Coded Hyperspectral Imaging and Blind Compressive Sensing

Ajit Rajwade, David Kittle, Tsung-Han Tsai, David Brady and Lawrence Carin
Department of Electrical & Computer Engineering
Duke University
Durham, NC 27708-0291, USA

Abstract

Blind compressive sensing (CS) is considered for reconstruction of hyperspectral data imaged by a coded aperture camera. The measurements are manifested as a superposition of the coded wavelength-dependent data, with the ambient three-dimensional hyperspectral datacube mapped to a two-dimensional measurement. The hyperspectral datacube is recovered using a Bayesian implementation of blind CS. Several demonstration experiments are presented, including measurements performed using a coded aperture snapshot spectral imager (CASSI) camera. The proposed approach is capable of efficiently reconstructing large hyperspectral datacubes, including hyperspectral video. Comparisons are made between the proposed algorithm and other techniques employed in compressive sensing, dictionary learning and matrix factorization.

Index Terms

hyperspectral images, image reconstruction, projective transformation, dictionary learning, non-parametric Bayesian, Beta-Bernoulli model, coded aperture snapshot spectral imager (CASSI).

I. INTRODUCTION

Feature-specific [1] and compressive sensing (CS) [2]–[4] have recently emerged as an important area of research in image sensing and processing. Compressive sensing has been particularly successful in multidimensional imaging applications, including magnetic resonance [5], projection [6], [7] and diffraction tomography [8], spectral imaging [9], [10] and video [11], [12]. Conventional sensing systems typically first acquire data in an uncompressed form (e.g., individual pixels in an image) and then perform compression subsequently, for storage or communication. In contrast, CS involves acquisition of the data in an already compressed form, reducing the quantity of data that need be measured in the first place. To perform CS, the underlying signal must be sparse or compressible in a basis or frame. In CS the
underlying signal to be measured is projected onto a set of vectors, and the vectors that define these compressive measurements should be incoherent with the vectors defining the basis/frame [4], [13]. If these conditions are met, one may achieve highly accurate signal reconstruction (even perfect, under appropriate conditions), using nonlinear inversion algorithms.

In most CS research it is assumed that one knows a priori the underlying basis in which the signal is compressible, with wavelets and local cosines [14] popular choices. Let \( x \in \mathbb{R}^M \) represent the underlying signal of interest, and \( x = \Psi \tilde{c} + \tilde{\nu} \), with \( \tilde{\nu} \in \mathbb{R}^M \); the columns of \( \Psi \in \mathbb{R}^{M \times M} \) define an orthonormal basis, \( \tilde{c} \in \mathbb{R}^M \) is sparse (i.e., \( \| \tilde{c} \|_0 \ll M \)), and \( \| \tilde{\nu} \|_2 \ll \| x \|_2 \). The vector \( \tilde{\nu} \) represents residual typically omitted after lossy compression [14].

Rather than directly measuring \( x \), in CS we seek to measure \( y \in \mathbb{R}^m \), with \( m \ll M \); measurements are defined by projecting \( x \) onto each of the rows of \( \Sigma \in \mathbb{R}^{m \times M} \). Specifically, we measure \( y = \Phi \tilde{c} + \tilde{\epsilon} \), with \( \Phi = \Sigma \Psi \) and \( \tilde{\epsilon} = \Sigma \tilde{\nu} + \tilde{\delta} \); \( \tilde{\delta} \) accounts for additional measurement noise. The aforementioned incoherence is desired between the rows of \( \Sigma \) and columns of \( \Psi \). Several nonlinear inversion algorithms have been developed for CS inversion and related problems [15]–[20].

In this paper we consider an alternative measurement construction and inverse problem. Rather than seeking to measure data associated with a single \( x \in \mathbb{R}^M \), we wish to measure \( \{ y_i \}_{i=1,N} \) and jointly recover \( \{ x_i \}_{i=1,N} \), with \( x_i \in \mathbb{R}^M \) and \( y_i \in \mathbb{R}^m \), again with \( m \ll M \). It is assumed that each \( x_i = D c_i + \nu_i \), where \( D \in \mathbb{R}^{M \times K} \), and typically \( K > M \) (\( D \) is overcomplete); \( c_i \) is sparse, and \( \nu_i \) again represents residual. Each measurement is of the form \( y_i = \Phi_i c_i + \epsilon_i \), with \( \Phi_i \in \mathbb{R}^{m \times K} \) defined in terms of matrix \( \Sigma_i \in \mathbb{R}^{m \times M} \) as \( \Phi_i = \Sigma_i D \), and \( \epsilon_i = \Sigma_i \nu_i + \delta_i \). In [21] the authors assumed \( \Sigma_i \) was the same for all \( i \), and in [22] it was demonstrated that there are significant advantages to allowing \( \Sigma_i \) and hence \( \Phi_i \) to change with index \( i \). In [21], [22] theoretical underpinnings are developed, with illustrative simulated experiments; in this paper we demonstrate how this framework may be applied to a real CS camera, with application to hyperspectral imaging. A key distinction with conventional CS is that we seek to recover \( D \) and \( \{ c_i \}_{i=1,N} \) simultaneously, implying that when performing the measurement we are “blind” to the underlying \( D \) in which each \( x_i \) may be sparsely rendered. This has been referred to as blind CS [21].

Signal models of the form \( x_i = D c_i + \nu_i \) are also called factor models, where the columns of \( D \) represent factor loadings. If one assumes that \( \{ c_i \}_{i=1,N} \) are block sparse (the sparsity patterns of \( \{ c_i \}_{i=1,N} \) are manifested in \( B \ll N \) blocks), and if \( \nu_i \) is assumed to be Gaussian, then this may also be viewed as a Gaussian mixture model (GMM). Models of this form have been employed successfully in CS [23]. The GMM representation may be used to approximate a manifold [23], and manifold signal models have
also proven effective in CS [24]. The $x_i = Dc_i + \nu_i$ representation is also related to a union-of-subspace model [25], particularly when $\{c_i\}_{i=1,N}$ are block sparse. The factor model, GMM, manifold and union-of-subspace models for $x_i$ have been demonstrated to often require far fewer CS measurements [23]–[25] than the ortho-basis model $x_i = \Psi \tilde{c}_i + \tilde{\nu}_i$. While the reduced number of CS measurements required of such formulations is attractive, previous CS research along these lines has typically assumed a priori knowledge of the detailed signal model. One therefore implicitly assumes prior access to appropriate training data, with which the signal model (e.g., dictionary $D$) may be learned; access to such data may not always be possible. In blind CS [21], [22] the form of the signal model $x_i = Dc_i + \nu_i$ is assumed, but $D$ and $\{c_i\}_{i=1,N}$ are inferred jointly based on $\{y_i\}_{i=1,N}$ (implying joint learning of the detailed signal model and associated data $\{x_i\}_{i=1,N}$).

Blind CS is related to dictionary learning [26]–[28], in which one is given $\{x_i\}_{i=1,N}$, and the goal is to infer the dictionary $D$. In many examples of this form one is given a large image, which is divided into small (overlapping) blocks (“patches”), with the collection of $N$ patches defining $\{x_i\}_{i=1,N}$. Application areas include image denoising and recovery of missing pixels (“inpainting”). In most previous dictionary learning research the underlying data $\{x_i\}_{i=1,N}$ was assumed observed (at least partially, in the context of inpainting), and compressive measurements were not employed.

We extend dictionary learning to blind CS, and demonstrate how this framework may be utilized to analyze data measured by a real CS camera. Specifically, we consider a coded aperture snapshot spectral imaging (CASSI) camera [29], [30], and demonstrate that data measured by such a system is ideally matched to the blind-CS paradigm. Previous inversion algorithms applied to CASSI data did not employ the blind-CS perspective. The reconstruction was accomplished using optimization algorithms, such as gradient projection for sparse reconstruction (GPSR) [29], and two-step iterative shrinkage/thresholding (TwIST) [30]. GPSR assumes sparsity of the entire image in a fixed (wavelet) basis, while TwIST is based on a piecewise-flat spatial intensity model for hyperspectral images. These methods do not account for correlation in the datacube as a function of wavelength, nor do they explicitly take into account the non-local self-similarity of natural scenes [31]. We develop a new inversion framework based on Bayesian dictionary learning, in which (i) a dictionary is learned to compactly represent patches in the form of small spatio-spectral cubes, and (ii) a Gaussian process is employed to explicitly account for correlation with wavelength. Related research was considered in [32], but each row of $\Sigma_i$ was composed of all zeros and a single one. In this paper we demonstrate how these methods may be applied to the CASSI camera, with more sophisticated $\Sigma_i$.

