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. 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 important areas 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 $\boldsymbol{x} \in \mathbb{R}^M$ represent the underlying signal of interest, and $\boldsymbol{x} = \boldsymbol{\Psi}\tilde{\boldsymbol{c}} + \tilde{\boldsymbol{\nu}}$, with $\tilde{\boldsymbol{\nu}} \in \mathbb{R}^M$; the columns of $\boldsymbol{\Psi} \in \mathbb{R}^{M \times M}$ define an orthonormal basis, $\tilde{\boldsymbol{c}} \in \mathbb{R}^M$ is sparse ($i.e.$, $\|\tilde{\boldsymbol{c}}\|_0 \ll M$), and $\|\tilde{\boldsymbol{\nu}}\|_2 \ll \|\boldsymbol{x}\|_2$. The vector $\tilde{\boldsymbol{\nu}}$ represents residual typically omitted after lossy compression [14].

Rather than directly measuring $\boldsymbol{x}$, in CS we seek to measure $\boldsymbol{y} \in \mathbb{R}^m$, with $m \ll M$; measurements are defined by projecting $\boldsymbol{x}$ onto each of the rows of $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times M}$. Specifically, we measure $\boldsymbol{y} = \boldsymbol{\Phi}\tilde{\boldsymbol{c}} + \tilde{\boldsymbol{\epsilon}}$, with $\boldsymbol{\Phi} = \boldsymbol{\Sigma}\boldsymbol{\Psi}$ and $\tilde{\boldsymbol{\epsilon}} = \boldsymbol{\Sigma}\tilde{\boldsymbol{\nu}} + \tilde{\boldsymbol{\delta}}$; $\tilde{\boldsymbol{\delta}}$ accounts for additional measurement noise. The aforementioned incoherence is desired between the rows of $\boldsymbol{\Sigma}$ and columns of $\boldsymbol{\Psi}$. 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 $\boldsymbol{x} \in \mathbb{R}^M$, we seek to simultaneously recover multiple $\{\boldsymbol{x}_i\}_{i=1,N}$, and since we analyze N signals jointly, we also infer the underlying dictionary with which the data may be represented. Specifically, we wish to measure $\{\boldsymbol{y}_i\}_{i=1,N}$ and *jointly* recover $\{\boldsymbol{x}_i\}_{i=1,N}$, with $\boldsymbol{x}_i \in \mathbb{R}^M$ and $\boldsymbol{y}_i \in \mathbb{R}^m$, again with $m \ll M$. It is assumed that each $\boldsymbol{x}_i = \mathbf{D}\boldsymbol{c}_i + \boldsymbol{\nu}_i$, where $\mathbf{D} \in \mathbb{R}^{M \times K}$, and typically $K > M$ ($\mathbf{D}$ is an overcomplete dictionary); $\boldsymbol{c}_i$ is sparse, and $\boldsymbol{\nu}_i$ again represents residual. Each measurement is of the form $\boldsymbol{y}_i = \boldsymbol{\Phi}_i\boldsymbol{c}_i + \boldsymbol{\epsilon}_i$, with $\boldsymbol{\Phi}_i \in \mathbb{R}^{m \times K}$ defined in terms of matrix $\boldsymbol{\Sigma}_i \in \mathbb{R}^{m \times M}$ as $\boldsymbol{\Phi}_i = \boldsymbol{\Sigma}_i\mathbf{D}$, and $\boldsymbol{\epsilon}_i = \boldsymbol{\Sigma}_i\boldsymbol{\nu}_i + \boldsymbol{\delta}_i$. In [21] the authors assumed $\boldsymbol{\Sigma}_i$ was the same for all i, and in [22] it was demonstrated that there are significant advantages to allowing $\boldsymbol{\Sigma}_i$ and hence $\boldsymbol{\Phi}_i$ 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 $\mathbf{D}$ and $\{\boldsymbol{c}_i\}_{i=1,N}$ simultaneously, implying that when performing the measurement we are "blind" to the underlying $\mathbf{D}$ in which each $\boldsymbol{x}_i$ may be sparsely rendered. This is achievable because we process N signals $\{\boldsymbol{y}_i\}_{i=1,N}$ jointly, and the framework has been referred to as *blind* CS [21].

Signal models of the form $\boldsymbol{x}_i = \mathbf{D}\boldsymbol{c}_i + \boldsymbol{\nu}_i$ are also called factor models, where the columns of $\mathbf{D}$ represent factor loadings. If one assumes that $\{\boldsymbol{c}_i\}_{i=1,N}$ are block sparse (the sparsity patterns of $\{\boldsymbol{c}_i\}_{i=1,N}$ are manifested in $B \ll N$ blocks), and if $\boldsymbol{\nu}_i$ 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 $\boldsymbol{x}_i = \mathbf{D}\boldsymbol{c}_i + \boldsymbol{\nu}_i$ representation is also related to a union-of-subspace model [25], particularly when $\{\boldsymbol{c}_i\}_{i=1,N}$ are block sparse. The factor model, GMM, manifold and union-of-subspace models for $\boldsymbol{x}_i$ have been demonstrated to often require far fewer CS measurements [23]–[25] than the ortho-basis model $\boldsymbol{x}_i = \boldsymbol{\Psi}\tilde{\boldsymbol{c}}_i + \tilde{\boldsymbol{\nu}}_i$. 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 $\mathbf{D}$) may be learned; access to such data may not always be possible. In blind CS [21], [22] the *form* of the signal model $\boldsymbol{x}_i = \mathbf{D}\boldsymbol{c}_i + \boldsymbol{\nu}_i$ is assumed, but $\mathbf{D}$ and $\{\boldsymbol{c}_i\}_{i=1,N}$ are inferred jointly based on $\{\boldsymbol{y}_i\}_{i=1,N}$ (implying joint learning of the detailed signal model and associated data $\{\boldsymbol{x}_i\}_{i=1,N}$).

Blind CS is related to dictionary learning [26]–[28], in which one is given $\{\boldsymbol{x}_i\}_{i=1,N}$, and the goal is to infer the dictionary $\mathbf{D}$. 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 $\{\boldsymbol{x}_i\}_{i=1,N}$. Application areas include image denoising and recovery of missing pixels ("inpainting"). In most previous dictionary learning research the underlying data $\{\boldsymbol{x}_i\}_{i=1,N}$ 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. We again note that while there exists significant prior research on theoretical aspects of CS [21], [22], there is very little work on its application to a real physical system. 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 $\boldsymbol{\Sigma}_i$ 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 $\mathbf{\Sigma}_i$.

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. Some issues and observations pertaining to optimal aperture code design in the CASSI system are discussed in Section V. Conclusions are provided in Section VI.

II. CASSI CAMERA AND BLIND CS

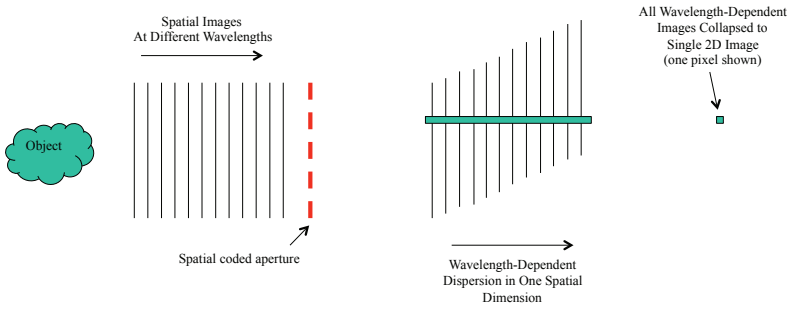
A. Mathematical representation of CASSI measurement

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_λ represents the number of wavelengths. Let $\mathbf{X}_j \in \mathbb{R}^{N_x \times N_y}$ represent the image at wavelength λ_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.

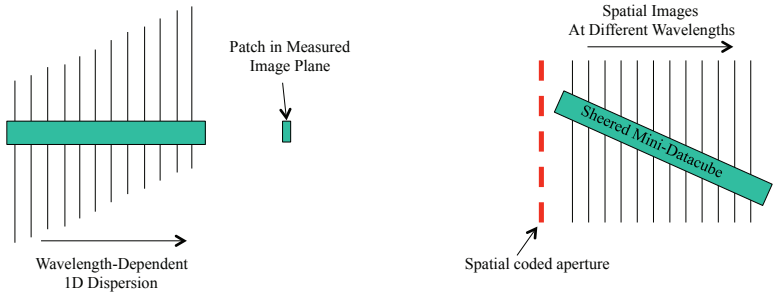
Let $\hat{\mathbf{X}}_j(u,v)$ represent pixel (u,v) in image $\hat{\mathbf{X}}_j$. We now define a *shifted* version of $\hat{\mathbf{X}}_j$, denoted $\mathbf{S}_j$; $\mathbf{S}_j(u,v) = \hat{\mathbf{X}}_j(u - \ell_j, v)$, where $\ell_j > 0$ denotes the shift in pixels at wavelength λ_j, with $\ell_j \neq \ell_{j'}$ for $j \neq j'$; typically the shift ℓ_j is a smooth increasing function of wavelength, manifested physically via a dispersive element [9], [10]. 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 $\mathbf{M}(u,v) = \sum_{j=1}^{N_\lambda} \mathbf{S}_j(u,v)$, defined for all $v \in \{1, \ldots, N_y\}$ and u for which $\mathbf{S}_j$ is defined for all j. Note that component $\mathbf{M}(u,v)$ corresponds to a superposition of coded data from all wavelengths, and because of the shifts $\{\ell_j\}_{j=1,N_\lambda}$, the contribution toward $\mathbf{M}(u,v)$ 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 $\mathbf{M}(u,v)$ 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.



(a)



(b)

Fig. 1. Summary of CASSI measurement process (see [30] for description of physical hardware). (a) The CASSI measurement corresponds to passive hyper spectral 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.

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 $\boldsymbol{y}_i \in \mathbb{R}^m$, where $m = d^2$. Because of the wavelength-dependent shift through the hyperspectral datacube through which $\mathbf{M}$ is constituted, there is a spatially *sheared* set of voxels from the original datacube that contribute toward $\boldsymbol{y}_i$ (see Figure 1); let this sheared subset of voxels define a vector $\boldsymbol{x}_i \in \mathbb{R}^M$, where $M = N_\lambda d^2$. Further, we may consider all possible (overlapping) $d \times d$ contiguous patches of pixels in $\mathbf{M}$, yielding the set of measurement vectors $\{\boldsymbol{y}_i\}_{i=1,N}$, with corresponding sheared

mini-datacubes $\{\boldsymbol{x}_i\}_{i=1,N}$.

