatial and the spectral dimensions which can be exploited by judicious sampling and reconstruction techniques, which in turn gives rise to sensing equipment with reduced complexity and cost. One such set of techniques is Compressive Sensing (CS) [1, 2] which exploits the sparsity of a signal in order to integrate acquisition and compression. CS theory enables reconstruction of a sparse signal from a few linear measurements (relative to the size of the reconstructed signal) via suitable non-linear minimization processes.

Traditional CS theory is well-suited for sampling and reconstructing one-dimensional signals, and naive extensions of the CS framework to multidimensional problems typically rely on vectorial representations of the data (see [3, 4, 5] for examples of two-dimensional imaging tasks tackled with a vectorized CS approach), which result in increased computational and memory requirements. Applications involving color/hyperspectral imagery and video data, which are intrinsically high-order, are not adequately addressed by vectorial approaches.

The extension of CS theory to multidimensional data is a research topic that has received increased attention in recent years. One branch of research attempts to recover the best low-rank tensor approximation of the original signal: in [6], the existence and uniqueness of the best rank- $R$  tensor approximation in the case of third order tensorial data were proven; in [7], a two-stage process comprising fitting of a low-rank model followed by per-mode decompression was proposed; in [8], a stable method for reconstruction of a low multilinear rank tensor from a set of multilinear projections was introduced. A separate branch uses Kronecker product matrices as sparsifying bases which jointly model the structure present in all of the signal dimensions within a CS framework [9, 10] via an approach termed Kronecker Compressive Sensing (KCS). Some research has focused on the explicit application of sparse image models to hyperspectral imaging: an overview of the state-of-the-art sparse models for hyper-

spectral image modeling was provided in [11]; a method that relies on separate sensing of spectral rows and that is compatible with onboard systems with a pushbroom configuration was proposed in [12]; lastly, a hyperspectral imager based on a single pixel camera architecture was introduced in [13].

In this paper, we introduce a method that is a hybrid between traditional vectorized approaches and the Generalized Tensorial Compressive Sensing (GTCS) framework introduced in [14, 15]. Unlike tensor-based approaches, the proposed method is compatible with real-life devices which usually acquire and process at most two dimensions of the data cube at a time: for example, systems with a pushbroom configuration [12] are equipped with a linear array of sensors which at a given instant in time acquires a spectral row, that is, one spatial dimension across all wavelengths; in contrast, single-pixel-camera-based architectures [13] sample both spatial dimensions corresponding to a single spectral band at a given time. Note that tensorial approaches rely on sampling all modes of the hyperspectral data cube simultaneously, which renders them infeasible for implementation on traditional CS-based hyperspectral imaging devices. Also, unlike existing approaches, the proposed method is highly computationally efficient, and thus does not impose significant computational requirements on the underlying hardware platform.

This paper is organized as follows: Section 2 briefly reviews concepts and notation from multilinear algebra used throughout the paper. Section 3 describes the proposed system architecture and the sampling and reconstruction methodologies. Section 4 contains experimental validation of the proposed approach. Finally, Section 5 concludes the paper.

## 2. NOTATION AND TERMINOLOGY

Lower-case italic characters represent scalar values (*e.g.*,  $a, b$ ), bold-face characters represent vectors (*e.g.*,  $\mathbf{a}, \mathbf{b}$ ), capital italic characters represent matrices (*e.g.*,  $A, B$ ) and capital calligraphic characters represent tensors (*e.g.*,  $\mathcal{A}, \mathcal{B}$ ). A tensor is a multidimensional array. The order of a tensor corresponds to its number of modes. For instance, tensor  $\mathcal{X} \in \mathbb{R}^{N_1 \times \dots \times N_d}$  has order  $d$  and the dimensionality of its  $i^{\text{th}}$  mode (also denoted mode  $i$ ) is  $N_i$ .