The remainder of the paper is organized as follows. In Section II we present a summary of the CASSI
camera, and how it yields measurements that are well aligned with the blind-CS paradigm. In Section III we describe how the proposed Bayesian dictionary-learning framework may be employed for blind-CS inversion. Experimental results are presented in Section IV, with comparison to alternative inversion algorithms. Conclusions are provided in Section V.

II. CASSI CAMERA AND BLIND CS

A. Mathematical representation of CASSI measurement

Assume we are interested in measuring a hyperspectral datacube \( \mathbf{X} \in \mathbb{R}^{N_x \times N_y \times N_{\lambda}} \), where the data at each wavelength corresponds to an \( N_x \times N_y \) image, and \( N_{\lambda} \) represents the number of wavelengths. Let \( \mathbf{X}_j \in \mathbb{R}^{N_x \times N_y} \) represent the image at wavelength \( \lambda_j \), for \( j \in \{1, \ldots, N_{\lambda}\} \). In a CASSI camera, each of the \( \mathbf{X}_j \) is multiplied by the same binary code \( \mathbf{C} \in \{0, 1\}^{N_x \times N_y} \), where typically the code is constituted at random, with each element drawn Bernoulli(\( p \)), with \( p \in (0, 1) \) (typically \( p = 0.5 \)). After this encoding, each wavelength-dependent image is represented as \( \hat{\mathbf{X}}_j = \mathbf{X}_j \cdot \mathbf{C} \), where \( \cdot \) denotes a pointwise or Hadamard product.

Let \( \hat{\mathbf{X}}_j(u, v) \) represent pixel \((u, v)\) in image \( \hat{\mathbf{X}}_j \). We now define a shifted version of \( \hat{\mathbf{X}}_j \), denoted \( \mathbf{S}_j \); \( \mathbf{S}_j(u, v) = \hat{\mathbf{X}}_j(u - \ell_j, v) \), where \( \ell_j > 0 \) denotes the shift in pixels at wavelength \( \lambda_j \), with \( \ell_j \neq \ell_{j'} \) for \( j \neq j' \). In defining \( \mathbf{S}_j(u, v) = \hat{\mathbf{X}}_j(u - \ell_j, v) \), we only consider \( u \) for which \( u - \ell_j \in \{1, \ldots, N_x\} \), and other components of \( \mathbf{S}_j(u, v) \) will not be measured, as made clear below.

The above construction yields a set of shifted, wavelength-dependent images \( \{\mathbf{S}_j\}_{j=1, N_{\lambda}} \). The CASSI measurement is a single two-dimensional image \( \mathbf{M} \), where component \( M(u, v) = \sum_{j=1}^{N_{\lambda}} \mathbf{S}_j(u, v) \), defined for all \( v \in \{1, \ldots, N_y\} \) and \( u \) for which \( \mathbf{S}_j \) is defined for all \( j \). Note that component \( M(u, v) \) corresponds to a superposition of coded data from all wavelengths, and because of the shifts \( \{\ell_j\}_{j=1, N_{\lambda}} \), the contribution toward \( M(u, v) \) at the different wavelengths corresponds to a different spatial location in the original datacube \( \mathbf{X} \). This also implies that the portion of the coded aperture contributing toward \( M(u, v) \) is different for each of the wavelengths.

A schematic of the physical composition of the CASSI camera is depicted in Figure 1. Note that the wavelength-dependent shift is manifested with a dispersive element [29], [30], characterized by wavelength-dependent velocity through a material of fixed dimension.

B. Blind CS representation

Consider a \( d \times d \) contiguous block of pixels in the measured CASSI image \( \mathbf{M} \); let this set of pixels be denoted \( \mathbf{y}_t \in \mathbb{R}^m \), where \( m = d^2 \). Because of the wavelength-dependent shift through the hyperspectral
Fig. 1. Summary of CASSI measurement process (see [30] for description of physical hardware). (a) The CASSI measurement corresponds to passive hyperspectral emissions from an object (left) which manifests space-dependent images at multiple wavelengths. Each of these wavelength-dependent images is point multiplied by a binary spatial coded aperture. A dispersive element then causes a wavelength-dependent translation in one dimension. The final 2D CASSI measurement corresponds to summing all of the wavelength-dependent data at a given spatial pixel. (b) The sum of space-dependent pixels may be interpreted as summing a “sheared” coded mini-datacube.

datacube through which \( M \) is constituted, there is a spatially sheared set of voxels from the original datacube that contribute toward \( y_i \) (see Figure 1); let this sheared subset of voxels define a vector \( x_i \in \mathbb{R}^M \), where \( M = N_\lambda d^2 \). Further, we may consider all possible (overlapping) \( d \times d \) contiguous patches of pixels in \( M \), yielding the set of measurement vectors \( \{ y_i \}_{i=1,N} \), with corresponding sheared mini-datacubes \( \{ x_i \}_{i=1,N} \).

We model each \( x_i \) in terms of a dictionary \( x_i = Dc_i + \nu_i \), with \( c_i \) sparse and \( \| \nu_i \|_2 \ll \| x_i \|_2 \). Further, we may express \( y_i = \Phi_i c_i + \epsilon_i \), with \( \Phi_i = \Sigma_i D \) and with \( \epsilon_i \) as defined in the Introduction. With the
CASSI code design, each $\Sigma_i$ is a known sparse binary vector, and the dependence on $i$ is naturally manifested by the CASSI spatially-dependent coded aperture and wavelength-dependent shifts.

We consider the importance of the two key components of the CASSI design: (a) wavelength-dependent shifts (dispersion) and (b) the coded aperture. Concerning (b), if there is no coded aperture, then the projections $\Sigma_i$ are independent of index $i$. It was proven in [22] that the effectiveness of blind CS is significantly enhanced if $\Sigma_i$ changes with $i$. Concerning (a), if there is no dispersion, the measurement $M$ would have a form like the original code, with data entirely absent at spatial locations at which the code blocks photons. Further, at the points at which photons are not blocked by the code, all spectral bands at a given spatial location are simply added to constitute the measurement. This implies that all pixels in $M$ at which non-zero data are measured correspond to the same type of projection measurement (with no spatial dependence to the projection measurement), which the theory in [22] indicates is detrimental to blind-CS performance. Through the joint use of a coded aperture and dispersion, each $\Sigma_i$ has a unique form across each $d \times d$ spatial patch and as a function of wavelength, as encouraged in [22] (i.e., the $\{\Sigma_i\}_{i=1,N}$ have spatial and spectral variation as a function of $i$). The importance of these features of the CASSI measurement are discussed further in Section III-D, when discussing computations.

C. Multi-frame CASSI

The compression rate of the CASSI system as discussed above is $N_\lambda : 1$, as there is a single image $M$ measured, from which the goal is to recover $N_\lambda$ spectral bands, each of the same spatial extent as $M$. In [30] the authors devised a means by which the compression rate can be diminished (with the richness of measured data enhanced), through the measurement of $T$ images $\{M_t\}_{t=1,T}$, where each $M_t$ is measured in the same basic form as described above. To implement this physically, the camera is placed on a piezoelectric translator, allowing quick translation of the camera to $T$ different positions relative to the scene being measured. Since the scene is fixed (or changes slowly relative to the piezoelectric translations), the $T$ snapshots effectively yield $T$ different coded projections on a given hyperspectral datacube (while the code is the same for all $T$ measurements, it is shifted to different positions with respect to the scene being measured). Each of the $T$ images, $\{M_t\}_{t=1,T}$, is divided into patches of the form $\{y_{it}\}_{i=1,N; t=1,T}$, which are analyzed as discussed above, effectively increasing the quantity of data available for inversion. Multi-frame CASSI has a compression rate of $N_\lambda : T$. 
III. BAYESIAN BLIND CS INVERSION

A. Basic model

Beta process factor analysis (BPFA) is a non-parametric Bayesian dictionary learning technique that has been applied for denoising and inpainting of grayscale and RGB images [27], and it has also been utilized for inpainting hyperspectral images [32] with substantial missing data. The beta process is coupled with a Bernoulli process, to impose explicit sparseness on the coefficients \( \{c_i\}_{i=1,N} \). Specifically, consider the representation

\[
y_i = \sum_i D c_i + \epsilon_i, \quad c_i = s_i \cdot z_i, \quad \epsilon_i \sim \mathcal{N}(0, \frac{1}{\gamma_\epsilon}I_M), \quad s_i \sim \mathcal{N}(0, \frac{1}{\gamma_s}I_K)
\]