We model each $\boldsymbol{x}_i$ in terms of a dictionary $\boldsymbol{x}_i = \mathbf{D}\boldsymbol{c}_i + \boldsymbol{\nu}_i$, with $\boldsymbol{c}_i$ sparse and $\|\boldsymbol{\nu}_i\|_2 \ll \|\boldsymbol{x}_i\|_2$. Further, we may express $\boldsymbol{y}_i = \boldsymbol{\Phi}_i\boldsymbol{c}_i + \boldsymbol{\epsilon}_i$, with $\boldsymbol{\Phi}_i = \boldsymbol{\Sigma}_i\mathbf{D}$ and with $\boldsymbol{\epsilon}_i$ as defined in the Introduction. With the CASSI code design, each $\boldsymbol{\Sigma}_i$ 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 $\boldsymbol{\Sigma}_i$ are *independent* of index i. It was proven in [22] that the effectiveness of blind CS is significantly enhanced if $\boldsymbol{\Sigma}_i$ changes with i. Additionally, if there is no wavelength-dependent code, any permutation of the order of the wavelength-dependent signals will yield the same measurement, undermining uniqueness of the inversion. Concerning (a), if there is no dispersion, the measurement $\mathbf{M}$ 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 $\mathbf{M}$ 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 $\boldsymbol{\Sigma}_i$ has a unique form across each $d \times d$ spatial patch and as a function of wavelength, as encouraged in [22] ($i.e.$, the $\{\boldsymbol{\Sigma}_i\}_{i=1,N}$ 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 $\mathbf{M}$ measured, from which the goal is to recover N_λ spectral bands, each of the same spatial extent as $\mathbf{M}$. 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 $\{\mathbf{M}_t\}_{t=1,T}$, where each $\mathbf{M}_t$ 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. While different coding patterns could be obtained using a rotating wheel of masks as well, the translator system was seen to be adequate, and in fact, provided two degrees of freedom (translations in X and Y directions) as opposed to a single in-plane rotation. 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, $\{\mathbf{M}_t\}_{t=1,T}$, is divided into patches of the form $\{\boldsymbol{y}_{it}\}_{i=1,N;t=1,T}$, 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 $\{\boldsymbol{c}_i\}_{i=1,N}$. Specifically, consider the representation

$$\boldsymbol{y}_i = \boldsymbol{\Sigma}_i \mathbf{D} \boldsymbol{c}_i + \boldsymbol{\epsilon}_i, \qquad \boldsymbol{c}_i = \boldsymbol{s}_i \cdot \boldsymbol{z}_i, \qquad \boldsymbol{\epsilon}_i \sim \mathcal{N}(0, \frac{1}{\gamma_\epsilon}\mathbf{I}_m), \qquad \boldsymbol{s}_i \sim \mathcal{N}(0, \frac{1}{\gamma_s}\mathbf{I}_K) \tag{1}$$

where $\boldsymbol{z}_i \in \{0,1\}^K$, symbol $\cdot$ again represents the Hadamard vector product, and $\mathbf{I}_m$ denotes the $M \times M$ identity matrix. To draw sparse binary vectors $\{\boldsymbol{z}_i\}_{i=1,N}$, consider

$$z_{ik} \sim \text{Bernoulli}(\pi_k), \qquad \pi_k \sim \text{Beta}(a_\pi/K, b_\pi(K-1)/K), \qquad \boldsymbol{d}_k \sim f(\boldsymbol{d}) \tag{2}$$

with the prior $f(\boldsymbol{d})$ discussed below; π_k defines the probability with which dictionary element $\boldsymbol{d}_k$ is used to represent any of the $\boldsymbol{x}_i$. In the limit $K \rightarrow \infty$, note that for finite a_π and b_π 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 π_k will be negligibly small, and most dictionary elements $\{\boldsymbol{d}_k\}_{k=1,K}$ are unlikely to be utilized when representing $\{\boldsymbol{x}_i\}_{i=1,N}$. One may show that the number of non-zero components in each $\boldsymbol{z}_i$ 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 $\boldsymbol{x}_i$ is finite (*i.e.*, $\|\boldsymbol{c}_i\|_0$ is finite). Gamma priors are placed on γ_s and γ_ϵ, $\gamma_s \sim \text{Gamma}(a_s, b_s)$ and $\gamma_\epsilon \sim \text{Gamma}(a_\epsilon, b_\epsilon)$, with hyperparameter settings discussed when presenting results.

Note that we have assumed a zero mean i.i.d. Gaussian model for each the noise vectors $\{\boldsymbol{\epsilon}_i\}_{i=1,N}$. We have visually noticed that the noise affecting actual CASSI measurements has a very low variance. The noise in an actual CASSI system may follow a statistical model different from the zero mean i.i.d. Gaussian model. However, we do not consider that the incorporation of such a model will have any noticeable effect on our results.

Concerning the prior $f(\boldsymbol{d})$ on the columns of $\mathbf{D}$, 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

$$\boldsymbol{d}_k \sim \mathcal{N}(0, \boldsymbol{\Omega}) \tag{3}$$

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

$$\ell(j, j') = \|\boldsymbol{r}_j - \boldsymbol{r}_{j'}\|_2^2 + \beta(\lambda_j - \lambda_{j'})^2 \tag{4}$$

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

$$\boldsymbol{\Omega}(j, j') = \exp[-\ell(j, j')/2\sigma^2] \tag{5}$$

We discuss the setting of β and σ 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. Note that here, we are representing each mini-datacube $\boldsymbol{x}_i$ as a sparse linear combination of spatio-spectral dictionary vectors $\{\boldsymbol{d}_k\}$. Thus, we are imposing sparsity in a spatial as well as spectral sense. Nevertheless, we have found the added regularization afforded by the GP to be useful, as demonstrated in the experimental results.

B. Multi-frame CASSI

Assume we measure T frames of CASSI measurements, $\{\mathbf{M}_t\}_{t=1,T}$. 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 $\boldsymbol{x}_i$. Specifically, for $\boldsymbol{x}_i$ we perform measurements

$$\boldsymbol{y}_{it} = \boldsymbol{\Sigma}_{it}\mathbf{D}(\boldsymbol{s}_i \cdot \boldsymbol{z}_i) + \boldsymbol{\epsilon}_{it} \tag{6}$$

where $\boldsymbol{\Sigma}_{it}$ represents the CASSI projection matrix for measurement t of $\boldsymbol{x}_i$. Therefore, the multiframe CASSI design [30] allows multiple classes of projection measurements on the same $\boldsymbol{x}_i$, substantially

enhancing robustness for inference of $\mathbf{D}$ and $\{\boldsymbol{c}_i\}_{i=1,N}$, recalling $\boldsymbol{c}_i = \boldsymbol{s}_i \cdot \boldsymbol{z}_i$. The priors within the Bayesian blind-CS formulation are exactly as elucidated in the previous subsection, but now a given $\boldsymbol{c}_i$ is essentially inferred via multiple $\{\boldsymbol{\Sigma}_{it}\}_{t=1,T}$.

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

$$p(\{\mathbf{D},\{\boldsymbol{s}_i\},\{\boldsymbol{z}_i\},\gamma_s,\gamma_\epsilon,\{\pi_k\}\}|\{\boldsymbol{y}_{it}\}) \propto \text{Gamma}(\gamma_s|a_s,b_s)\text{Gamma}(\gamma_\epsilon|a_\epsilon,b_\epsilon) \tag{7}$$
$$\times \prod_{i,t} \mathcal{N}(\boldsymbol{y}_{it}|\boldsymbol{\Sigma}_{it}\mathbf{D}(\boldsymbol{s}_i \cdot \boldsymbol{z}_i), \frac{1}{\gamma_\epsilon}\mathbf{I}_m) \prod_{i,k} \mathcal{N}(s_{ik}|0,\gamma_s^{-1})\text{Bernoulli}(z_{ik}|\pi_k)$$
$$\times \prod_k \mathcal{N}(\boldsymbol{d}_k|0,\boldsymbol{\Omega})\text{Beta}(\pi_k|a_\pi/K, b_\pi(K-1)/K)$$

The log of the posterior may therefore be expressed as

$$-\log p(\{\mathbf{D},\{\boldsymbol{s}_i\},\{\boldsymbol{z}_i\},\gamma_s,\gamma_\epsilon,\{\pi_k\}\}|\{\boldsymbol{y}_{it}\}) = \tag{8}$$
$$\tfrac{\gamma_\epsilon}{2}\textstyle\sum_{i,t} \|\boldsymbol{y}_{it} - \boldsymbol{\Sigma}_{it}\mathbf{D}(\boldsymbol{s}_i \cdot \boldsymbol{z}_i)\|_2^2 + \tfrac{1}{2}\sum_k \boldsymbol{d}_k^T\boldsymbol{\Omega}^{-1}\boldsymbol{d}_k \tag{9}$$
$$+\log \text{Gamma}(\gamma_s|a_s,b_s) + \log \text{Gamma}(\gamma_\epsilon|a_\epsilon,b_\epsilon) \tag{10}$$
$$+\tfrac{\gamma_s}{2}\textstyle\sum_{i,k} s_{ik}^2 + \sum_k \log \text{Beta}(\pi_k|a_\pi/K, b_\pi((K-1)/K) + \sum_{i,k} \log \text{Bernoulli}(z_{ik}|\pi_k) \tag{11}$$

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 posterior* (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 ℓ_2 fit between the model and observed data $\{\boldsymbol{y}_{it}\}$, and the second term imposes regularization on the dictionary elements $\{\boldsymbol{d}_k\}_{k=1,K}$, which constitute the columns of $\mathbf{D}$. For the special case in which $\boldsymbol{\Omega} = \mathbf{I}_m$, the second term in (9) reduces to $\frac{1}{2}\sum_k \|\boldsymbol{d}_k\|_2^2$, which corresponds to widely employed ℓ_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 γ_ϵ. The terms in (11) impose explicit sparsity on the weights $\boldsymbol{c}_i = \boldsymbol{s}_i \cdot \boldsymbol{z}_i$, and $\frac{\gamma_s}{2}\sum_{i,k} s_{ik}^2 = \frac{\gamma_s}{2}\sum_i \|\boldsymbol{s}_i\|_2^2$ again imposes ℓ_2 regularization on $\{\boldsymbol{s}_i\}$.

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 $\boldsymbol{c}_i$, via ℓ_1 regularization $\gamma_s \sum_i \|\boldsymbol{c}_i\|_1$; in such an approach all the terms in (11) are essentially just replaced with $\gamma_s \sum_i \|\boldsymbol{c}_i\|_1$. So an optimization-based analog to the proposed approach is of the form [36]

$$\gamma_\epsilon \sum_{i,t} \|\boldsymbol{y}_{it} - \boldsymbol{\Sigma}_{it}\mathbf{D}(\boldsymbol{s}_i \cdot \boldsymbol{z}_i)\|_2^2 + \sum_k \boldsymbol{d}_k^T \boldsymbol{\Omega}^{-1} \boldsymbol{d}_k + \gamma_s \sum_i \|\boldsymbol{c}_i\|_1 \tag{12}$$

In optimization-based approaches one seeks to minimize (12), and the parameters γ_ϵ and γ_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 γ_ϵ and γ_s *a priori*. One advantage of the Bayesian setup is that we infer posterior distributions for γ_ϵ and γ_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 $\{\boldsymbol{c}_i\}$. Specifically, all of the terms in (11) may be replaced by a shrinkage prior. Letting c_{ik} denote the kth component of $\boldsymbol{c}_i$, we may draw $c_{ik} \sim \mathcal{N}(0, \alpha_{ik}^{-1})$, and place a gamma prior separately on each of the α_{ik}. Related priors have been considered in [39]–[41]. We choose to employ the beta-Bernoulli construction because it imposes that components of $\boldsymbol{c}_i$ 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 $\boldsymbol{c}_i$. However, this is essentially a modeling choice, and the methods in [39]–[41] may also be employed to impose sparsity (or near sparsity) on $\{\boldsymbol{c}_i\}$.

