A Dictionary-Based Generalization of Robust PCA Part II: Applications to Hyperspectral Demixing

Sirisha Rambhatla, *Student Member, IEEE,* Xingguo Li, *Student Member, IEEE,* Jineng Ren, *Student Member, IEEE,* and Jarvis Haupt, *Senior Member, IEEE*

***Abstract*—We consider the task of localizing targets of interest in a hyperspectral (HS) image based on their spectral signature(s), by posing the problem as two distinct convex demixing task(s). With applications ranging from remote sensing to surveillance, this task of target detection leverages the fact that each material/object possesses its own characteristic spectral response, depending upon its composition. However, since *signatures* of different materials are often correlated, matched filtering-based approaches may not be apply here. To this end, we model a HS image as a superposition of a low-rank component and a dictionary sparse component, wherein the dictionary consists of the *a priori* known characteristic spectral responses of the target we wish to localize, and develop techniques for two different sparsity structures, resulting from different model assumptions. We also present the corresponding recovery guarantees, leveraging our recent theoretical results from a companion paper. Finally, we analyze the performance of the proposed approach via experimental evaluations on real HS datasets for a classification task, and compare its performance with related techniques.**

***Index Terms*—Hyperspectral imaging, Robust-PCA, dictionary sparse, target localization, and remote sensing.**

I. Introduction

Hyperspectral (HS) imaging is an imaging modality which senses the intensities of the reflected electromagnetic waves (responses) corresponding to different wavelengths of the electromagnetic spectra, often invisible to the human eye. As the spectral response associated with an object/material is dependent on its composition, HS imaging lends itself very useful in identifying the said target objects/materials via their characteristic spectra or *signature* responses, also referred to as *endmembers* in the literature. Typical applications of HS imaging range from monitoring agricultural use of land, catchment areas of rivers and water bodies, food processing and surveillance, to detecting various minerals, chemicals, and even presence of life sustaining compounds on distant planets; see [1], [2], and references therein for details. However, often, these spectral *signatures* are highly correlated, making it difficult to detect regions of interest based on these endmembers. In this work, we present two techniques to localize target materials/objects in a given HS image based on some structural assumptions on the data, using the *a priori* known signatures of the target of interest.

This work was supported by the DARPA YFA, Grant N66001-14-1-4047. Preliminary versions appeared in the proceedings of the 2016 IEEE Global Conference on Signal & Information Processing (GlobalSIP), 2017 Asilomar Conference on Signals, Systems, & Computers, and the 2018 IEEE International Conference on Acoustics, Speech & Signal Processing (ICASSP).

S. Rambhatla, J. Ren, and J. Haupt are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, 55455, USA e-mail: {rambh002, renxx282, jdhaupt}@umn.edu, respectively. X. Li is with the Computer Science Department, Princeton University, Princeton, NJ 08540, USA email: xingguol@cs.princeton.edu.

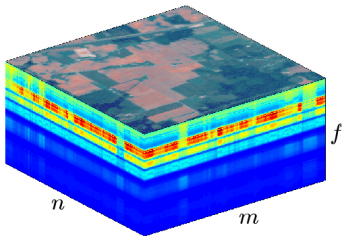


Fig. 1: The HS image data-cube corresponding to the Indian Pines dataset.

The primary property that enables us to localize a target is the approximate low-rankness of HS images when represented as a matrix, owing to the fact that a particular scene is composed of only a limited type of objects/materials [3]. For instance, while imaging an agricultural area, one would expect to record responses from materials like biomass, farm vehicles, roads, houses, water bodies, and so on. Moreover, the spectra of complex materials can be assumed to be a linear mixture of the constituent materials [3], [4], i.e. the received HS responses can be viewed as being generated by a linear mixture model [5]. For the target localization task at hand, this approximate low-rank structure is used to decompose a given HS image into a low-rank part, and a component that is sparse in a known dictionary – a *dictionary sparse* part– wherein the dictionary is composed of the spectral signatures of the target of interest. We begin by formalizing the specific model of interest in the next section.