(1)

where \( z_i \in \{0, 1\}^K \), symbol \( \cdot \) again represents the Hadamard vector product, and \( I_M \) denotes the \( M \times M \) identity matrix. To draw sparse binary vectors \( \{z_i\}_{i=1,N} \), consider

\[
z_{ik} \sim \text{Bernoulli}(\pi_k), \quad \pi_k \sim \text{Beta}(a_\pi/K, b_\pi(K - 1)/K), \quad d_k \sim f(d)
\]

(2)

with the prior \( f(d) \) discussed below; \( \pi_k \) defines the probability with which dictionary element \( d_k \) is used to represent any of the \( x_i \). In the limit \( K \to \infty \), note that for finite \( a_\pi \) and \( b_\pi \) each draw from \( \text{Beta}(a_\pi/K, b_\pi(K - 1)/K) \) is favored to be near zero, implying that it is likely that most \( \pi_k \) will be negligibly small, and most dictionary elements \( \{d_k\}_{k=1,K} \) are unlikely to be utilized when representing \( \{x_i\}_{i=1,N} \). One may show that the number of non-zero components in each \( z_i \) is drawn from \( \text{Poisson}(a_\pi/b_\pi) \), and therefore although the number of dictionary elements \( K \) goes to infinity, the number of dictionary elements used to represent any \( x_i \) is finite (i.e., \( \|c_i\|_0 \) is finite). Gamma priors are placed on \( \gamma_\epsilon \) and \( \gamma_s \), \( \gamma_s \sim \text{Gamma}(a_s, b_s) \) and \( \gamma_\epsilon \sim \text{Gamma}(a_\epsilon, b_\epsilon) \), with hyperparameter settings discussed when presenting results.

Concerning the prior \( f(d) \) on the columns of \( D \), we wish to impose the prior belief that the hyperspectral datacube is likely (but not required) to vary smoothly as a function of spatial location and wavelength. We therefore draw

\[
d_k \sim \mathcal{N}(0, \Omega)
\]

(3)

where \( \Omega \) is an \( M \times M \) covariance matrix. The form of \( \Omega \) defines the correlation structure imposed on each \( d_k \). The following construction has proven effective in the context of the hyperspectral data considered here. Recall that each \( d_k \) is used to expand/represent a sheared mini-datacube \( x_i \), i.e., \( x_i = \sum_{k=1}^K c_{ik} d_k + \nu_i \), where \( c_i = (c_{i1}, \ldots, c_{iK})^T \) is sparse and \( \|\nu_i\|_2 \ll \|x_i\|_2 \). Let \( r_j \in \mathbb{R}^2 \) represent the spatial location and \( \lambda_j \) the wavelength of the \( j \)th component of \( x_i \). A distance is defined between
components $j$ and $j'$ of $x_i$, as

$$\ell(j, j') = \| r_j - r_{j'} \|^2 + \beta(\lambda_j - \lambda_{j'})^2$$

(4)

and therefore $\ell(j, j')$ characterizes the weighted spatial-spectral difference between components $(r_j, \lambda_j)$ and $(r_{j'}, \lambda_{j'})$ of any $x_i$. The goal is to impose through $\Omega$ that if $\ell(j, j')$ is small, then the corresponding components of $x_i$ should be correlated. The $(j, j')$ component of $\Omega$ is defined here as

$$\Omega(j, j') = \exp\left[\frac{-\ell(j, j')}{2\sigma^2}\right]$$

(5)

We discuss the setting of $\beta$ and $\sigma$ when presenting results. Other forms for the definition of $\ell(j, j')$ are clearly possible, with the one considered here an example means of linking correlation in the dictionary element to spatial-spectral proximity.

B. Multi-frame CASSI

Assume we measure $T$ frames of CASSI measurements, $\{M_t\}_{t=1,T}$. Each of these images can be represented in terms of a set of overlapping $d \times d$ patches, as above, and therefore we manifest $T$ different projection measurements for each underlying $x_i$. Specifically, for $x_i$ we perform measurements

$$y_{it} = \Sigma_{it} D(s_i \cdot z_i) + \epsilon_{it}$$

(6)

where $\Sigma_{it}$ represents the CASSI projection matrix for measurement $t$ of $x_i$. Therefore, the multiframe CASSI design [30] allows multiple classes of projection measurements on the same $x_i$, substantially enhancing robustness for inference of $D$ and $\{c_i\}_{i=1,N}$, recalling $c_i = s_i \cdot z_i$. The priors within the Bayesian blind-CS formulation are exactly as elucidated in the previous subsection, but now a given $c_i$ is essentially inferred via multiple $\{\Sigma_{it}\}_{t=1,T}$.

C. Relationship to previous models

The basic construction proposed here may be related to other models proposed in the CS and dictionary learning communities. To see this, note that for multi-frame CASSI the posterior density function of model parameters may be represented as

$$p(\{D, \{s_i\}, \{z_i\}, \gamma_s, \gamma_e, \{\pi_k\}, \{\epsilon_{it}\}) \propto \text{Gamma}(\gamma_s|a_s, b_s)\text{Gamma}(\gamma_e|a_e, b_e)$$

$$\times \prod_{i,t} N(y_{it}|\Sigma_{it} D(s_i \cdot z_i), \frac{1}{\gamma_{it}}) \prod_{i,k} N(s_{ik}|0, \gamma_{ik}^{-1})\text{Bernoulli}(z_{ik}|\pi_k)$$

$$\times \prod_{k} N(d_k|0, \Omega)\text{Beta}(\pi_k|a_\pi/K, b_\pi(K-1)/K)$$

(7)
The log of the posterior may therefore be expressed as

\[
-\log p(\{D_i, \{s_i\}, \{z_i\}, \gamma_s, \gamma_\epsilon, \{\pi_k\}\} | \{y_{it}\}) = \tag{8}
\]

\[\frac{\gamma_\epsilon}{2} \sum_{i,t} \|y_{it} - \Sigma_{it} D(s_i \cdot z_i)\|^2 + \frac{1}{2} \sum_k d_k^T \Omega^{-1} d_k \tag{9}\]

\[+ \log \text{Gamma}(\gamma_s | a_s, b_s) + \log \text{Gamma}(\gamma_\epsilon | a_\epsilon, b_\epsilon) \tag{10}\]

\[+ \frac{\gamma_\epsilon}{2} \sum_{i,t} s_{ik}^2 + \sum_k \log \text{Beta}(\pi_k | a_{\pi}/K, b_{\pi}((K-1)/K)) + \sum_{i,k} \log \text{Bernoulli}(z_{ik} | \pi_k) \tag{11}\]

In the work considered here we will seek an approximation to the full posterior, via Gibbs sampling, as discussed in the next subsection. However, there is much related work on effectively seeking a point approximation for the model parameters, via a maximum a posteriori (MAP) solution, corresponding to inferring model parameters that minimize (8).

The two terms in (9) are widely employed in optimization-based dictionary learning (see for example [33]–[38], and the references therein). The first term in (9) imposes an $\ell_2$ fit between the model and observed data $\{y_{it}\}$, and the second term imposes regularization on the dictionary elements $\{d_k\}_{k=1,K}$, which constitute the columns of $D$. For the special case in which $\Omega = I_m$, the second term in (9) reduces to $\frac{1}{2} \sum_k \|d_k\|^2$, which corresponds to widely employed $\ell_2$ regularization on the dictionary elements. The term $\log \text{Gamma}(\gamma_\epsilon | a_\epsilon, b_\epsilon)$ effectively imposes regularization on the relative importance of the two terms in (9), via the weighting $\gamma_\epsilon$. The terms in (11) impose explicit sparsity on the weights $c_i = s_i \cdot z_i$, and $\frac{\gamma_s}{2} \sum_{i,k} s_{ik}^2 = \frac{\gamma_s}{2} \sum_i \|s_i\|^2$ again imposes $\ell_2$ regularization on $\{s_i\}$.

The sparsity manifested via (11) is the most distinctive aspect of the proposed model, relative to previous optimization-based approaches [33]–[38]. In that work one often places shrinkage priors on the weights $c_i$, via $\ell_1$ regularization $\gamma_s \sum_i \|c_i\|_1$; in such an approach all the terms in (11) are essentially just replaced with $\gamma_s \sum_i \|c_i\|_1$. So an optimization-based analog to the proposed approach is of the form [36]

\[
\gamma_\epsilon \sum_{i,t} \|y_{it} - \Sigma_{it} D(s_i \cdot z_i)\|^2 + \sum_k d_k^T \Omega^{-1} d_k + \gamma_s \sum_i \|c_i\|_1 \tag{12}\]

In optimization-based approaches one seeks to minimize (12), and the parameters $\gamma_\epsilon$ and $\gamma_s$ are typically set by hand (e.g., via cross validation). Such approaches may have difficulties for blind CS, for which there may not be appropriate training data to learn $\gamma_\epsilon$ and $\gamma_s$ a priori. One advantage of the Bayesian setup is that we infer posterior distributions for $\gamma_\epsilon$ and $\gamma_s$, along with similar posterior estimates for all model parameters (there is no cross-validation).