Finally, note that in (7), we have assumed all patches $\{\boldsymbol{y}_{it}\}$ are statistically independent. This is not an accurate assumption, as neighboring patches overlap with one another. Within the same basic statistical framework as discussed above, one may impose statistical dependence (like a Markov random field) on the binary weights $\{\boldsymbol{z}_i\}$ [43] as a function of spatial location, thereby accounting for statistical dependencies between proximate patches. We have examined this approach within the context of hyperspectral data, and have not found significant performance improvement to warrant the added computational complexity ($e.g.$, one must introduce a Metropolis Hastings step to the computations, which can be expensive).

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 [44]. The conditional distributions given below can all be derived using standard formulae for conjugate priors [45]. In the following formulae, the symbol '$-$' refers to 'all other parameters except the one being sampled'.

Sampling $\boldsymbol{d}_k$:

$$p(\boldsymbol{d}_k|-) \propto \prod_{i,t} \mathcal{N}(\boldsymbol{y}_{it}|\boldsymbol{\Sigma}_{it}\mathbf{D}(\boldsymbol{s}_i \cdot \boldsymbol{z}_i), \frac{1}{\gamma_\epsilon}\mathbf{I}_m)\mathcal{N}(\boldsymbol{d}_k|0, \boldsymbol{\Omega}), \tag{13}$$

$$p(\boldsymbol{d}_k|-) \sim \mathcal{N}(\boldsymbol{d}_k|\boldsymbol{\mu}_{dk}, \boldsymbol{\Sigma}_{dk}), \tag{14}$$

$$\boldsymbol{\Sigma}_{dk} = (\boldsymbol{\Omega}^{-1} + \gamma_\epsilon \sum_{i,t} s_{ik}^2 z_{ik}^2 \boldsymbol{\Sigma}_{it}^T \boldsymbol{\Sigma}_{it})^{-1} \tag{15}$$

$$\boldsymbol{\mu}_{dk} = \gamma_\epsilon \boldsymbol{\Sigma}_{dk} \sum_{i,t} z_{ik} s_{ik} \boldsymbol{\Sigma}_{it}^T \boldsymbol{y}_{(i,t,-k)}, \tag{16}$$

$$\boldsymbol{y}_{(i,t,-k)} = \boldsymbol{y}_{it} - \boldsymbol{\Sigma}_{it}\mathbf{D}(\boldsymbol{s}_i \cdot \boldsymbol{z}_i) + \boldsymbol{\Sigma}_{it} s_{ik} z_{ik} \boldsymbol{d}_k. \tag{17}$$

The expression for sampling $\boldsymbol{\Sigma}_{dk}$ (and hence, $\boldsymbol{\mu}_{dk}$) reveals the importance of having projections that vary spatially. While the matrices $\boldsymbol{\Sigma}_{it}^T \boldsymbol{\Sigma}_{it}$ 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 $\{\boldsymbol{\Sigma}_{it}\}$ vary spatially and employ (non-zero weights) different components of the mini-datacube.

Sampling z_{ik}:

$$p(z_{ik}|-) \sim \text{Bernoulli}(\frac{p_1}{p_1 + p_0}), \tag{18}$$

$$p_1 = \pi_k \exp\left(-\frac{\gamma_\epsilon}{2}\left(\sum_t s_{ik}^2 \boldsymbol{d}_k^T \boldsymbol{\Sigma}_{it}^T \boldsymbol{\Sigma}_{it} \boldsymbol{d}_k - 2 s_{ik} \boldsymbol{d}_k^T \boldsymbol{\Sigma}_{it}^T \boldsymbol{y}_{(i,t,-k)}\right)\right), \tag{19}$$

$$p_0 = 1 - \pi_k. \tag{20}$$

Sampling s_{ik}:

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

$$\sigma_{sik} = (\gamma_s + \gamma_\epsilon z_{ik}^2 \boldsymbol{d}_k^T \boldsymbol{\Sigma}_{it}^T \boldsymbol{\Sigma}_{it} \boldsymbol{d}_k)^{-1}, \tag{22}$$

$$\mu_{sik} = \gamma_\epsilon \sigma_{sik} z_{ik} \boldsymbol{d}_k^T \sum_t \boldsymbol{\Sigma}_{it}^T \boldsymbol{y}_{(i,t,-k)}. \tag{23}$$

Sampling π_k:

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

Sampling γ_s:

$$p(\gamma_s|-) \sim \Gamma(a_s + \frac{1}{2}KNT, b_s + \frac{1}{2}\sum_i \boldsymbol{s}_i^T \boldsymbol{s}_i). \tag{25}$$

Sampling γ_ϵ:

$$p(\gamma_\epsilon|-) \sim \Gamma(a_\epsilon + \frac{1}{2}\sum_{i,t} \|\boldsymbol{\Sigma}_{it}\|_0, b_\epsilon + \frac{1}{2}\sum_{i,t} \|\boldsymbol{y}_{it} - \boldsymbol{\Sigma}_{it}\mathbf{D}(\boldsymbol{s}_i \cdot \boldsymbol{z}_i)\|^2). \tag{26}$$

Traditionally, Gibbs sampling is run for many burn-in iterations to allow for mixing, followed by the collection phase [44].

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 $\mathbf{D}$, 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 $\mathbf{D}$ 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 (one for each of the collection samples). Computing an average of these dictionary samples would be inappropriate owing to the possiblity of label switching or sign changes. Hence, the maximum likelihood sample is used to define $\mathbf{D}$. In other words, out of N_c collection samples, we choose sample number l ($1 \leq l \leq N_c$), if $\forall m, l \neq m, 1 \leq m \leq N_c, \prod_{k=1}^{K} p(\boldsymbol{d}_k^{(l)}|-) \geq \prod_{k=1}^{K} p(\boldsymbol{d}_k^{(m)}|-)$. 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 [44]. However, we observed that the Gibbs sampler yielded excellent results with as few as 30 iterations. Executing more iterations of the Gibbs sampler did not improve the results significantly (and did not worsen the results). This was also observed for the BPFA model for denoising and inpainting applications in earlier work in [27], [43]. We don't have a complete theoretical understanding of this behavior, but suspect that it may be because the posterior probability density is highly peaked.

After the dictionary is so learned *in situ* for a given CASSI-measured image, the learned $\mathbf{D}$ is then fixed and used to infer $\boldsymbol{c}_i$ for all patches i, and from this an estimate to the underlying patch pixel values is $\mathbf{D}\boldsymbol{c}_i$. 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_λ wavelengths, the inferred patches are of size $d^2 N_\lambda \times 1$. 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], [46] (this was found to be unnecessary). The parameters for the hyperpriors were set as follows: $a_\pi = 0, b_\pi = \frac{N}{2}$, $a_\epsilon = b_\epsilon = a_\alpha = b_\alpha = 10^{-6}$. The parameters of the GP prior for the dictionary elements were set to $\sigma = 5, \beta = 1$. These are standard parameter settings, *i.e.,* they were not 'tuned'. Moreover, perturbations to

the hyperparameters to within $\pm 20\%$ of their original values had no effect on the reconstruction results. Importantly, we have also empirically observed that *in situ* dictionary learning on each new CASSI image was necessary to obtain good inversion results (the data-dependent dictionaries aided inversion performance).

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

$$E(\boldsymbol{x}) = \sum_{t=1}^{T} \|\boldsymbol{y}_t - \boldsymbol{\Sigma}_t \boldsymbol{x}\|^2 + \tau \boldsymbol{\Upsilon}(\boldsymbol{x}). \tag{27}$$

where $\boldsymbol{x}$ and $\{\boldsymbol{y}\}_{t=1,T}$ are the original and encoded data, respectively. The term $\boldsymbol{\Upsilon}(\boldsymbol{x})$ is a regularizer, which could be chosen to impose sparsity in some basis (*e.g.,* wavelet), or chosen to be the total variation (TV) of the underlying 3D spatio-spectral cube $\boldsymbol{x}$ (as considered in this paper). The TV term is defined as follows:

$$\text{TV}(\boldsymbol{x}) = \sum_{\lambda} \sum_{i_y, i_x} \sqrt{(\boldsymbol{x}(i_y+1, i_x, \lambda) - \boldsymbol{x}(i_y, i_x, \lambda))^2 + (\boldsymbol{x}(i_y, i_x+1, \lambda) - \boldsymbol{x}(i_y, i_x, \lambda))^2} \tag{28}$$

where i_y, i_x index discrete spatial coordinates, and λ indexes wavelengths. The parameter τ 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], where superior results have been reported using the TV regularizer as opposed to a sparsity-based term. We performed experiments with different values of τ and wherever possible picked the value of τ that yielded the least mean squared error (MSE) with respect to the ground truth (when available). Generally, we observed that this "optimal" τ 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

$$E(\mathbf{D}, \mathbf{S} = [\boldsymbol{s}_1 | \boldsymbol{s}_2 | ... | \boldsymbol{s}_N]) = \sum_{i,t} \|\boldsymbol{y}_{it} - \boldsymbol{\Sigma}_{it} \mathbf{D} \boldsymbol{s}_i\|^2 \text{ s.t. } \forall i, \|\boldsymbol{s}_i\|_0 \leq T_0$$

where $\mathbf{D} \in \mathbb{R}^{MN_\lambda \times K}$ is a dictionary, $\mathbf{S} \in \mathbb{R}^{K \times N}$ 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) [47]. 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

$$E(\mathbf{D}, \mathbf{S} = [\boldsymbol{s}_1|\boldsymbol{s}_2|...|\boldsymbol{s}_N]) = \sum_{i,t} \|\boldsymbol{y}_{it} - \boldsymbol{\Sigma}_{it}\mathbf{D}\boldsymbol{s}_i\|^2 \text{ s.t.} \quad (29)$$
$$\|\mathbf{D}\|_{2,\infty}^2 \leq B, \|\mathbf{S}^T\|_{2,\infty}^2 \leq B$$

where $\mathbf{D} \in \mathbb{R}^{MN_\lambda \times K}$ is a dictionary and $\mathbf{S} \in \mathbb{R}^{K \times N}$ is a matrix of dictionary coefficients. The max-norm of matrix $\mathbf{D}$ is defined as $\|\mathbf{D}\|_{2,\infty}^2 = \max_j \sqrt{\sum_k \mathbf{D}_{jk}^2}$. 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 $\mathbf{D}$ and $\mathbf{S}$ 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 $\mathbf{D}$ and columns of $\mathbf{S}$ whose norms exceeded $\sqrt{B}$, in order to make those norms equal to $\sqrt{B}$ (see Section 3 of [47]). The step-size for the descent was chosen to be the maximum value in the interval (0,2], which decreased the energy $E(\mathbf{D}, \mathbf{S})$ *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 [47]. 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.