The mode- $i$  product between tensor  $\mathcal{X} = [x_{\alpha_1, \dots, \alpha_i, \dots, \alpha_d}] \in \mathbb{R}^{N_1 \times \dots \times N_i \times \dots \times N_d}$  and matrix  $U = [u_{j, \alpha_i}] \in \mathbb{R}^{J \times N_i}$  is denoted by  $\mathcal{X} \times_i U$  and is of size  $N_1 \times \dots \times N_{i-1} \times J \times N_{i+1} \times \dots \times N_d$ . The elements of the mode- $i$  product satisfy  $(\mathcal{X} \times_i U)_{\alpha_1, \dots, \alpha_{i-1}, j, \alpha_{i+1}, \dots, \alpha_d} = \sum_{\alpha_i=1}^{N_i} x_{\alpha_1, \dots, \alpha_i, \dots, \alpha_d} u_{j, \alpha_i}$ .

The mode- $i$  fibers of tensor  $\mathcal{X} = [x_{\alpha_1, \dots, \alpha_i, \dots, \alpha_d}] \in \mathbb{R}^{N_1 \times \dots \times N_i \times \dots \times N_d}$  are obtained by fixing every index but  $\alpha_i$ . The mode- $i$  unfolding of  $\mathcal{X}$ , denoted  $X_{(i)}$ , equals a matrix of size  $N_i \times (N_1 \cdot \dots \cdot N_{i-1} \cdot N_{i+1} \cdot \dots \cdot N_d)$  whose columns are the mode- $i$  fibers of  $\mathcal{X}$ .

## 3. PROPOSED APPROACH

Similar to most compressive sensing methodologies, the proposed approach is implemented in two stages: a sampling and a reconstruction stage. In the sampling stage, a sensing device obtains a set of linear measurements from the scene; a hyperspectral representation of the scene is then recovered from the measurements in the reconstruction stage. This is illustrated in Fig. 1.

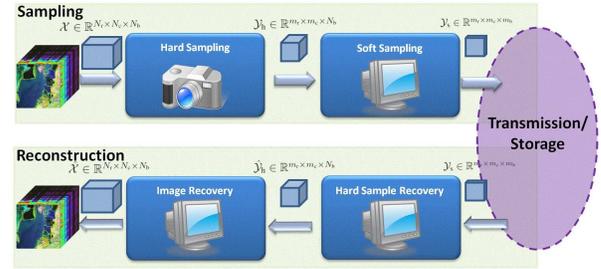


Fig. 1. Block diagram of proposed approach.

### 3.1. Sampling Stage

We denote the sampled hyperspectral cube representing an image of the scene as a tensor of order three  $\mathcal{X} \in \mathbb{R}^{N_r \times N_c \times N_b}$ , where  $N_r$  and  $N_c$  are the number of rows and columns of the image, respectively, and  $N_b$  is the number of bands. We refer to the row, column and spectral dimensions of the image as modes 1, 2 and 3 of the tensor, respectively. Sampling of  $\mathcal{X}$  is achieved by performing a set of mode- $i$  products between  $\mathcal{X}$  and sampling matrices  $U_i$ , for  $i = 1, 2, 3$ , one for each tensor mode. The entries of the sampling matrices are randomly generated, for example, drawn from a Gaussian distribution.

#### 3.1.1. Hard Sampling Module

The hard sampling module performs measurements across two tensorial modes of the tensor representing the hyperspectral image of the scene; this approach is realizable with real-life sensing devices such as those described in [12] and [13], the former sampling the two modes corresponding to the rows and spectral bands, and the latter sampling the two modes corresponding to the rows and columns. Without loss of generality, assume the hard sampling module performs sampling across the first two modes. Then the hard sampling module produces a set of hard samples that can be represented in the form of a tensor  $\mathcal{Y}_h$  satisfying

$$\mathcal{Y}_h = \mathcal{X} \times_1 U_1 \times_2 U_2, \quad (1)$$

where  $U_1 \in \mathbb{R}^{m_r \times N_r}$  and  $U_2 \in \mathbb{R}^{m_c \times N_c}$ . Here,  $m_r$  and  $m_c$  are the effective number of measurements along the columns and rows, respectively, and  $\mathcal{Y}_h \in \mathbb{R}^{m_r \times m_c \times N_b}$ .