We also note that there are other ways to constitute sparsity of $\{c_i\}$. Specifically, all of the terms in (11) may be replaced by a shrinkage prior. Letting $c_{ik}$ denote the $k$th component of $c_i$, we may draw
\( c_{ik} \sim \mathcal{N}(0, \alpha_{ik}^{-1}) \), and place a gamma prior separately on each of the \( \alpha_{ik} \). Related priors have been considered in [39]–[41]. We choose to employ the beta-Bernoulli construction because it imposes that components of \( c_i \) are exactly zero (not just negligibly small), and via the beta-Bernoulli construction [42], explicit priors are placed on the number of non-zero components of each \( c_i \). However, this is essentially a modeling choice, and the methods in [39]–[41] may also be employed to impose sparsity (or near sparsity) on \( \{c_i\} \).

D. Gibbs Sampling

Inference is performed by Gibbs sampling, which consists of iteratively sampling from the conditional distribution of each parameter, given the most recent values of the remaining ones [43]. The conditional distributions given below can all be derived using standard formulae for conjugate priors [44]. In the following formulae, the symbol ‘−’ refers to ‘all other parameters except the one being sampled’.

**Sampling \( d_k \):**

\[
p(d_k|\cdot) \propto \prod_{i,t} \mathcal{N}(y_{it}|D(s_i \cdot z_i), \frac{1}{\gamma_{\epsilon}}I_m) \mathcal{N}(d_k|0, \Omega),
\]

\[
p(d_k|\cdot) \sim \mathcal{N}(d_k|\mu_{dk}, \Sigma_{dk}),
\]

\[
\Sigma_{dk} = (\Omega^{-1} + \gamma_{\epsilon} \sum_{i,t} s_{ik}^2 z_{ik}^2 \Sigma_{it}^T \Sigma_{it})^{-1}
\]

\[
\mu_{dk} = \gamma_{\epsilon} \Sigma_{dk} \sum_{i,t} z_{ik} s_{ik} \Sigma_{it}^T y_{(i,t,-k)},
\]

\[
y_{(i,t,-k)} = y_{it} - \Sigma_{it} D(s_i \cdot z_i) + \Sigma_{it} s_{ik} z_{ik} d_k.
\]

The expression for sampling \( \Sigma_{dk} \) (and hence, \( \mu_{dk} \)) reveals the importance of having projections that vary spatially. While the matrices \( \Sigma_{it}^T \Sigma_{it} \) are of low rank, their (weighted) summation will have full rank, assuming (a) that there are sufficiently many patches for which \( z_{ik} = 1 \), and (b) that the \( \{\Sigma_{it}\} \) vary spatially and employ (non-zero weights) different components of the mini-datacube.

**Sampling \( z_{ik} \):**

\[
p(z_{ik}|\cdot) \sim \text{Bernoulli}(\frac{p_1}{p_1 + p_0}),
\]

\[
p_1 = \pi_k \exp \left( -\gamma_{\epsilon} \left( \sum_{i} s_{ik}^2 d_k^T \Sigma_{it} \Sigma_{it} d_k - 2s_{ik} d_k^T \Sigma_{it}^T y_{(i,t,-k)} \right) \right),
\]

\[
p_0 = 1 - \pi_k.
\]
Sampling $s_{ik}$:

$$p(s_{ik}|-) \sim \mathcal{N}(s_{ik}|\mu_{sik}, \sigma_{sik}), \quad (21)$$

$$\sigma_{sik} = (\gamma_s + \gamma_\epsilon z_{ik}^2 d_k^T \Sigma_{it}^T \Sigma_{it} d_k)^{-1}, \quad (22)$$

$$\mu_{sik} = \gamma_\epsilon \sigma_{sik} z_{ik} d_k^T \sum_t \Sigma_{it}^T y_{(i,t,k)}, \quad (23)$$

Sampling $\pi_k$:

$$p(\pi_k|-) \sim \text{Beta}(a_{\pi}/K + \sum_i z_{ik}, b_{\pi}(K-1)/K + N - \sum_i z_{ik}). \quad (24)$$

Sampling $\gamma_s$:

$$p(\gamma_s|-) \sim \Gamma(a_s + \frac{1}{2} K N T, b_s + \frac{1}{2} \sum_i s_i^T s_i). \quad (25)$$

Sampling $\gamma_\epsilon$:

$$p(\gamma_\epsilon|-) \sim \Gamma(a_\epsilon + \frac{1}{2} \sum_{i,t} \|\Sigma_{it}\|_0, b_\epsilon + \frac{1}{2} \sum_{i,t} \|y_{it} - \Sigma_{it} D(s_i \cdot z_i)\|^2). \quad (26)$$

Traditionally, Gibbs sampling is run for many burn-in iterations to allow for mixing, followed by the collection phase [43].

IV. EXPERIMENTAL RESULTS

A. Parameter Settings for BPFA

An encoded image of size $N_x \times N_y$ is divided into $N = (N_x - d + 1)(N_y - d + 1)$ overlapping patches, each of size $d \times d$. When learning the dictionary $D$, which is shared among all $N$ patches, we typically select 10 to 20% of the patches (depending on the size of $N$), selected uniformly at random from the different spatial locations in the acquired image. Since $N$ is typically quite large, it has been found that it is unnecessary to use all $N$ patches from a given image to learn $D$ well. This is because most natural images exhibit a high degree of self-similarity at the level of small patches [31]. The Gibbs sampler yields multiple dictionaries (for each of the collection samples), and the maximum likelihood sample is used to define $D$. Traditionally, MCMC based methods need to be run for several (typically a few thousand) iterations to “burn in”, followed by a collection phase where the obtained samples are averaged to yield a final estimate [43]. However, we observed that the Gibbs sampler yielded excellent results with as few as 30 iterations. This was also observed for the BPFA model for denoising and inpainting applications in [27].
After the dictionary is so learned \textit{in situ} for a given CASSI-measured image, the learned $D$ is then fixed and used to infer $c_i$ for all patches $i$, and from this an estimate to the underlying patch pixel values is $Dc_i$. Since multiple overlapping patches are employed, the final pixel value at each point in the underlying image is represented as the average from all overlapping patches (we also average across collection samples).

We used the same BPFA settings in all experiments, without requiring tuning for specific types of data. We set $K = 32$ and $d = 4$. For a datacube of $N\lambda$ wavelengths, the inferred patches are of size $d^2N\lambda \times 1$. We set $K$ to a relatively small value, to aid computational efficiency; one may make $K$ large and infer the subset of dictionary elements required, at increased computational expense [27], [45] (this was found to be unnecessary). The parameters for the hyperpriors were set as follows: $a_\pi = 0, b_\pi = \frac{N}{2}$, $a_\varepsilon = b_\varepsilon = a_\alpha = b_\alpha = 10^{-6}$. The parameters of the GP prior for the dictionary elements were set to $\sigma = 5, \beta = 1$. These are standard parameter settings, \textit{i.e.}, they were not tuned. Moreover, we have empirically observed that \textit{in situ} dictionary learning on each new CASSI image was necessary to obtain good inversion results (the data-dependent dictionaries aided inversion performance).