[1]http://www.lx.it.pt/~bioucas/TwIST/TwIST.htm, http://www.cs.technion.ac.il/~elad/software/

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 [48]. 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 [48]. 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 $\mathbf{I}$ and an estimate $\mathbf{J}$, this MSE (averaged over the spectral channels) is given by:

$$e = \frac{1}{N_\lambda |\mathcal{H}|} \sum_{\lambda} \sum_{h \in \mathcal{H}} \|\hat{\mathbf{I}}_{\lambda,h} - \hat{\mathbf{J}}_{\lambda,h}\|^2 \tag{30}$$

where $\hat{\mathbf{I}}_{\lambda,h}$ refers to the magnitude of the h^{th} Fourier coefficient from the spectral channel λ of the image $\mathbf{I}$, 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 $10 \log_{10} \frac{\hat{I}_m \times \hat{I}_m}{e}$ where $\hat{I}_m := \max_{\lambda,h \in \mathcal{H}} \hat{\mathbf{I}}_{\lambda,h}$, is hereafter referred to as HF-PSNR.

The measure SSIM has been proposed in the recent image processing literature [48] 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 [48]. 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 11×11.

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, nor do we consider noise from other statistical models. Note that these synthetic examples are linked to the physical geometry of the CASSI system, as this constitutes a physical manifestation of a blind CS problem; however, these simulations also demonstrate the ability of the Bayesian blind CS setting on general problems of this form, which may be extended to systems beyond CASSI (the proposed Bayesian inversion method is not explicitly tied to the CASSI geometry).

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 CASSI measurements discussed in Section III-B. For N_λ wavelengths in the inferred datacube, each spatial image at a particular wavelength is termed a "channel", and therefore we refer to an N_λ channel CASSI datacube. In an N_λ 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 μm (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

[2]https://ece.uwaterloo.ca/~z70wang/research/ssim/

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 [49]. 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 [50]; 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 λ (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 λ and a signal 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[4] [51]. 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

[3]http://personalpages.manchester.ac.uk/staff/david.foster/Hyperspectral_images_of_natural_scenes_02.html

[4]http://cvrl.ioo.ucl.ac.uk/cmfs.htm

TABLE I

RECONSTRUCTION QUALITY MEASURES (PSNR, HF-PSNR, SSIM) FOR 3-FRAME ($T = 3$) RECONSTRUCTION USING BPFA, KSVD, MAXNORM AND TWIST.

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

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).

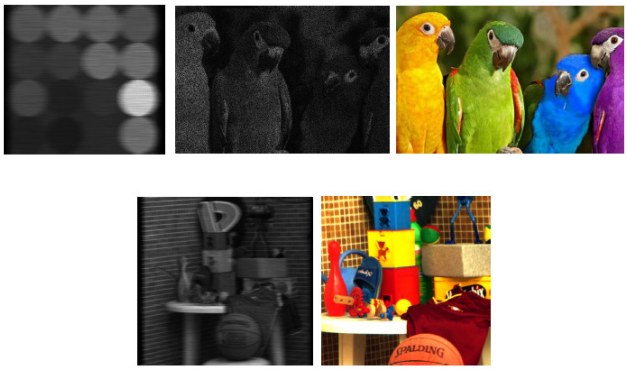


Fig. 2. Top row, first image: Example CASSI measurement for the Phantom dataset. Top row, last two images: Example CASSI measurement (left) and RGB representation (right) for the Birds dataset. Bottom two images: Example CASSI measurement (left) and RGB representation (right) for the Objects dataset. The RGB representations are not spatially aligned with the coded measurements (especially for the Objects dataset) and are provided for reference only.

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 Figure 6 – the spectral patterns in this figure were averaged over a 5×5 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. 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 5×5. One can observe that the BPFA plots are closer to the ground truth.

Given the success of the KSVD algorithm in denoising and inpainting of grayscale images, the inadequate performance of KSVD on reconstruction from CASSI data warrants detailed explanation. To begin with, we again note that the exact same number of dictionary elements was used in both KSVD and BPFA, and the amount of noise added in the simulated snapshots was negligible. The tendency of

KSVD to produce nearly uniform spectral responses 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 are considering diverse spectral patterns over several wavelengths, and devising a similarly appropriate scheme is beyond the scope of this paper. In fact, in previous work on denoising and inpainting of hyperspectral images [32], it has been observed that BPFA outperformed KSVD significantly. Furthermore, there is a lot of difference in the manner in which sparse codes are updated in KSVD and BPFA. Given a dictionary, the sparse codes for each patch are updated independently in KSVD. In BPFA, we have a hierarchical model for $\{s_{ik}\}$ as well as $\{z_{ik}\}$, which are governed by parameters γ_s and $\{\pi_k\}$ respectively. The parameters γ_s and $\{\pi_k\}$ are also inferred along with the dictionary vectors and the dictionary coefficients. While such dependencies on the sparse codes could be adopted within the KSVD algorithm, doing so falls outside the scope of our paper. A popular optimization-based method that incorporates the notion of group sparsity is the method proposed in [52], where it was used for image denoising and demosaicking. However, this method essentially works with groups of structurally similar patches. Finding groups of similar patches is a meaningful operation in experiments on image denoising or demosaicking. However, in the case of experiments with a system like CASSI, this is not possible, since the original signal has been modulated by *random* aperture patterns to produce the measured snapshots.

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.

The GP prior on the dictionary elements imposes the belief that the spectral patterns vary smoothly, a reasonable assumption obeyed by hyperspectral data, including all the simulated datasets we have worked with as well as the data underlying the real CASSI acquisitions from Section IV-G. We studied the effect

TABLE II

EFFECT OF GP ON RECONSTRUCTION (SYNTHESIZED OBJECT DATASET).

# samples for dictionary learning	PSNR with GP	PSNR without GP
5000	24.82	22.68
10000	25.99	23.81
20000	26.45	24.66
50000	27.2	25.2

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 4×4) in dictionary learning, from 5000 to 50,000 per frame (out of a total of 6.9×10^5 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 $\mathbf{\Sigma}_{it}$ 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.

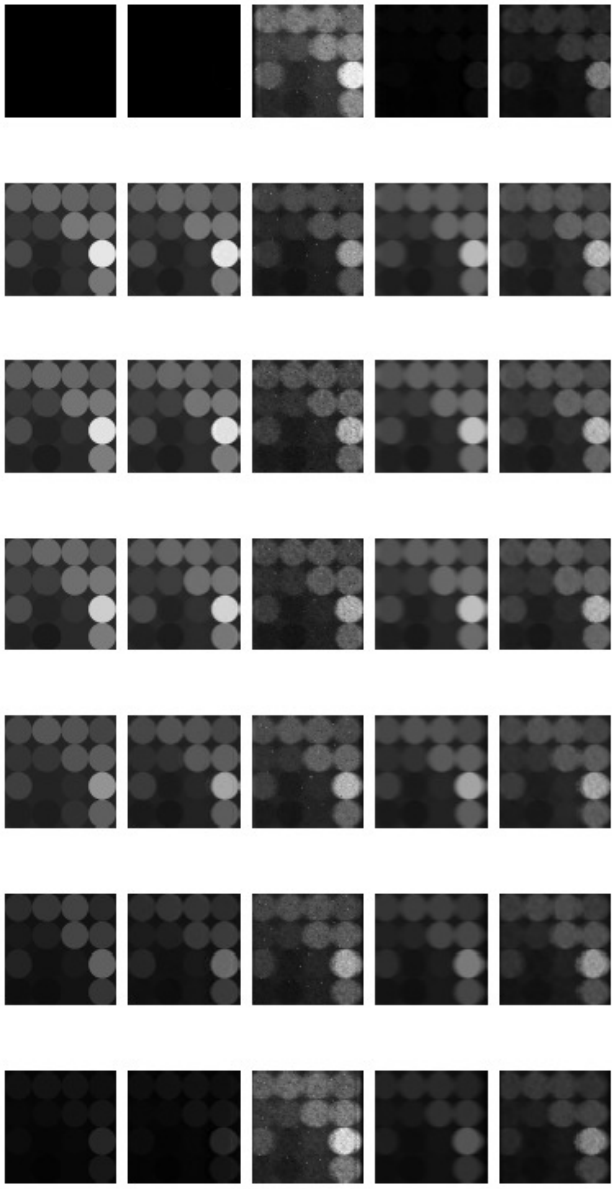


Fig. 3. Reconstruction results for the Phantom data using 3 frames - for channels 1, 11, 17, 22, 33, 38 and 44 (corresponding to wavelengths 501, 801, 981, 1131, 1461, 1611 and 1791 nm). From left to right in each row - Col. 1: true image, Col. 2: BPFA, Col. 3: KSVD , Col 4: MaxNorm, and Col. 5: TwIST.

G. Results on Real Data

We now present results to demonstrate that our method which based on blind CS works on actual data acquired by the CASSI system. This is an important contribution, as blind CS has not been applied to CASSI acquisitions so far.

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 24^{th} 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 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 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

TABLE III

PSNR VALUES FOR 4-FRAME RECONSTRUCTION OF DIFFERENT DATASETS USING BPFA, KSVD, MAXNORM AND TWIST.

Dataset	BPFA	KSVD	MaxNorm	TwIST
Real-birds:Ground-truth image after coarse registration	16.2	10.2	14.82	14.15

that it was computationally far more expensive than BPFA (120 minutes with MaxNorm as opposed to 45 minutes for BPFA).

In the case of some practical compressive sensing systems, the available forward model may be not be accurate, because precise calibration may be infeasible. In such cases, it is important to examine the effect of mis-calibration on the reconstruction results. However, in all our experiments with real CASSI acquisitions, the available forward model explicitly accounts for practical issues such as the effect of blur on the mask pattern or subpixel misalignments (see Section 3 of [30] for more details) and is not merely based on the ideal mask pattern. Our convincing experimental results indicate that the available forward model is accurate, and hence we do not consider a separate study of the effect of mis-calibration to be crucial.

V. CHOICE AND DESIGN OF MASK/PROJECTION MATRIX FOR CASSI

In all of the examples presented above, the CASSI mask was designed as binary, with elements of 0 or 1, with a probability 0.5 for each binary value. It is of interest to consider possible optimization of the projection matrix within the CS measurements. There have been studies that have examined design of the CS projection matrices [53], [54], and such approaches were also consider in the context of this study. The framework in [54] assumes that the underlying signals $\{\boldsymbol{x}_i\}$ of interest are drawn from a Gaussian mixture model (GMM), and under this setting the design of optimal projection matrices was considered. We first discuss how we may specialize our signal model readily to a GMM setting.