#### 3.1.2. Soft Sampling Module

The soft sampling module takes as input the hard samples  $\mathcal{Y}_h$  computed by the hard sampling module and performs sam-

pling across the remaining tensorial mode. Since all the information about the scene required to perform soft sampling is included in  $\mathcal{Y}_h$ , this stage of sampling does not require involvement of the sensing device, and is performed via software operations. To this end, the mode-3 product between the tensor representing the hard samples and a sampling matrix  $U_3 \in \mathbb{R}^{m_b \times N_r}$  is performed in order to obtain a set of soft samples represented in the form of a tensor  $\mathcal{Y}_s \in \mathbb{R}^{m_r \times m_c \times m_b}$  satisfying

$$\mathcal{Y}_s = \mathcal{Y}_h \times_3 U_3. \quad (2)$$

### 3.2. Reconstruction Stage

The compressive sensing reconstruction module takes the set of soft samples as well as the known set of sampling matrices, and reconstructs the hyperspectral data corresponding to the scene being imaged by solving a set of  $\ell_1$  minimization tasks. Just like the measurement stage, the reconstruction stage can be broken down into two steps.

#### 3.2.1. Recovery of $\mathcal{Y}_h$

In the first step of the reconstruction stage,  $\hat{\mathcal{Y}}_h$ , an estimate of  $\mathcal{Y}_h$ , is recovered from  $\mathcal{Y}_s$  and  $U_3$ . To this end,  $\mathcal{Y}_s$  is mode-3 unfolded to obtain  $Y_{s(3)} \in \mathbb{R}^{m_b \times (m_r \cdot m_c)}$ . A matrix  $Z_{(3)} \in \mathbb{R}^{N_b \times (m_r \cdot m_c)}$  is then formed by stacking solutions  $\mathbf{z}_{(3)i}^* \in \mathbb{R}^{N_b \times 1}$  to the set of  $\ell_1$  minimization tasks

$$\begin{aligned} \mathbf{z}_{(3)i}^* &= \min\{\|\mathbf{z}_{(3)i}\|_1\} \quad \text{subject to} \\ \mathbf{y}_{(3)i} &= U_3 \mathbf{z}_{(3)i}, \text{ for } i = 1, 2, \dots (m_r \cdot m_c), \end{aligned} \quad (3)$$

where  $\mathbf{y}_{(3)i}$  are the columns of  $Y_{s(3)}$  and  $\mathbf{z}_{(3)i}^*$  are the columns of  $Z_{(3)}$ . Then  $\hat{\mathcal{Y}}_h = \mathcal{Z}$  is the tensor whose mode-3 unfolding is  $Z_{(3)}$ . Note that, as expected,  $\hat{\mathcal{Y}}_h \in \mathbb{R}^{m_r \times m_c \times N_b}$ .

#### 3.2.2. Recovery of $\mathcal{X}$

In the second step of the reconstruction stage,  $\hat{\mathcal{X}}$ , an estimate of  $\mathcal{X}$ , is recovered from  $\hat{\mathcal{Y}}_h = \mathcal{Z}$  and the sampling matrices  $U_1$  and  $U_2$ . To this end,  $\mathcal{Z}$  is mode-2 unfolded to obtain  $Z_{(2)} \in \mathbb{R}^{m_c \times (m_r \cdot N_b)}$ . A matrix  $W_{(2)} \in \mathbb{R}^{N_c \times (m_r \cdot N_b)}$  is then formed by stacking solutions  $\mathbf{w}_{(2)i}^* \in \mathbb{R}^{N_c \times 1}$  to the set of  $\ell_1$  minimization tasks

$$\begin{aligned} \mathbf{w}_{(2)i}^* &= \min\{\|\mathbf{w}_{(2)i}\|_1\} \quad \text{subject to} \\ \mathbf{z}_{(2)i} &= U_2 \mathbf{w}_{(2)i}, \text{ for } i = 1, 2, \dots (m_r \cdot N_b), \end{aligned} \quad (4)$$

where  $\mathbf{z}_{(2)i}$  are the columns of  $Z_{(2)}$  and  $\mathbf{w}_{(2)i}^*$  are the columns of  $W_{(2)}$ . Let  $\mathcal{W} \in \mathbb{R}^{m_r \times N_c \times N_b}$  be the tensor whose mode-2 unfolding is  $W_{(2)}$ . Note that  $\mathcal{W}$  is an estimate of  $\mathcal{X} \times_1 U_1$ . With that in mind,  $\mathcal{W}$  is mode-1 unfolded to obtain  $W_{(1)} \in \mathbb{R}^{m_r \times (N_c \cdot N_b)}$ . A matrix  $V_{(1)} \in \mathbb{R}^{N_r \times (N_c \cdot N_b)}$  is then formed by stacking solutions  $\mathbf{v}_{(1)i}^* \in \mathbb{R}^{N_r \times 1}$  to the set of  $\ell_1$  minimization tasks