\section*{B. Comparisons with Other Methods}

BPFA results are compared to the following alternative methods:

1) TwIST (Two-step Iterative Shrinkage/Thresholding) [19]. This algorithm performs a descent on energy function

$$E(x) = \sum_{t=1}^{T} \|y_t - \Sigma_t x\|^2 + \tau TV(x).$$

where $x$ and $\{y\}_{t=1,T}$ are the original and encoded data, respectively. The regularizer $TV(x)$ is the total variation (TV) of the underlying 3D spatio-spectral cube $x$, defined as

$$TV(x) = \sum_{\lambda} \sum_{i_y,i_x} \sqrt{(x(i_y+1,i_x,\lambda) - x(i_y,i_x,\lambda))^2 + (x(i_y,i_x+1,\lambda) - x(i_y,i_x,\lambda))^2}$$

where $i_y, i_x$ index discrete spatial coordinates, and $\lambda$ indexes wavelengths. The parameter $\tau$ is a tradeoff between the likelihood and the regularizer, and depends on the noise variance. This algorithm was used for the CASSI inversion in [30]. We performed experiments with different values of $\tau$ and wherever possible picked the value of $\tau$ that yielded the least mean squared error (MSE) with respect to the ground truth (when available). Generally, we observed that this “optimal” $\tau$ was close to 0.3 (the scale of the original data was [0,1]).
2) KSVD [26]. Tuned here for the multi-frame CASSI problem \((T > 1)\), KSVD seeks to minimize

\[
E(D, S = [s_1|s_2|...|s_N]) = \sum_{i,t} \| y_{it} - \Sigma \Sigma_{it} D s_i \|^2 \text{ s.t. } \forall i, \|s_i\|_0 \leq T_0
\]

where \(D \in \mathbb{R}^{MN \times K}\) is a dictionary, \(S \in \mathbb{R}^{K \times N}\) is a matrix of dictionary coefficients, and \(T_0\) is a parameter that governs the sparsity of the dictionary codes. In practice the optimization for KSVD proceeds in two phases. Given a fixed dictionary, sparse coding is typically performed using Orthogonal Matching Pursuit (OMP) using either a fixed mean squared error \(e\) (as we do here) or a fixed sparsity level \(T_0\). The dictionary is then updated atom by atom, using an incremental form of the singular value decomposition (SVD) of a carefully defined error matrix [26]. The sparse coding and dictionary update steps are performed in an iterative manner. KSVD requires careful selection of various parameters: the number of dictionary atoms \(K\) and the error \(e\) for OMP. In our experiments, we set \(K = 32\) for the sake of computational speed (and consistency with the BPFA settings) and \(e = 0.002\) (the latter because measurement noise was typically very low).

3) Max-norm matrix factorization (referred to hereafter as MaxNorm) [46]. This is a state-of-the-art matrix factorization method, which uses the matrix “max-norm” as a regularizer, and has been successful in matrix completion problems. Tuned here for the multi-frame CASSI problem (again, this mean \(T > 1\) CASSI images are performed per hyperspectral datacube), this technique seeks to minimize

\[
E(D, S = [s_1|s_2|...|s_N]) = \sum_{i,t} \| y_{it} - \Sigma \Sigma_{it} D s_i \|^2 \text{ s.t. } (29)
\]

\[
\|D\|_{2,\infty} \leq B, \|S^T\|_{2,\infty} \leq B
\]

where \(D \in \mathbb{R}^{MN \times K}\) is a dictionary and \(S \in \mathbb{R}^{K \times N}\) is a matrix of dictionary coefficients. The max-norm of matrix \(D\) is defined as \(\|D\|_{2,\infty} = \max_j \sqrt{\sum_k D_{jk}}\). The max-norm implicitly imposes an upper bound \(B\) on the maximum absolute value of any pixel from the underlying image. In our experiments, we set \(B = 1\) as the original data had elements in the range \([0,1]\), and \(K = 32\) (consistent with the BPFA and KSVD settings). The matrices \(D\) and \(S\) were inferred using stochastic gradient descent with a dynamic step-size, on mini-batches of 1500 patches. Performing the gradient descent usually violates the max-norm constraints, even when starting from a feasible point, and therefore it was necessary to enforce the constraints by projection of the updated variables onto the constraint set. This was done by rescaling those rows of \(D\) and columns of \(S\) whose norms exceeded \(\sqrt{B}\), in order to make those norms equal to \(\sqrt{B}\) (see Section 3 of [46]). The step-size...
for the descent was chosen to be the maximum value in the interval (0,2], which decreased the energy $E(D,S)$ *after* imposition of the constraints.

For TwIST and KSVD, we used software provided online\(^1\), and suitably modified them for the CASSI problem. For MaxNorm, we used our own implementation of the algorithm described in [46]. As in the BPFA computations, for MaxNorm and KSVD we used only a small fraction (10 to 20\%) of the overlapping patches for dictionary learning. All patches were sparse-coded and their reconstructions were averaged to yield the final image.

**C. Computation time**

We have implemented the BPFA algorithm in C. Reconstruction of a $1000 \times 700 \times 24$ dataset (24 wavelengths) using 8 frames takes 28 minutes on a 3.4 GHz AMD Phenom II processor. This includes about 7-10 minutes for dictionary learning. The computational requirements of KSVD were similar to BPFA, while TwIST yielded the fastest reconstructions. In our experience, MaxNorm was computationally the most expensive method, as it required an adaptive selection of the step-size in gradient descent (taking care to ensure that the energy does not increase after projection onto the constraint set), and it typically required a large number of iterations to converge. In our experiments, we set an upper limit of 70 on the number of iterations of gradient descent in the MaxNorm method; our experiments also revealed that these many iterations were necessary to obtain a good result.

**D. Reconstruction Quality Metrics**

The MSE or the PSNR (peak signal to noise ratio) is the most popular measure to evaluate the quality of a reconstruction, if the underlying ground truth is known. The PSNR is however often not fully representative of image quality in a perceptual sense [47]. Hence we compute two other measures to quantify reconstruction quality - (a) the high frequency PSNR or HF-PSNR (defined below) and (b) the Structural Similarity Index (SSIM) from [47]. Textured regions in an image contain significant high frequency information, which some techniques like TwIST tend to smooth out. However, the PSNR is a global quality measure which does not quantify errors in individual spatial frequency bands. Hence, it is useful to calculate the MSE between the magnitudes of the higher spatial frequency Fourier coefficients of every channel of the true and reconstructed images. Given a reference image $I$ and an estimate $J$, this

\(^1\)http://www.lx.it.pt/~bioucas/TwIST/TwIST.htm, http://www.cs.technion.ac.il/~elad/software/

May 6, 2012
MSE (averaged over the spectral channels) is given by:

\[
e = \frac{1}{N|\mathcal{H}|} \sum_{\lambda} \sum_{h \in \mathcal{H}} \|\hat{I}_{\lambda,h} - \hat{J}_{\lambda,h}\|^2
\]

(30)

where \(\hat{I}_{\lambda,h}\) refers to the magnitude of the \(h^{th}\) Fourier coefficient from the spectral channel \(\lambda\) of the image \(I\), and \(\mathcal{H}\) refers to a set of higher frequencies. In our experiments, we divided the frequency plane into equal-sized bins denoted as \(b_{i,j}\), where \(i\) and \(j\) are bin indices along the two axes, and \(1 \leq i \leq 32\) and \(1 \leq j \leq 32\). The set \(\mathcal{H}\) contained frequencies from the bin corresponding to the highest frequencies from both axes, i.e. from \(b_{32,32}\). The corresponding PSNR value, computed as \(10 \log_{10} \frac{I_m \times f}{e}\) where \(I_m := \max_{\lambda,h \in \mathcal{H}} \hat{I}_{\lambda,h}\), is hereafter referred to as HF-PSNR.

The measure SSIM has been proposed in the recent image processing literature [47] to quantify full-reference grayscale image quality. Its values always lie in the range \([0,1]\), and it is known to represent perceptual image quality better than MSE or PSNR [47]. The SSIM is calculated by adding up values of a local index of similarity between corresponding patches from the two images being compared. The local similarity index is based on three quantities: the similarity between the mean intensity value of the two patches, the similarity between the standard deviation of the intensities in the two patches, and the cross-correlation between the two patches. We compute the SSIM values between every channel of the true and reconstructed image, and calculate an average of these per-channel SSIM values. The SSIM values are computed using software provided online\(^2\) and using the default patch-size of \(11 \times 11\).

### E. Results on Synthetic Encodings of Phantoms and Natural Scenes

We first present reconstruction results on three synthesized CASSI datasets, all encoded with a binary coded mask (with values defined Bernoulli(0.5)) as employed in the real CASSI experiments discussed below. These experiments employed the forward model (dispersion) characteristic of the actual CASSI system, and zero-mean white Gaussian noise was added to constitute the final simulated data (with standard deviation equal to 1% of the maximum amplitude in the hyperspectral datacube). This low noise is characteristic of the actual CASSI system, and therefore we do not consider high-noise simulations here.

In the simulated CASSI measurements, and in the physical measurements discussed in Section IV-G, wavelengths from 450-650 nm are considered, except for a phantom dataset for which wavelengths from 500-2000 nm are considered. A “frame” of CASSI images is defined by one of the \(T\) two-dimensional images.

\(^2\)https://ece.uwaterloo.ca/~z70wang/research/ssim/
CASSI measurements discussed in Section III-B. For $N_\lambda$ wavelengths in the inferred datacube, each spatial image at a particular wavelength is termed a “channel”, and therefore we refer to an $N_\lambda$ channel CASSI datacube. In an $N_\lambda$ measurement, the channels $1, \ldots, N_\lambda$ are indexed from smallest to largest measured wavelength. The $T$ different frames are manifested via translations of up to 20 $\mu$m (implemented in practice with a piezo system), which corresponds to a translation of up to 24 pixels. See [30] for details on how the translations and multiple frames are measured in practice.