Recall that, for the problem considered here, $\boldsymbol{x}_i \in \mathbb{R}^M$ represents the data in a sheared mini datacube of the overall hyperspectral datacube. In Section III-A we developed a signal model for each $\boldsymbol{x}_i$, as being represented in terms of a sparse linear combination of dictionary elements. In this setting each $\boldsymbol{x}_i$ uses a specific subset of columns of $\mathbf{D}$, specified by the sparse vector $\boldsymbol{z}_i$. In equation (1) we now further assume that the sparse vector $\boldsymbol{z}_i$ is drawn from a mixture model, with L mixture components, and each mixture component is characterized by a specific usage of dictionary elements (a specific $\boldsymbol{z}_l$ is associated with the lth mixture component). If all other aspects of the model remain unchanged, this

implies that the $\{\boldsymbol{x}_i\}$ are drawn from a Gaussian mixture model. Specifically, assume that $\boldsymbol{x}_i$ is drawn from a mixture of L components, and variable $\zeta_i \in \{1, \ldots, L\}$ denotes which mixture component $\boldsymbol{x}_i$ is drawn from. If $\mu_l \in (0, 1)$ represents the probability of mixture component l, with $\sum_{l=1}^{L} \mu_l = 1$, then $\zeta_i \sim \sum_{l=1}^{L} \mu_l \delta_l$, with δ_l a point measure concentrated at l. From (1), upon marginalizing out the $\boldsymbol{s}_i$, we have $\boldsymbol{x}_i \sim \mathcal{N}(0, \frac{1}{\gamma_s}\mathbf{D}\boldsymbol{\Lambda}_{\zeta_i}\mathbf{D}^T + \frac{1}{\gamma_\nu}\mathbf{I}_M))$, where γ_ν is the noise precision and $\boldsymbol{\Lambda}_{\zeta_i} = \text{diag}(\boldsymbol{z}_{\zeta_i})$; hence, $\boldsymbol{\Lambda}_{\zeta_i}$ selects columns of $\mathbf{D}$, defined by $\boldsymbol{z}_{\zeta_i}$, for representing $\boldsymbol{x}_i$. This GMM construction is equivalent to saying that the $\boldsymbol{x}_i$ come from a union of subspaces [25], where here we have L subspaces, each of which is characterized by which columns of $\mathbf{D}$ are employed within the respective mixture component. This is therefore a reasonable signal model, which using the theory in [54] we may employ to design optimal projection matrices.

Within the context of these experiments, we consider hyperspectral imagery similar to that considered in the above discussions, and we designed an associated GMM signal model for data $\{\boldsymbol{x}_i\}$. We then used the theory in [54] to design optimal masks, with the goal of maximizing the mutual information between the underlying $\{\boldsymbol{x}_i\}$ and the measured $\{\boldsymbol{y}_i\}$. This was done with the restriction that the mask used at each spectral band was a shifted version of the same template, with the wavelength-dependent mask shift defined by the CASSI dispersion. We did this design under the assumption that the mask may take values in the range $[0, 1]$, which may be implemented with a gray-scale graded mask. This design is therefore even more general than the binary $\{0, 1\}$ mask considered in the above experiments.

After designing masks of this type, we compared CS recovery for the type of data considered above. After an extensive set of simulated experiments, we found that the designed masks yielded only very slightly better CS recovery performance that using the simple binary mask with $\{0, 1\}$ elements, drawn Bernoulli(0.5). This is attributed to the fact that the requirement that the masks at the different wavelengths be a shifted version of the same template mask is very restrictive. Additionally, the big gains in designed masks found in [54] were manifested by adaptive masks, in which a sequence of compressive measurements were performed, and the mask changed sequentially for the next measurement based on the previous compressive measurements (with mask designed and adapted to optimize the mutual information between the underlying signal and the measured data). However, this would require time-dependent and fast adaptation of the CASSI mask, which is a significant challenge. Further, this adaptation must be performed within localized hyperspectral mini batches, which is quite complicated to implement in practice. Therefore, from a practical standpoint, and after extensive simulations and testing, our experience indicates that the simple binary mask, drawn Bernoulli(0.5) provides a good balance between simplicity and performance. Other Bernoulli probabilities were considered beyond 0.5, and extensive experiments

indicated that a probability of 0.5 yielded best performance.

VI. 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). Our method makes the assumption of smooth variation of image intensity values across spectra, a reasonable assumption obeyed by most hyperspectral images. This explains the uniformly good performance of our method on both simulated and real datasets in this paper. 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. We have generally found that three snapshots are needed for good quality recovery of the spectral patterns. This number is not universal and will depend on the nature of the spectral patterns, the coded aperture and the sparsity of image patches in the learned basis. Developing more precise relationships between these and the minimum number of snapshots required, is an interesting avenue for future work.

The dictionary learning approach presented in this paper is related to previous approaches based on 'endmember spectral signatures' [55], which are popular in hyperspectral image processing. Consider the spectral pattern in a hyperspectral image $\mathbf{X} \in \mathbb{R}^{N_x \times N_y \times N_\lambda}$ at pixel location i, denoted as $\mathbf{X}_i \in \mathbb{R}^{N_\lambda \times 1}$. The vector $\mathbf{X}_i$ can be considered to be a weighted average (more specifically, a convex combination) of a number 'endmember spectral signatures', each of which specifies the spectral pattern of a particular material or geographical entity such as vegetation, roads, water-bodies, *etc*. This is especially true of lower resolution hyperspectral images, as acquired in remote sensing applications. Thus we have $\mathbf{X}_i = \sum_{j=1}^{N_e} \rho_i^j \boldsymbol{s}^j$ where $\{\boldsymbol{s}^j\}_{j=1}^{N_e}$ are the endmember spectral signatures and $\{\rho^j\}_{j=1}^{N_e}$ are the endmember mixing proportions ($i.e.$, coefficients of the convex combination). The total number of endmembers N_e is usually far smaller than N_λ. There exist several recent research papers which exploit this fact to perform efficient and accurate endmember unmixing (*i.e.,* estimation of the mixing coefficients) and reconstruction of hyperspectral images from compressive measurements of very low dimensionality [10], [56]–[59]. For instance, in the work of Li *et al.* [10] or Martin *et al.* [59], unmixing and reconstruction are performed by constrained minimization of the total TV norm of all endmember coefficients, assuming fixed endmember

signatures. In addition, the method by Golbabaee *et al.* in [56] also exploits the inherent low-rank nature of hyperspectral datacubes represented as matrices, by minimizing a matrix nuclear norm term in addition to the TV norm, within a similar constrained minimization framework. The work by Zhang *et al.* [58] also estimates the endmember signatures *in situ* from the compressed measurements, along with the mixing coefficients, in an alternating minimization framework. The approach by Golbabaee *et al.* in [57] estimates mixing proportions by treating them as sparse linear combinations of vectors from a known (*e.g.,* wavelet) basis.

In this paper the dictionary learning is closely related to endmember learning. In this paper the dictionary has been learned based on the goal of fitting the data, and we have not used endmember information to impose prior knowledge about material properties. In future research it is of interest to extend the framework developed here, in which we employ prior knowledge about the types of materials that may be sensed, and leverage known endmember libraries. In this setting, we may assume access to an endmember library of possible materials that may be encountered. When inferring an appropriate dictionary for the data under test, each of the dictionary elements in our inversion method may either be drawn from the known endmember library, or new dictionary elements may be inferred from the Gaussian dictionary prior. It is anticipated that this approach may yield significant performance improvements, by leveraging the aforementioned and vast prior research on endmember design and the associated linkage to known material characteristics.

The CASSI system corresponds to projecting a three-dimensional datacube into a coded two-dimensional measurement. We have recently applied a similar methodology to another class of compressive video measurements [60], with very encouraging results. In [60] 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 [60] 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 (the statistical methodology extends the CASSI system we have used here for demonstration).

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; the Office of Naval Research under the Basic Research Challenge in Compressive Sensing; and by the

Department of Energy, NA-22.

REFERENCES

[1] M. A. Neifeld and P. Shankar, "Feature-specific imaging", *Appl. Opt.*, vol. 42, pp. 3379–3389, 2003.

[2] D. J. Brady, N. Pitsianis, X. Sun, and P. Potuluri, "Compressive sampling and signal inference", *U.S. Patent 7,283,231*.

[3] E. J. Candès, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements", *Commun. Pure Appl. Math.*, vol. 59, pp. 1207–1223, 2006.

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

[5] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly, "Compressed sensing MRI", in *IEEE Signal Processing Magazine*, 2007.

[6] E. Y. Sidky and X. Pan, "Image reconstruction in circular conebeam computed tomography by constrained, total-variation minimization", *Phys. Med. Biol.*, vol. 53, pp. 4777–4807, 2008.

[7] G.-H. Chen, J. Tang, and S. Leng, "Prior image constrained compressed sensing: a method to accurately reconstruct dynamic CT images from highly under sampled projection data sets", *Med. Phys.*, vol. 35, pp. 660–663, 2008.

[8] D. J. Brady, K. Choi, D. L. Marks, R. Horisaki, and S. Lim, "Compressive holography", *Opt. Express*, vol. 17, pp. 13040–13049, 2009.

[9] M. E. Gehm, R. John, D. J. Brady, R. M. Willett, and T. J. Schulz, "Single-shot compressive spectral imaging with a dual-disperser architecture", *Opt. Express*, vol. 15, pp. 14013–14027, 2007.

[10] C. Li, T. Sun, K. Kelly, and Y. Zhang, "A compressive sensing and unmixing scheme for hyperspectral data processing", *IEEE Transactions on Image Processing*, vol. 21, no. 3, pp. 1200–1210, 2011.

[11] M. Shankar, N.P. Pitsianis, and D.J. Brady, "Compressive video sensors using multichannel imagers", *Appl. Opt.*, vol. 49, pp. B9–B17, 2010.

[12] Y. Hitomi, J. Gu, M. Gupta, T. Mitsunaga, and S. K. Nayar, "Video from a single coded exposure photograph using a learned over-complete dictionary", in *IEEE International Conference on Computer Vision (ICCV)*, Nov 2011.

[13] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information", *IEEE Transactions on Information Theory*, vol. 52, pp. 489–509, 2006.

[14] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, second edition, 1998.

[15] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression", *Annals of Statistics (with discussion)*, vol. 32, pp. 407–499, 2004.

[16] D.L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit", Tech. Rep., Stanford, 2006.

[17] T. Goldstein and S. Osher, "The split Bregman method for l1 regularized problems", *SIAM J. Imaging Sciences*, 2009.

[18] D. Needell and J.A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples", *Applied and Computational Harmonic Analysis*, 2009.

[19] J. M. Bioucas-Dias and M. A. T. Figueiredo, "A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration", *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2992–3004, 2007.

[20] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing", *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[21] S. Gleichman and Y. C. Eldar, "Blind compressed sensing", *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 69586975, 2011.

[22] J. Silva, M. Chen, Y. Eldar, G. Sapiro, and L. Carin, "Blind compressed sensing over a structured union of subspaces", *arXiv.org*, vol. abs/1103.2469, 2011.

[23] M. Chen, J. Silva, J. Paisley, C. Wang, D. Dunson, and L. Carin, "Compressive sensing on manifolds using a nonparametric mixture of factor analyzers: Algorithm and performance bounds", *IEEE Trans. Signal Processing*, vol. 58, pp. 6140 – 6155, 2010.

[24] R. Baraniuk and M. Wakin, "Random projections of smooth manifolds", *Foundations of Computational Mathematics*, vol. 9, pp. 51–77, 2009.

[25] Y. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery", *IEEE Trans. Signal Processing*, vol. 58, pp. 3042–3054, 2010.

[26] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries", *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[27] M. Zhou, H. Chen, J. Paisley, L. Ren, L. Li, Z. Xing, D. Dunson, G. Sapiro, and L. Carin, "Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images", *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 130 –144, 2012.

[28] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration", *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 53 –69, 2008.

[29] A. Wagadarikar, R. John, R. Willett, and D. J. Brady, "Single disperser design for coded aperture snapshot spectral imaging", *Applied Optics*, vol. 47, no. 10, pp. B44–B51, 2008.