$$\begin{aligned} \mathbf{v}_{(1)i}^* &= \min\{\|\mathbf{v}_{(1)i}\|_1\} \quad \text{subject to} \\ \mathbf{v}_{(1)i} &= U_1 \mathbf{w}_{(1)i}, \text{ for } i = 1, 2, \dots (N_c \cdot N_b), \end{aligned} \quad (5)$$

where  $\mathbf{w}_{(1)i}$  are the columns of  $W_{(1)}$  and  $\mathbf{v}_{(1)i}^*$  are the columns of  $V_{(1)}$ . Then  $\hat{\mathcal{X}} = \mathcal{V}$  is the tensor whose mode-1 unfolding is  $V_{(1)}$ . Note that, as expected,  $\hat{\mathcal{X}} \in \mathbb{R}^{N_r \times N_c \times N_b}$ . Although the uniqueness of this solution can be demonstrated, the proof is beyond the scope of this paper.

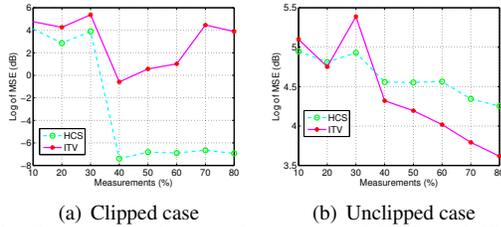
## 4. EXPERIMENTAL RESULTS

We tested the performance of the proposed algorithm both in terms of reconstruction accuracy and execution time. We compared our results to those achieved by the Iterative Total Variation algorithm (ITV) introduced in [12]. We consider ITV to be state-of-the-art among algorithms that are implementable in real-life devices since it was shown to outperform both KCS [9, 10], and one of its iterative implementations proposed in [16] under the feasibility constraints imposed by existing hyperspectral imaging devices. The plots in the figures in this section refer to the proposed algorithm as HCS, which stands for Hybrid Compressive Sensing.

In order to test the performance of the algorithm, acquisition and reconstruction of hyperspectral images from the AVIRIS Yellowstone dataset proposed in [17] was simulated. Reconstruction accuracy was measured in terms of Mean-Squared Error (MSE) between the reconstructed and original hyperspectral image, and execution time was measured in seconds on a Windows 7 machine with 16GB of RAM and an Intel i7 2.80GHz processor. The implementation of both algorithms was done in Matlab R2013b. The basis pursuit method provided by the  $\ell_1$ -MAGIC toolbox [18] was used to solve the optimization problems in Eqs. (3)-(5). Each data point in every figure corresponds to an average of five runs.

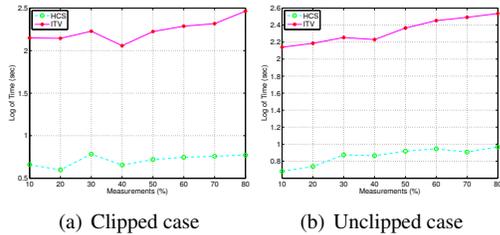
So as to keep the computational complexity of the problem manageable for the ITV algorithm, the hyperspectral images were cropped to  $32 \times 32$  pixels and 32 bands. In the first experiment, the rows and columns of the image were hard sampled. The plots in Fig. 2 contain results relative to reconstruction accuracy. Figure 2(a) plots the reconstruction error as a function of the percentage of number of samples acquired relative to the sparse representation of the original image in the discrete cosine transform (DCT) domain, where the sparse representation of the image is obtained by clipping DCT coefficients below a given threshold to 0. Figure 2(b) plots the reconstruction error as a function of the percentage of number of samples acquired relative to the full representation of the original image in the discrete cosine transform (DCT) domain. In the case when the clipped DCT representation of the image is used as the target image, the proposed algorithm outperforms ITV regardless of the number of samples. On the other hand, when the full DCT representation of the image is used as a reference, ITV outperforms the proposed algorithm when the number of samples exceeds 40% of the total number of image samples. These results indicate that when the image representation of the scene is not truly sparse in the target sparsifying domain, the proposed algorithm is

better suited to perform reconstruction when there are constraints in the number of samples that can be acquired (which happens for example, when the scene is dynamic or the imaging device is in motion); on the other hand, when the image is sparse in the target sparsifying domain, the proposed algorithm outperforms ITV regardless of the number of samples.