The first dataset of size $512 \times 512 \times 50$ is a synthetically created phantom. The phantom consists of 17 regions - 16 circularly shaped non-overlapping regions and one region corresponding to a background. Within each region, the spectral patterns are constant. The spectral patterns used here correspond to those recorded from a variety of drugs (the reference data were measured using an independent camera, and will be made available for comparisons). The original spectral patterns for each drug consisted of 1300 wavelengths, out of which we uniformly sampled 50, taking care to preserve the overall (macroscopic) shape of the original pattern. We refer to these data henceforth as the Phantom data. The second dataset, of size $1021 \times 703 \times 25$, is an image of a photograph of birds, observed at 25 wavelengths (henceforth referred to as the Birds data). The Birds data were acquired using a pushbroom imager built on CASSI hardware. The coded aperture in the CASSI system was replaced by a vertical slit (of effective width 1 pixel). This system acquires a 2D space-wavelength slice of the 3D data cube in each time step. The full (noncompressive) data cube for a static object is acquired by translating the slit along the dispersion axis. This optical system is described in [48]. The third dataset is of size $820 \times 820 \times 31$ (31 wavelengths). It was obtained online\(^3\), from a hyperspectral image database at the University of Manchester [49]; these data are henceforth referred to as the Objects data. Sample encoded images of all three datasets, as well as colored (RGB) pictures of the latter two scenes, are shown in Figure 2. The RGB images are not spatially aligned with the coded measurements and are provided only for reference. In fact, for the Objects data, the RGB image even shows a slightly different part of the scene as compared to what was imaged with the hyperspectral sensor.

We display the reconstruction results by plotting images corresponding to a subset of the wavelengths from the reconstructed hyperspectral image. The dominant wavelengths in the datasets in this paper fall within the visible spectrum (except for the Phantom data). Hence, the data at wavelength $\lambda$ (in all datasets except the Phantom data) can plotted using a specifically chosen color in the RGB format. This color is obtained using a “color matching function” which takes two inputs: the wavelength $\lambda$ and a signal

\(^3\)http://personalpages.manchester.ac.uk/staff/david.foster/Hyperspectral\_images\_of\_natural\_scenes\_02.html
TABLE I
RECONSTRUCTION QUALITY MEASURES (PSNR, HF-PSNR, SSIM) FOR 3-FRAME ($T = 3$) RECONSTRUCTION USING BPFA, KSVD, MAXNORM AND TWIST.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Quality metric</th>
<th>BPFA</th>
<th>KSVD</th>
<th>MaxNorm</th>
<th>TwIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phantom ($512 \times 512 \times 45$)</td>
<td>PSNR</td>
<td>30.9</td>
<td>17</td>
<td>23.76</td>
<td>22.36</td>
</tr>
<tr>
<td></td>
<td>HF-PSNR</td>
<td>41.61</td>
<td>28</td>
<td>31.3</td>
<td>30.52</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.934</td>
<td>0.6</td>
<td>0.84</td>
<td>0.8388</td>
</tr>
<tr>
<td>Birds (synthetic) ($1000 \times 703 \times 25$)</td>
<td>PSNR</td>
<td>30.8</td>
<td>17</td>
<td>25</td>
<td>29.33</td>
</tr>
<tr>
<td></td>
<td>HF-PSNR</td>
<td>45.11</td>
<td>26.8</td>
<td>27.51</td>
<td>41.31</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.95</td>
<td>0.4</td>
<td>0.81</td>
<td>0.89</td>
</tr>
<tr>
<td>Scene with objects ($820 \times 820 \times 31$)</td>
<td>PSNR</td>
<td>27.04</td>
<td>25.18</td>
<td>19.89</td>
<td>26.74</td>
</tr>
<tr>
<td></td>
<td>HF-PSNR</td>
<td>58.19</td>
<td>55.53</td>
<td>51.04</td>
<td>56.28</td>
</tr>
<tr>
<td></td>
<td>SSIM</td>
<td>0.824</td>
<td>0.789</td>
<td>0.6</td>
<td>0.759</td>
</tr>
</tbody>
</table>

magnitude value, and outputs an RGB tuple. The particular color matching function we used is CIE 1964 10-degree (International Commission on Illumination), a convention commonly used in color display\textsuperscript{4} [50]. For the Phantom dataset, the results are displayed using a simple grayscale map. All hyperspectral datacubes are plotted on a common scale from 0 to 1. Note that images at different wavelengths are not individually rescaled. Although the birds dataset includes one violet/indigo colored bird, the wavelengths corresponding to these colors ($\leq 450$ nm) were masked off with a band-pass filter during actual acquisition of the underlying data used to synthesize CASSI measurements.

Reconstruction results for the three datasets (for a subset of the wavelengths) are shown in Figures 3, 4 and 5, respectively, alongside corresponding ground-truth and reconstruction PSNRs. The PSNR, HF-PSNR and SSIM values are all presented in Table I. In case of all three measures, the higher values correspond to better image quality, and we observe that BPFA always produces significantly higher values than other methods (this is especially true for the SSIM and HF-PSNR).

For the Phantom data (see sample encoded image in Figure 2), we observed that BPFA and MaxNorm preserved the spectral properties of the data, which TwIST and KSVD were unable to, as evident from Figure 3. For instance, the signal magnitude in the first and last two channels is very low, a constraint which KSVD and (to a lesser extent) TwIST fail to satisfy. For several regions, BPFA and MaxNorm preserve the spectral variation beautifully. This can be observed from the comparative spectral plots in

\textsuperscript{4}http://cvrl.ioo.ucl.ac.uk/cmfs.htm
Fig. 2. Top row, first image: Example CASSI measurement for the Phantom dataset. Top row, last two images: Example CASSI measurement (left) and RGB representation (right) for the Birds dataset. Bottom two images: Example CASSI measurement (left) and RGB representation (right) for the Objects dataset. The RGB representations are not spatially aligned with the coded measurements (especially for the Objects dataset) and are provided for reference only.

Figure 6 – the spectral patterns in this figure were averaged over a $5 \times 5$ neighborhood around points selected from different regions of the phantom. A point to be noted here is that BPFA easily outperformed TwIST on this dataset (which has a relatively larger number of channels than the other datasets presented in this paper) even though the image was a piecewise-constant cartoon, an image model that is favored by TwIST. BPFA outperformed all methods including MaxNorm in terms of all three image quality metrics.

For the Birds dataset, we observed that BPFA preserves the spectral properties of the data much better than the other methods, as seen in Figures 4 (compare to Figure 2) and 7. For instance, the signal magnitude for the first wavelength is actually very low, and hence very little structure is visible in the original image at this wavelength. While the BPFA result is similar to the ground truth, the TwIST and KSVD reconstruction (and to a lesser extent the MaxNorm result) show a considerable amount of (inappropriate) structure at this wavelength. We found that KSVD does not reproduce the variation in spectral profiles across wavelengths, and tends to produce nearly uniform spectral responses. This behavior has been observed earlier for color (RGB) images in [28], and the authors used a special weighting scheme (designed for RGB) inside the OMP sparse coding to overcome this problem. However, here we consider diverse spectral patterns over several wavelengths, and devising a similarly appropriate scheme is beyond the scope of this paper. MaxNorm preserves spectral patterns well, however it tends to produce noisy artifacts spatially. TwIST tends to erase subtle textures (a well-known problem with the TV regularizer, which assumes a piece-wise constant intensity model for natural images). Further, at various various
wavelengths, TwIST produces artifacts. BPFA, on the other hand, preserves the spatial textures well. In the bottom row of Figure 4, we show a zoomed-in version of a small portion of the 19th wavelength of the bird image (denoting the first wavelength the smallest considered), and its reconstruction using BPFA, KSVD, MaxNorm and TwIST. One can see that MaxNorm produces noisy grainy artifacts, while TwIST tends to erase subtle textures present on the head and below the eye of the bird. The BPFA result is devoid of these artifacts. Moreover, the BPFA result produced a higher PSNR value than other methods. In Figure 7, we also present sample spectral plots at a few points – the spectral patterns are averaged over a small spatial neighborhood of $5 \times 5$. One can observe that the BPFA plots are closer to the ground truth.