[30] D. Kittle, K. Choi, A. Wagadarikar, and D. J. Brady, "Multiframe image estimation for coded aperture snapshot spectral imagers", *Applied Optics*, vol. 49, no. 36, pp. 6824–6833, Dec 2010.

[31] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising", in *CVPR (2)*, 2005, pp. 60–65.

[32] Z. Xing, M. Zhou, A. Castrodad, G. Sapiro, and L. Carin, "Dictionary learning for noisy and incomplete hyperspectral images", *SIAM Journal of Imaging Sciences*, vol. 5, no. 1, pp. 33–56, 2012.

[33] M. Aharon, M. Elad, and A. M. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation", *IEEE Trans. Signal Processing*, vol. 54, pp. 4311–4322, 2006.

[34] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration", *IEEE Trans. Image Processing*, vol. 17, pp. 53–69, 2008.

[35] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding", in *Proc. International Conference on Machine Learning*, 2009.

[36] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Supervised dictionary learning", in *Proc. Neural Information Processing Systems*, 2008.

[37] J. Mairal, G. Sapiro, and M. Elad, "Learning multiscale sparse representations for image and video restoration", *SIAM Multiscale Modeling and Simulation*, vol. 7, pp. 214 – 241, 2008.

[38] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration", in *Proc. International Conference on Computer Vision*, 2009.

[39] M. Tipping, "Sparse bayesian learning and the relevance vector machine", *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, 2001.

[40] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing", *IEEE Transactions on Signal Processing*, vol. 56, pp. 2346–2356, 2008.

[41] S. Ji, D. Dunson, and L. Carin, "Multi-task compressive sensing", *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 92–106, 2009.

[42] R. Thibaux and M. Jordan, "Hierarchical beta processes and the indian buffet process", *AISTATS*, vol. 2, pp. 564–571, 2007.

[43] M. Zhou, H. Yang, G. Sapiro, D. Dunson, and L. Carin, "Dependent hierarchical beta process for image interpolation and denoising", in *AISTATS*, 2012.

[44] C. Robert and G. Casella, *Monte Carlo Statistical Methods (Springer Texts in Statistics)*, Springer-Verlag New York, Inc., 2005.

[45] D. Fink, "A compendium of conjugate priors. in progress report: Extension and enhancement of methods for setting data quality objectives", Tech. Rep., Montana State University, 1995.

[46] T. L. Griffiths and Z. Ghahramani, "Infinite latent feature models and the indian buffet process", in *Neural Information Processing Systems*, 2005, pp. 475–482.

[47] J. Lee, B. Recht, R. Salakhutdinov, N. Srebro, and J. A. Tropp, "Practical large-scale optimization for max-norm regularization", in *NIPS*, 2010, pp. 1297–1305.

[48] Z. Wang and A. Bovik, "Mean squared error: Love it or leave it?-a new look at signal fidelity measures", *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98–117, 2009.

[49] D. Kittle, K. Choi, A. Wagadarikar, and D. J. Brady, "Multiframe image estimation for coded aperture snapshot spectral imagers", *Applied Optics*, vol. 49, pp. 6824–6833, 2010.

[50] S. Nascimento, F. Ferreira, and D. Foster, "Statistics of spatial cone-excitation ratios in natural scenes", *Journal of the Optical Society of America (A)*, , no. 19, pp. 1484–1490, 2002.

[51] G. Wyszecki and W. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, John Wiley & Sons, Inc., New York, second edition, 1982.

[52] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration", in *International Conference on Computer Vision*, 2009.

[53] J.M. Duarte-Carvajalino and G. Sapiro, "Learning to sense sparse signals: simultaneous sensing matrix and sparsifying dictionary optimization", *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1395–1408, 2009.

[54] W. Carson, M. Chen, M. Rodrigues, R. Calderbank, and L. Carin, "Communications inspired projection design with application to compressive sensing", *SIAM J. Imaging Sciences*, 2012.

[55] N. Keshava and J. F. Mustard, "Spectral unmixing", *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 44 –57, 2002.

[56] M. Golbabaee and P. Vandergheynst, "Joint trace/tv norm minimization: a new efficient approach for spectral compressive imaging", in *IEEE International Conference on Image Processing*, 2012.

[57] M. Golbabaee, S. Arberet, and P. Vandergheynst, "Compressive source separation: theory and methods for hyperspectral imaging", *arXiv.org*, vol. 1208.4505, 2012.

[58] Q. Zhang, R. Plemmons, D. Kittle, D. Brady, and S. Prasad, "Joint segmentation and reconstruction of hyperspectral data with compressed measurements", *Applied Optics*, vol. 50, no. 22, pp. 4417–4435, 2011.

[59] G. Martin, J. Bioucas-Dias, and A. Plaza, "A new technique for hyperspectral compressive sensing using spectral unmixing", in *SPIE Optics and Photonics, Satellite Data Compression, Communication, and Processing Conference*, 2012.

[60] Y. Hitomi, J. Gu, M. Gupta, T. Mitsunaga, and S. K. Nayar, "Video from a single coded exposure photograph using a learned over-complete dictionary", in *IEEE International Conference on Computer Vision*, Nov 2011.

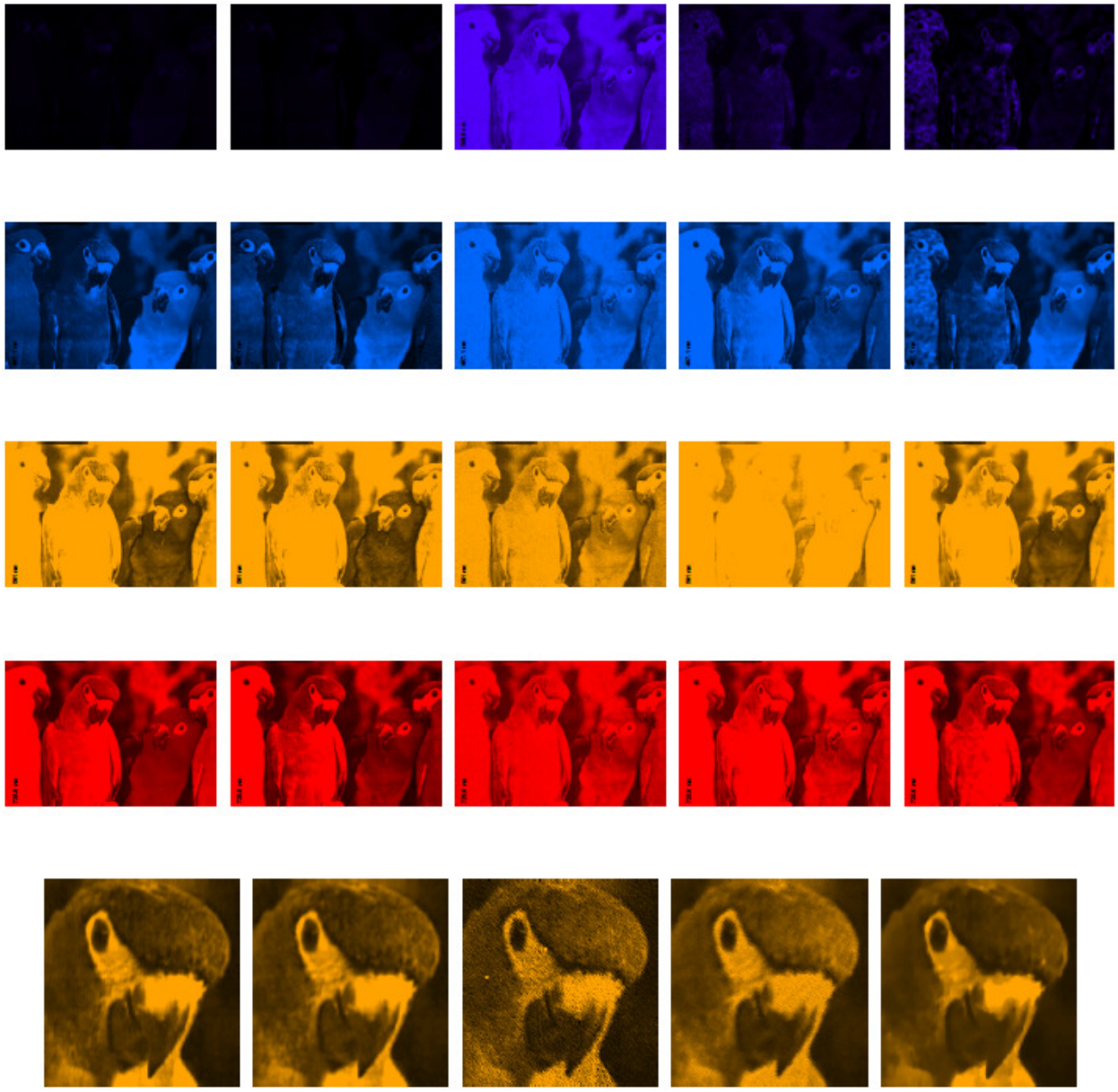


Fig. 4. First four rows: Channels 1, 10, 19, 25 (wavelengths 398, 467, 591, 725 nm). In each row, Col. 1: true image, followed by reconstructions with Col. 2: BPFA, Col. 3: KSVD, Col. 4: MaxNorm, Col. 5: TwIST (all with $T = 3$ frames). Bottom row: Zoomed-in subimages from channel 19. From left to right - 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.