**Fig. 2.** Reconstruction performance of ITV and HCS with hard sampling of rows and columns.

The plots in Fig. 3 contain the execution times required to achieve the results illustrated in Fig 2. It can be seen that the proposed algorithm consistently and significantly outperforms ITV in terms of computational efficiency regardless of the sparsity assumptions imposed on the target image, as well as of the number of samples used in the reconstruction process.



**Fig. 3.** Reconstruction time of ITV and HCS with hard sampling of rows and columns.

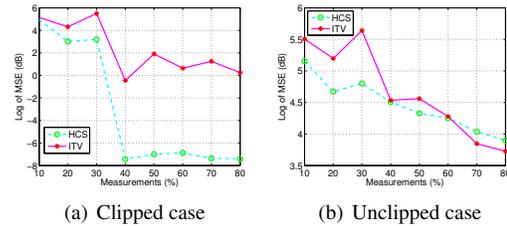
In the second experiment, the rows and spectral bands of the image were hard sampled. The plots in Fig. 4 contain results relative to reconstruction accuracy. Figure 4(a) plots the reconstruction error as a function of the percentage of number of samples acquired relative to the sparse representation of the original image in the discrete cosine transform (DCT) domain. Figure 4(b) plots the reconstruction error as a function of the percentage of number of samples acquired relative to the full representation of the original image in the discrete cosine transform (DCT) domain. In this case, the proposed algorithm more consistently outperforms ITV, except when the full representation of the image is maintained and a large number of samples are used in the reconstruction. These results indicate that the image is sparser in the DCT domain along its spectral dimension than along its spatial dimensions. The plots in Fig. 5 contain the execution times required to achieve the results illustrated in Fig 4. Once again, the proposed algorithm consistently and significantly outperforms ITV in terms of computational efficiency.

In the last experiment, we examined the behavior of the execution times of both ITV and the proposed algorithm as a

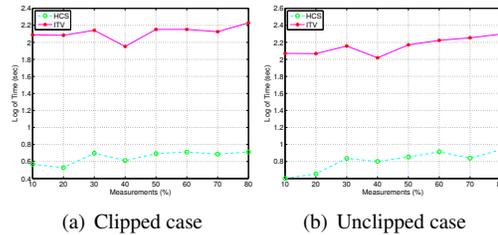
**Table 1.** Execution times as a function of image dimensionality.

Method	$n$	Time (sec)					
		2	3	4	5	6	7
HCS		0.06	0.3	1.6	8.3	18.6	111.8
ITV		11.0	24.9	59.4	231.7	3527.1	82361.0

function of the dimensionality of the data cube. To this end, we used target images with  $2^n$  rows,  $2^n$  columns and  $2^n$  bands for  $n = 2, 3, \dots, 7$ . Table 1 contains the results. It can be seen that the efficiency advantage of the proposed algorithm over ITV increases as the dimensionality of the target image increases.



**Fig. 4.** Reconstruction performance of ITV and HCS with hard sampling of rows and spectral bands.



**Fig. 5.** Reconstruction time of ITV and HCS with hard sampling of rows and spectral bands.

## 5. CONCLUSIONS

We have proposed a novel CS framework for the acquisition and reconstruction of hyperspectral imagery with real-life devices which are constrained to sampling one or two dimensions of the scene at any given time. While the main advantage of the proposed approach lies in its extreme computational efficiency, we demonstrated via simulation that its reconstruction performance is on par or better than that of the state-of-the-art algorithms under reasonable constraints, and only slightly worse in more general scenarios. We believe that, given the significant computational efficiency gains achieved, the proposed algorithm will enable the execution of CS algorithms in embedded portable architectures. We expect that this will reflect in an increased number of algorithms performing in quasi-real time on portable devices; this, since up until today, devices showcased significant delays in image reconstruction or were limited to offline reconstruction at a central processing unit with significant computational power requirements.