For the Objects data (see Figures 2 and 5), which has greater spectral diversity than the Birds dataset, we make the following observations. BPFA and TwIST are able to recover the spectral properties well, although BPFA produced a slightly higher PSNR. However, the BPFA result preserves some object boundaries better than the TwIST result, as shown in Figure 8; observe how TwIST blurs out the boundaries of the robot and the letter ‘P’, which BPFA preserves. The KSVD result shows some errors in recovery of the spectral properties; for instance, it produces undesirably high intensities for the maroon rucksack and the red-colored block in the ‘violet’, ‘blue’ and ‘green’ channels - see rows 1, 2 and 3 (channels 2, 4 and 6) of Figure 5. Similarly, although the MaxNorm algorithm performed well on the Birds dataset, it often failed to preserve spectral variations on the Objects data. For instance, it produces a much stronger intensity on the red vase for wavelength 500 nm (in row 3 of Figure 5) or the maroon rucksack for wavelength 660 nm (in row 5 of Figure 5), as compared to the ground truth (see Figure 2). Here again, BPFA produced a higher PSNR value than other methods.

We studied the effect of the GP prior on BPFA dictionary elements as follows. For the Object data with $T = 3$ CASSI frames, we used different numbers of patches (of size $4 \times 4$) in dictionary learning, from 5000 to 50,000 per frame (out of a total of $6.9 \times 10^5$ patches). Keeping all other parameters the same, we performed the reconstruction with and without the GP prior ($\sigma = 5$) on the dictionary elements. As observed in Table II, the reconstruction PSNRs were better with the GP prior than without. The relative advantages of a good GP prior are the strongest with smaller sample sizes. With increasing sample sizes, the GP prior may not be not be as important (but this comes at increased computational cost).

F. Simultaneous Reconstruction and Inpainting

For the Birds dataset, we performed an additional experiment using BPFA: we deliberately removed (i.e., set to zero) 70% of the pixels from the encoded CASSI images, with these removed pixels selected at
random. This implies only 30% of the data need be measured, consistent with related studies with BPFA applied to RGB and hyperspectral data [27]. We reconstructed the original hyperspectral datacube, after suitably modifying the forward model, i.e., by nullifying appropriate entries from the $\Sigma_{it}$ matrices. The results for this experiment are shown in Figure 9. The reconstruction PSNR does reduce from 30.8 (based upon all CASSI data measured) to 28.85 (with 30% measured per frame). However, despite the reduced data, the fine textures on the wings of the birds, as well as the spectral patterns, are generally reconstructed well. The reduced measurements does introduce noisy artifacts, however these can be discerned only upon careful zooming. While one may not wish to utilize such a reduced number of CASSI measurements in practice, these experiments demonstrate that the inpainting capabilities of BPFA considered in [27] generalize to CASSI measurements.

G. Results on Real Data

We now present results on actual data acquired by the CASSI system. We again consider the Birds dataset, of size $1021 \times 703 \times 24$, and a dataset of size $404 \times 400 \times 23$ that images holly leaves and fruits (referred to as Holly data). The measured CASSI snapshot images are shown in Figure 10. The CASSI reconstructions (based on the real measure data) presented in this section are compared to the Birds data (mentioned previously), which were acquired using a different (and non-compressive) hyperspectral imaging setup (see Section IV-E for details). During actual measurement of the Birds scene by the CASSI system, a band-pass filter was used, which blocked wavelengths beyond 680 nm (in addition to the filter which blocked wavelengths below 450 nm). Hence the Bird data here has only 24 channels (as against 25 channels in Section IV-E), and the signal intensity in the 24th channel (wavelength 700 nm) is very low. For this dataset, we report results on BPFA with the number of frames set to $T = 4$ and $T = 12$. These results are displayed in Figure 11 alongside the independent “ground truth” measurement and a 24-frame reconstruction using TwIST. The ground truth image was collected using a slightly different physical

---

**TABLE II**

<table>
<thead>
<tr>
<th># samples for dictionary learning</th>
<th>PSNR with GP</th>
<th>PSNR without GP</th>
</tr>
</thead>
<tbody>
<tr>
<td>5000</td>
<td>24.82</td>
<td>22.68</td>
</tr>
<tr>
<td>10000</td>
<td>25.99</td>
<td>23.81</td>
</tr>
<tr>
<td>20000</td>
<td>26.45</td>
<td>24.66</td>
</tr>
<tr>
<td>50000</td>
<td>27.2</td>
<td>25.2</td>
</tr>
</tbody>
</table>
Fig. 3. Reconstruction results for the Phantom data using 3 frames - for channels 1, 11, 17, 22, 33, 38 and 44 (corresponding to wavelengths 501, 801, 981, 1131, 1461, 1611 and 1791 nm). From left to right in each row - Col. 1: true image, Col. 2: BPFA, Col. 3: KSVD, Col. 4: MaxNorm, and Col. 5: TwIST.

setup, and hence is slightly misregistered with the CASSI reconstructions; it has an intensity profile that resembles the underlying scene that was measured by the CASSI system, but is not identical to it. Therefore, the PSNR values computed with reference to the ground truth image (after a registration over translation parameters) are only an approximation. As expected, we see a distinct visual improvement
TABLE III
PSNR VALUES FOR 4-FRAME RECONSTRUCTION OF DIFFERENT DATASETS USING BPFA, KSVD, MAXNORM AND TWIST.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>BPFA</th>
<th>KSVD</th>
<th>MaxNorm</th>
<th>TWIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-birds:Ground-truth image after coarse registration</td>
<td>16.2</td>
<td>10.2</td>
<td>14.82</td>
<td>14.15</td>
</tr>
</tbody>
</table>

in the BPFA results when the number of frames is increased from 4 to 12; this holds true for spectral properties as well as quality of recovery of texture patterns (zoom into Figure 11).

TwIST tends to incorrectly reconstruct some detailed structure in channels 1 and 24, absent in the ground truth. In fact, this artifact is present even in the \( T = 24 \) frame TwIST reconstruction, whereas the BPFA result with a smaller number of frames does not exhibit this artifact. For this dataset, we again observed that KSVD was unable to recover spectral properties, whereas MaxNorm produced a good reconstruction, albeit with a few errors. For instance, it overestimated the intensity of the smaller bird in channels 16 and 19 (rows 6, 7 and 8 of Figure 11). The PSNR values for these results are presented in Table III. Note that, as the ground truth was misregistered, we measured the reconstruction PSNRs after performing a coarse registration over translation in the X and Y direction ranging from -7 to +7.

Results for the Holly dataset are shown in Figure 12. As there is no ground truth available for these data, we report PSNR values with reference to a \( T = 24 \) TwIST reconstruction. BPFA is able to capture important spectral properties of the Holly scene, even with only \( T = 4 \) frames; note how the holly fruits (which are red in color) become brighter as the wavelength increases. As compared to TwIST, we observed that BPFA and MaxNorm (all with \( T = 4 \) frames) produced a better definition of the boundaries between the different fruits, which TwIST tends to blur out. This can be observed in the last row (channel 23) of Figure 12. Although MaxNorm produced a higher PSNR than BPFA for this dataset, we emphasize that it was computationally far more expensive than BPFA (120 minutes with MaxNorm as opposed to 45 minutes for BPFA).

H. Extension to Video

We now present a reconstruction result on real data acquired from a video version of the CASSI system. In the video CASSI system, at each time instant \( 1 \leq t \leq N_t \), a single two-dimensional (2D) coded snapshot of the underlying 3D hyperspectral scene is acquired. The collection of the hyperspectral datacubes at all \( N_t \) time instants can be regarded as a single 4D signal of size \( N_x \times N_y \times N_\lambda \times N_t \). The aim is to reconstruct this 4D signal given a video sequence of 2D encoded snapshots (recovery
of $N_t$ hyperspectral datacubes based on $N_t$ two-dimensional images). Note that the multi-frame CASSI acquisitions discussed above differ from the video CASSI system. In the former, we considered a (nearly) static scene and the different frames corresponded to snapshots acquired using different translations of the same code. In video CASSI, the scene is dynamic and snapshots at all time instants are acquired with the same (non-translated) code. However, we assume that any change in the scene is slow, and hence the scenes at nearby time instants are structurally similar. This additional redundancy can be exploited to afford better quality scene reconstruction (we exploit dynamics of the scene, and therefore do not have to shift the coded aperture).