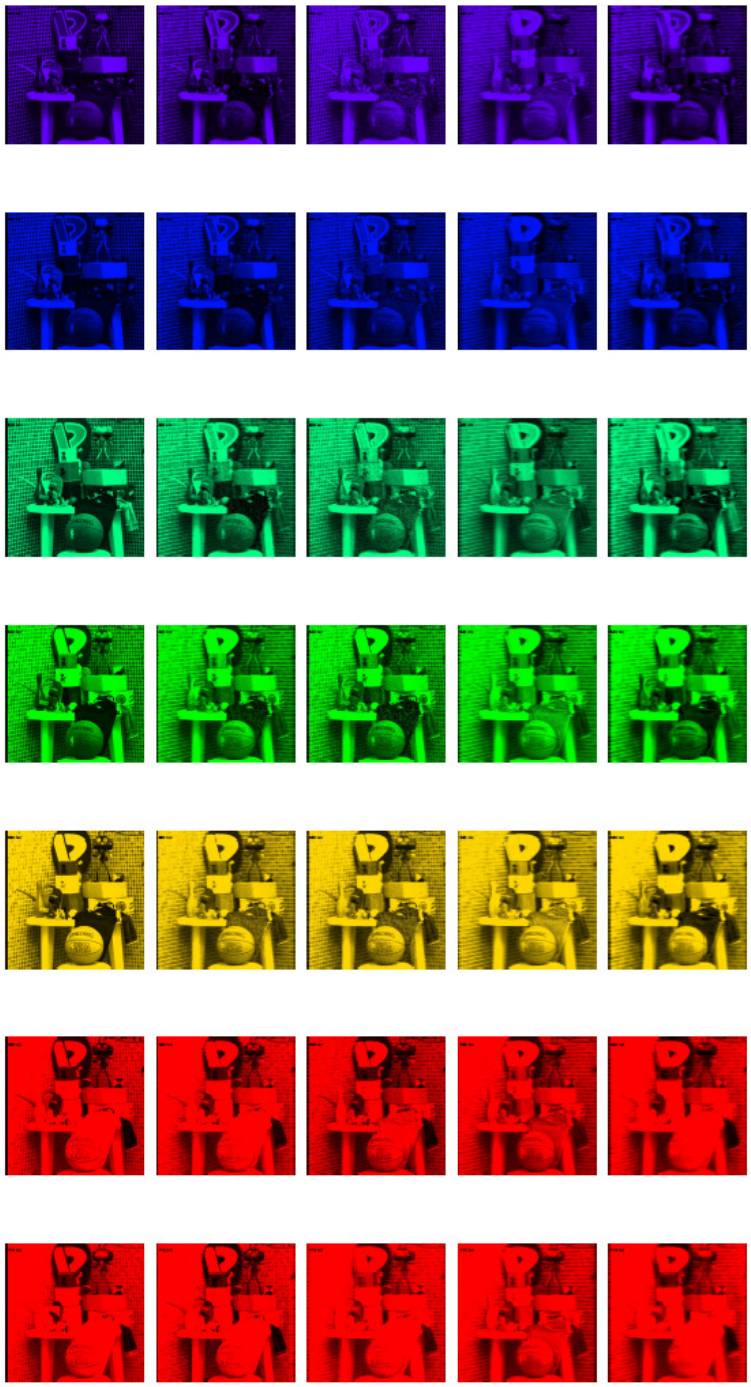


Fig. 5. Top to bottom, the rows correspond to channels 2, 6, 10, 14, 18, 26, 31 (wavelengths 420, 460, 500, 540, 580, 660, 710 nm). In each row, Col. 1: true image, followed by $T = 3$ frame reconstructions using Col. 2: BPFA, Col. 3: KSVD, Col. 4: MaxNorm and Col. 5: TwIST. Best viewed electronically, with monitor settings at highest brightness and contrast; zoom in electronically to study results.

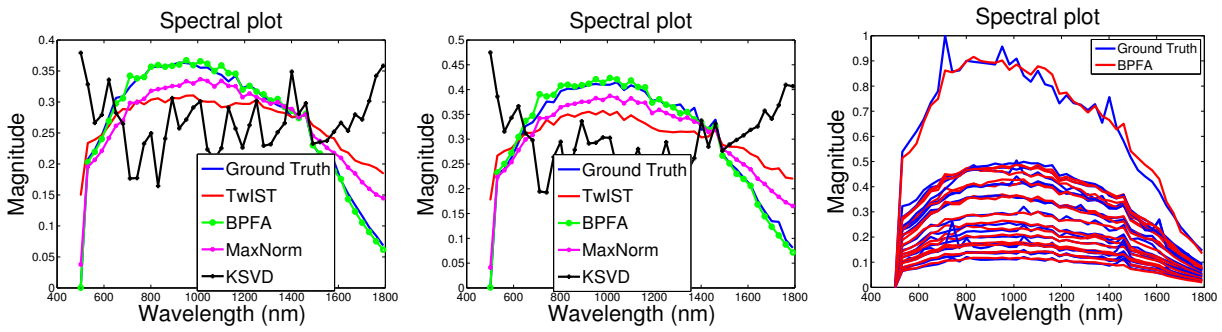


Fig. 6. Left two images: Overlay of reconstructed spectral patterns (using BPFA, TwIST, MaxNorm and KSVD) for two regions from the phantom, against the original patterns, Rightmost image: Overlay of spectral patterns of 17 regions from the phantom, reconstructed using BPFA, against the original patterns

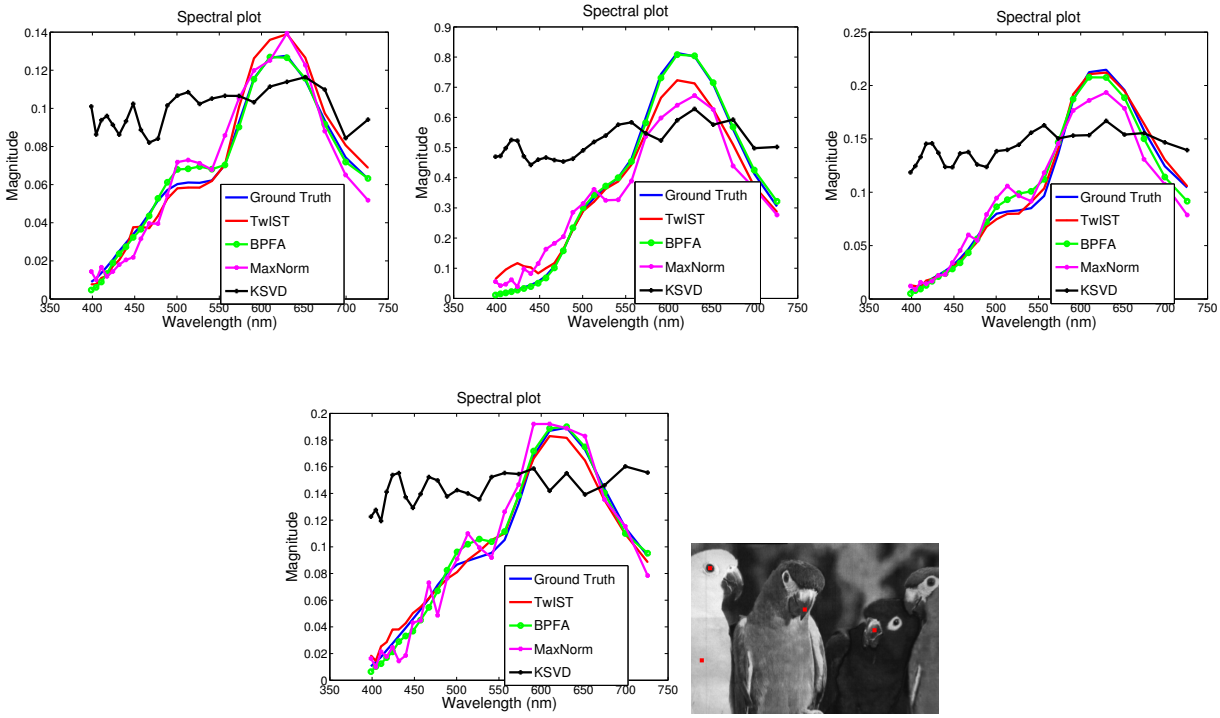


Fig. 7. Comparison between average spectral patterns computed over small neighborhoods around four points chosen from the synthetic birds dataset and its reconstructions using BPFA, KSVD, MaxNorm and TwIST. The four points are highlighted in red, in the bottom-right sub-figure.

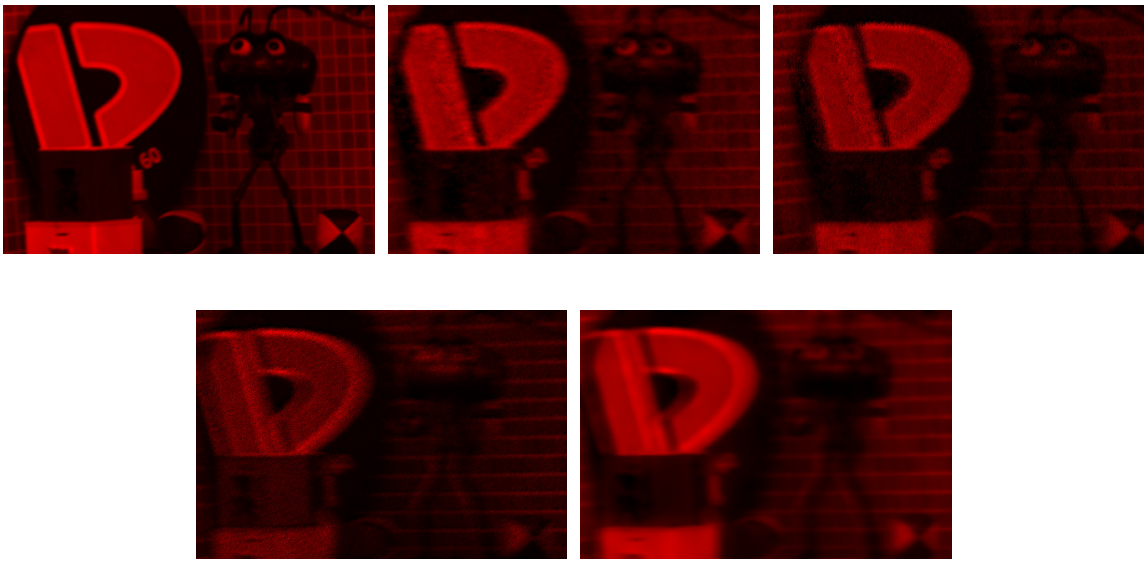


Fig. 8. A small portion of channel 26 (wavelength 660 nm) of the image in Figure 5. Left to right, top to bottom: true image, followed by $T = 3$ frame reconstructions using BPFA, KSVD, MaxNorm and TwIST. Best viewed electronically, with monitor settings at highest brightness and contrast; zoom in electronically to study results.