## 6. REFERENCES

- [1] E. Candès, “Compressive sampling,” in *Proceedings of the International Congress of Mathematicians*. European Mathematical Society, 2006, vol. I, pp. 1433–1452.
- [2] D.L. Donoho, “Compressed sensing,” *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, April 2006.
- [3] J. Romberg, “Imaging via compressive sampling,” *Signal Processing Magazine, IEEE*, vol. 25, no. 2, pp. 14–20, March 2008.
- [4] M.F. Duarte, M.A Davenport, D. Takhar, J.N. Laska, Ting Sun, K.F. Kelly, and R.G. Baraniuk, “Single-pixel imaging via compressive sampling,” *Signal Processing Magazine, IEEE*, vol. 25, no. 2, pp. 83–91, March 2008.
- [5] M.B. Wakin, J.N. Laska, M.F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K.F. Kelly, and R.G. Baraniuk, “An architecture for compressive imaging,” in *Image Processing, 2006 IEEE International Conference on*, Oct 2006, pp. 1273–1276.
- [6] L.H. Lim and P. Comon, “Multiarrray signal processing: Tensor decomposition meets compressed sensing,” *CoRR*, vol. 338, pp. 311–320, 2010.
- [7] N.D. Sidiropoulos and A. Kyrillidis, “Multi-way compressed sensing for sparse low-rank tensors,” *Signal Processing Letters, IEEE*, vol. 19, no. 11, pp. 757–760, 2012.
- [8] C.F. Caiafa and A Cichocki, “Fast and stable recovery of approximately low multilinear rank tensors from multi-way compressive measurements,” in *IEEE ICASSP 2014*, May 2014, pp. 6790–6794.
- [9] M.F. Duarte and R.G. Baraniuk, “Kronecker product matrices for compressive sensing,” in *IEEE ICASSP 2010*, March 2010, pp. 3650–3653.
- [10] M.F. Duarte and R.G. Baraniuk, “Kronecker compressive sensing,” *Image Processing, IEEE Transactions on*, vol. 21, no. 2, pp. 494–504, Feb 2012.
- [11] R.M. Willett, M.F. Duarte, M.A Davenport, and R.G. Baraniuk, “Sparsity and structure in hyperspectral imaging : Sensing, reconstruction, and target detection,” *Signal Processing Magazine, IEEE*, vol. 31, no. 1, pp. 116–126, Jan 2014.
- [12] S. Kamdem Kuiteing, G. Coluccia, A. Barducci, M. Barni, and E. Magli, “Compressive hyperspectral imaging using progressive total variation,” in *IEEE ICASSP 2014*, May 2014, pp. 7794–7798.
- [13] F. Magalhães, M. Abolbashari, F. Arajo, M. Correja, and F. Farahi, “High-resolution hyperspectral single-pixel imaging system based on compressive sensing,” *Optical Engineering*, vol. 51, no. 7, pp. 1–6, July 2012.
- [14] S. Friedland, Q. Li, D. Schonfeld, and E. A. Bernal, *Two Algorithms for Compressed Sensing of Sparse Tensors*, Springer, 2014.
- [15] S. Friedland, Q. Li, and D. Schonfeld, “Compressive sensing of sparse tensors,” *Image Processing, IEEE Transactions on*, vol. 23, no. 10, pp. 4438–4447, Oct 2014.
- [16] G. Coluccia, S.K. Kuiteing, A Abrardo, M. Barni, and E. Magli, “Progressive compressed sensing and reconstruction of multidimensional signals using hybrid transform/prediction sparsity model,” *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, vol. 2, no. 3, pp. 340–352, Sept 2012.
- [17] AB. Kiely and M.A Klimesh, “Exploiting calibration-induced artifacts in lossless compression of hyperspectral imagery,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 47, no. 8, pp. 2672–2678, Aug 2009.
- [18] E. Candès and J Romberg, “11-magic: Recovery of sparse signals via convex programming,” 2005.