In previous work [51], the hyperspectral video was reconstructed using a method called ‘NeAREst’. We denote the original 4D signal as $x$, the acquired measurement as $y$ (both in vector form), and the measurement matrix as $\Sigma$. Let the subscript $\lambda$ refer to a slice of the hyperspectral signal at the wavelength $\lambda$. We then have:

$$y = \sum_{\lambda=1}^{N_{\lambda}} \Sigma_{\lambda} x_{\lambda} + \epsilon.$$  \hspace{1cm} (31)

The method NeAREst solves for $x$ by minimizing the following constrained version of the Kullback-Leibler divergence between $y$ and $\Sigma x$:

$$E(x) = \sum_{i=1}^{N_y} \sum_{j=1}^{N_x} y_{ij} \left( \log y_{ij} - \log((\Sigma x)_{ij}) \right) - (y_{ij} - \text{sign}(y_{ij})(\Sigma x)_{ij}) \text{ subject to } \forall i,j \ x_{ij} \geq 0.$$ \hspace{1cm} (32)

While this method has an inbuilt regularizer [51], it does not explicitly account for the inter-dependence or smooth variation between the signal values across wavelengths, as with TwIST [19] or GPSR [29]. Neither does it account for temporal redundancy in slowly moving scenes under (relatively) constant illumination.

We perform reconstruction of the hyperspectral video by accounting for both these properties. The reconstruction is performed in a blind-CS setting, with the dictionary learned from the encoded data using BPFA, just as done for static scenes. While the dictionary learning and sparse coding can be performed independently for every time-frame, this essentially ignores the obvious relationships between adjacent time-frames. It also does not explicitly encode temporal information or the movement of parts of the scene over time. Hence, we learn dictionaries in a 4D space, in terms of spatial, spectral and temporal coordinates. Several previous papers on video reconstruction (albeit not dealing with hyperspectral video) have learned and used dictionaries that incorporate temporal information [37], [52].

The dataset considered here is an acquired video sequence of coded snapshots that image a group of moving fish. We name this as the Fish data hereafter. The Fish data contain of $N_t = 200$ time-frames,
each of size $512 \times 512$ and containing $N_\lambda = 15$ spectral channels. For the reconstruction, we build a space-time-spectral dictionary on patches of spatial size $4 \times 4$, 4 consecutive time instants, and all 15 channels, i.e., each dictionary element has dimensionality 960. A dictionary was built using about 10% of the patches from the encoded data. The BPFA inversion was able to capture the movement of the fish. We compared the BPFA result to that obtained using TwIST. We used TwIST for comparison because it has produced results that are more visually pleasing than GPSR or NeAREst in previous applications [30], [53]. We show some snapshots of the acquired video snapshots, as well as the reconstructions for each frame in Figure 13. The complete acquired video and reconstructions of all 15 channels are available in the supplemental material, that can be downloaded from http://people.ee.duke.edu/~ar219/CASSI/.

V. CONCLUSION

The beta process factor analysis (BPFA) model has been employed for inversion of CASSI hyperspectral compressive measurements. Based on several experiments with synthesized and real compressive measurements, BPFA was demonstrated to generally perform better than other related inversion methods, specifically TwIST, KSVD and MaxNorm. Encouraging results were demonstrated on multi-frame measurement of images (multiply translated coded aperture), and single-frame (fixed coded aperture) video measurements. The BPFA formulation is Bayesian, and inference is performed based upon Gibbs sampling. In practice we have found that a very small number of Gibbs samples are required to obtain high-quality datacube reconstructions. Although it has not been emphasized here, the posterior collection samples may be used to infer uncertainty (e.g., variance) of the inferred datacube.

The CASSI system correspond to projecting a three-dimensional datacube into a coded two-dimensional measurement, and we have also demonstrated how this may be extended to hyperspectral video (4D data mapped to a 3D compressive measurement). We have recently applied a similar methodology to another class of compressive video measurements [52], with very encouraging results. In [52] the authors learned a dictionary offline based upon training data. Using methods discussed in this paper, details of which will be reported elsewhere, we have been able to invert compressive measurements of the type in [52] with dictionary learning and recovery performed in situ (like in the CASSI recovery). This points out the generality and utility of the proposed Bayesian dictionary learning for inversion of compressively measured high-dimensional data.

ACKNOWLEDGMENT

The research reported here was supported in part by the Defense Advanced Research Projects Agency (DARPA), under the KeCom program managed by Dr. Mark Neifeld (grant number N66001-11-1-4002).
Partial support was also provided by the National Geospatial Agency (NGA) under the NURI program.

REFERENCES


Fig. 4. First four rows: Channels 1, 10, 19, 25 (wavelengths 398, 467, 591, 725 nm). In each row, Col. 1: true image, followed by reconstructions with Col. 2: BPFA, Col. 3: KSVD, Col. 4: MaxNorm, Col. 5: TwIST (all with $T = 3$ frames). Bottom row: Zoomed-in subimages from channel 19. From left to right - original, reconstructions with BPFA, KSVD, MaxNorm and TwIST. Best viewed electronically, with monitor settings at highest brightness and contrast; zoom in electronically to study results.
Fig. 5. Top to bottom, the rows correspond to channels 2, 6, 10, 14, 18, 26, 31 (wavelengths 420, 460, 500, 540, 580, 660, 710 nm). In each row, Col. 1: true image, followed by $T = 3$ frame reconstructions using Col. 2: BPFA, Col. 3: KSVD, Col. 4: MaxNorm and Col. 5: TwIST. Best viewed electronically, with monitor settings at highest brightness and contrast; zoom in electronically to study results.
Fig. 6. Left two images: Overlay of reconstructed spectral patterns (using BPFA, TwIST, MaxNorm and KSVD) for two regions from the phantom, against the original patterns, Rightmost image: Overlay of spectral patterns of 17 regions from the phantom, reconstructed using BPFA, against the original patterns.

Fig. 7. Comparison between average spectral patterns computed over small neighborhoods around four points chosen from the synthetic birds dataset and its reconstructions using BPFA, KSVD, MaxNorm and TwIST. The four points are highlighted in red, in the bottom-right sub-figure.
Fig. 8. A small portion of channel 26 (wavelength 660 nm) of the image in Figure 5. Left to right, top to bottom: true image, followed by $T = 3$ frame reconstructions using BPFA, KSVD, MaxNorm and TwIST. Best viewed electronically, with monitor settings at highest brightness and contrast; zoom in electronically to study results.
Fig. 9. Left to right, top to bottom: Channels 1, 4, 7, 10, 13, 16, 19, 22, 25 (wavelengths 398, 417, 439, 467, 500, 541, 591, 651, 726 nm). In each group of three image, leftmost: true image, middle: BPFA (3 frames, 100% data, PSNR 30.8) and rightmost: BPFA (3 frames, 30% data, PSNR 28.85). Best viewed electronically, with monitor settings at highest brightness and contrast; zoom in electronically to study results.

Fig. 10. Sample encoded images (real acquisition): Birds dataset (left), and Holly leaf dataset (right)
Fig. 11. From top to bottom the rows correspond to channels 1, 4, 7, 10, 13, 16, 19, 23, 24 (wavelengths 398, 417, 439, 467, 500, 541, 591, 674, 699 nm). In each rows - Col. 1: Original (misregistered), Col. 2 and 3: TwiST - 24 frames and 4 frames (PSNR 14.15), Col. 4 and 5: BPFA with 4 frames (PSNR 16.2) and 12 frames (PSNR 17.1), Col. 6: KSVD (PSNR 10.2), Col. 7: MaxNorm (PSNR 14.82). PSNRs computed after a coarse registration with the ground-truth image. Best viewed electronically, with monitor settings at highest brightness and contrast; zoom in electronically to study results.
Fig. 12. From top to bottom, the rows correspond to channels 1, 12, 16, 20, 23 (wavelengths: 460, 522, 555, 598, 642 nm). In each row, Col. 1: 24-frame TwIST reconstruction, Col. 2: 4 frame TwIST reconstruction (PSNR 29.0), Cols. 3 to 5: BPFA reconstruction with 4, 8, 12 frames (PSNRs 29.03, 33.8, 34.26 resp.), Col. 6: 4-frame KSVD reconstruction (PSNR: 23.86), Col. 7: 4-frame MaxNorm reconstruction (PSNR 30.2). PSNRs measured w.r.t. 24-frame TwIST reconstruction. Best viewed electronically, with monitor settings at highest brightness and contrast; zoom in electronically to study results.
Fig. 13. Reconstruction results for selected frames from the Fish data (containing 200 frames in all) plotted in eight rows. Rows 1 and 2: Results for frame #30, Rows 3 and 4: Results for frame #90, Rows 5 and 6: Results for frame #150, Rows 7 and 8: Results for frame #190. In each pair of rows, the top row contains the encoded image, followed by a TwIST reconstruction of spectral channels 1, 4, 8, 12, 15. The bottom row contains the BPFA reconstruction of channels 1, 4, 8, 12, 15. These channels correspond to wavelengths 448, 477, 526, 591 and 651 nm respectively. Best viewed electronically, with monitor settings at highest brightness and contrast; zoom in electronically to study results.