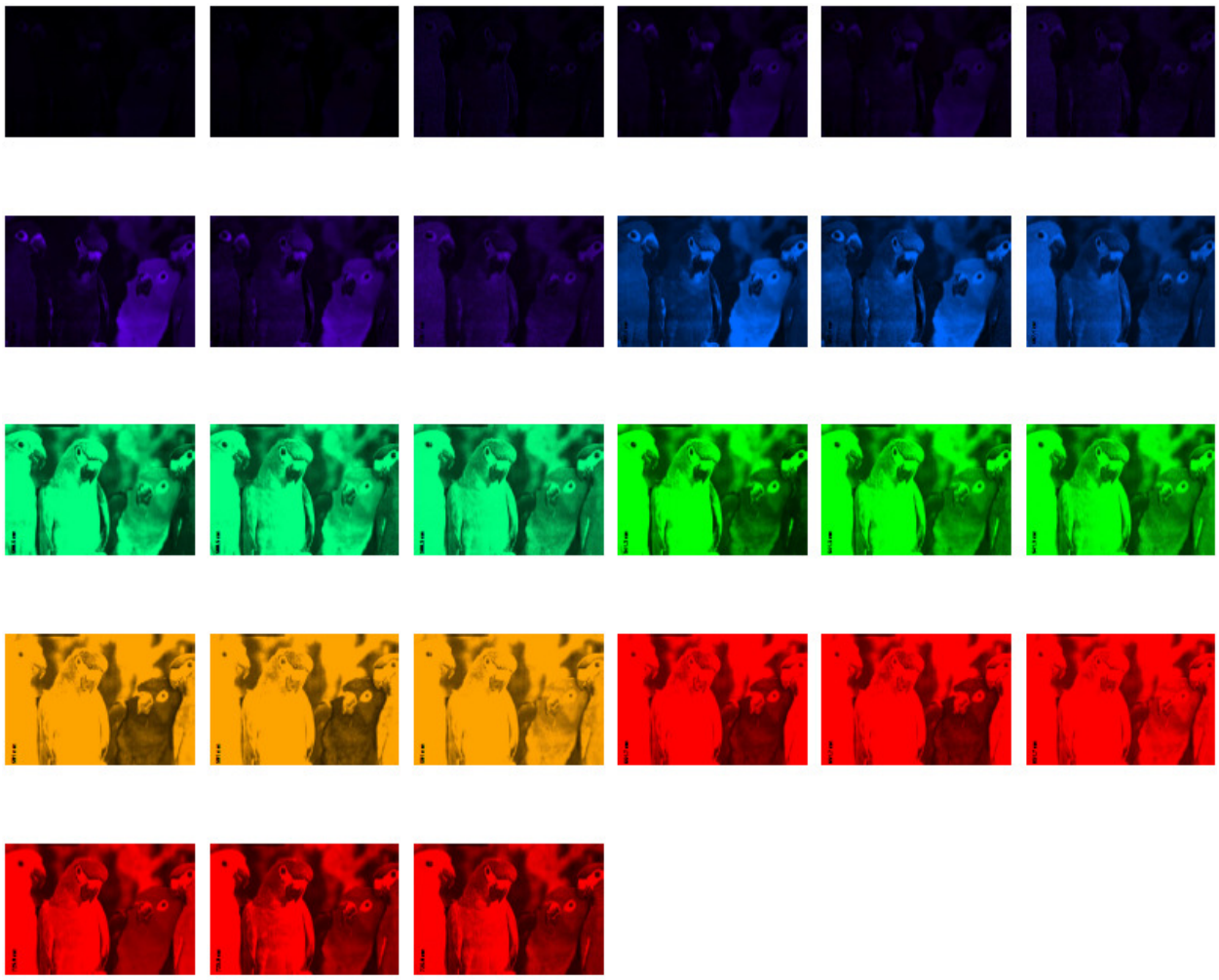


Fig. 9. Left to right, top to bottom: Channels 1, 4, 7, 10, 13, 16, 19, 22, 25 (wavelengths 398, 417, 439, 467, 500, 541, 591, 651, 726 nm). In each group of three image, leftmost: true image, middle: BPFA (3 frames, 100% data, PSNR 30.8) and rightmost: BPFA (3 frames, 30% data, PSNR 28.85). Best viewed electronically, with monitor settings at highest brightness and contrast; zoom in electronically to study results.

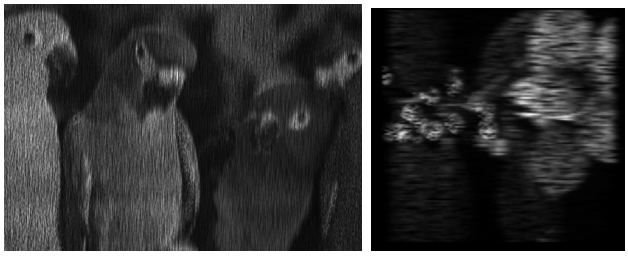


Fig. 10. Sample encoded images (real acquisition): Birds dataset (left), and Holly leaf dataset (right)

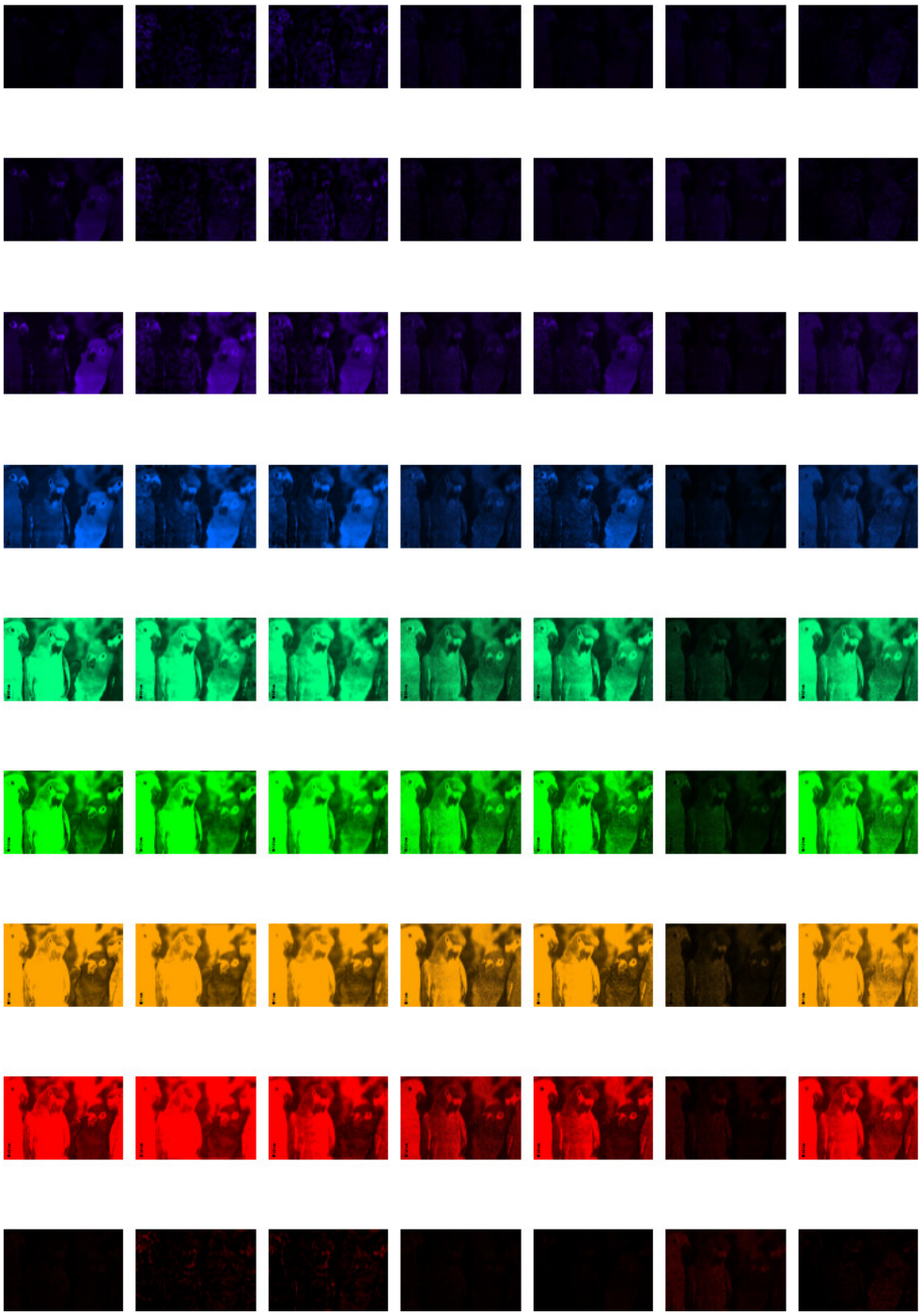


Fig. 11. From top to bottom the rows correspond to channels 1, 4, 7, 10, 13, 16, 19, 23, 24 (wavelengths 398, 417, 439, 467, 500, 541, 591, 674, 699 nm). In each rows - Col. 1: Original (misregistered), Col. 2 and 3: TwIST - 24 frames and 4 frames (PSNR 14.15), Col. 4 and 5: BPFA with 4 frames (PSNR 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.

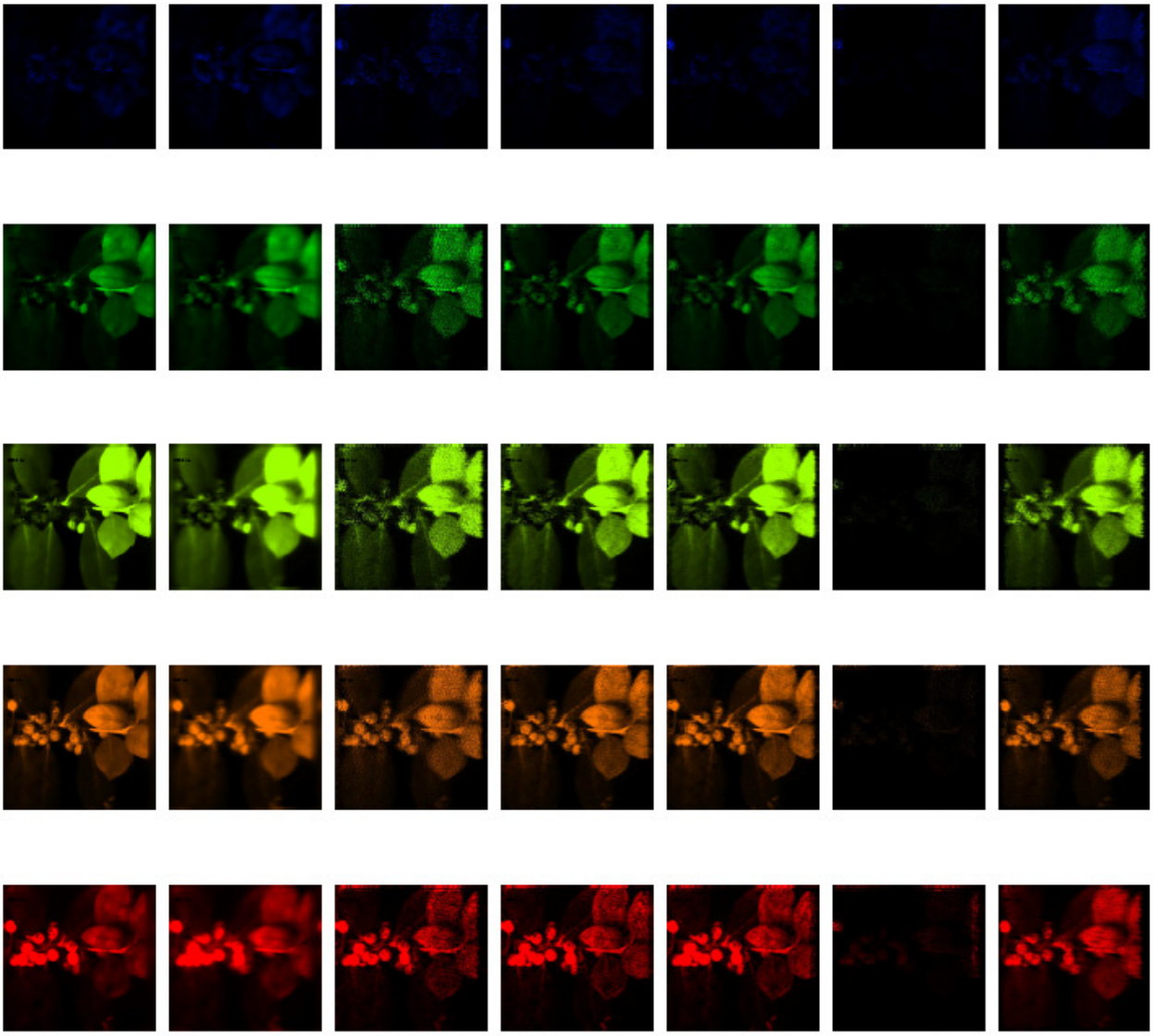


Fig. 12. From top to bottom, the rows correspond to channels 1, 12, 16, 20, 23 (wavelengths: 460, 522, 555, 598, 642 nm). In each row, Col. 1: 24-frame TwIST reconstruction, Col. 2: 4 frame TwIST reconstruction (PSNR 29.0), Cols. 3 to 5: BPFA reconstruction with 4, 8, 12 frames (PSNRs 29.03, 33.8, 34.26 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.