A. Model

A HS sensor records the response of a region, which corresponds to a pixel in the HS image as shown in Fig. 1, to different frequencies of the electromagnetic spectrum. As a result, each HS image $\mathbf{I} \in \mathbb{R}^{n\times m\times f}$, can be viewed as a data-cube formed by stacking f matrices of size $n \times m$, as shown in Fig. 1. Therefore, each volumetric element or *voxel*, of a HS image is a vector of length f, and represents the response of the material to f measurement channels. Here, f is determined by the number of channels or frequency bands across which measurements of the reflectances are made.

Formally, let $\mathbf{M} \in \mathbb{R}^{f\times nm}$ be formed by *unfolding* the HS image $\mathbf{I}$, such that, each column of $\mathbf{M}$ corresponds to a voxel of the data-cube. We then model $\mathbf{M}$ as arising from a superposition of a low-rank component $\mathbf{L} \in \mathbb{R}^{f\times nm}$ with rank r, and a dictionary-sparse component, expressed as $\mathbf{DS}$, i.e.,

$$\mathbf{M} = \mathbf{L} + \mathbf{DS}. \tag{1}$$

Here, $\mathbf{D} \in \mathbb{R}^{f\times d}$ represents an *a priori* known dictionary composed of appropriately normalized characteristic responses of the material/object (or the constituents of the material), we

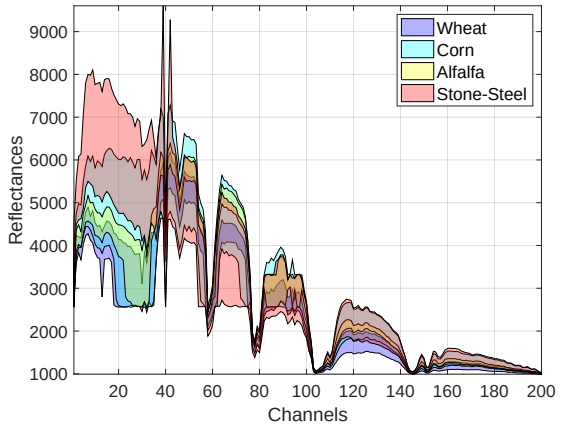


Fig. 2: Correlated spectral signatures. The spectral signatures of even different materials are highly correlated. Shown here are spectral signatures of classes from the Indian Pines dataset [14]. Here, the shaded region shows the lower and upper ranges of reflectance values the signatures take.

wish to localize, and $\mathbf{S} \in \mathbb{R}^{d \times nm}$ refers to the *sparse* coefficient matrix (also referred to as *abundances* in the literature). Note that $\mathbf{D}$ can also be constructed by learning a dictionary based on the known spectral signatures of a target; see [6]–[9].

B. Our Contributions

In this work, we present two techniques[1] for target detection in a HS image, depending upon different sparsity assumptions on the matrix $\mathbf{S}$, by modeling the data as shown in (1). Building on the theoretical results of [10]–[12], our techniques operate by forming the dictionary $\mathbf{D}$ using the *a priori* known spectral signatures of the target of interest, and leveraging the approximate low-rank structure of the data matrix $\mathbf{M}$ [13]. Here, the dictionary $\mathbf{D}$ can be formed from the *a priori* known signatures directly, or by learning an appropriate dictionary based on target data; see [6]–[9].

We consider two types of sparsity structures for the coefficient matrix $\mathbf{S}$, namely, a) *global* or *entry-wise* sparsity, wherein we let the matrix $\mathbf{S}$ have s_e non-zero entries globally, and b) *column-wise* sparse structure, where at most s_c columns of the matrix $\mathbf{S}$ have non-zero elements. The choice of a particular sparsity model depends on the properties of the dictionary matrix $\mathbf{D}$. In particular, if the target signature admits a sparse representation in the dictionary, entry-wise sparsity structure is preferred. This is likely to be the case when the dictionary is overcomplete ($f < d$) or *fat*, and also when the target spectral responses admit a sparse representation in the dictionary. On the other hand, the column-wise sparsity structure is amenable to cases where the representation can use all columns of the dictionary. This potentially arises in the cases when the dictionary is undercomplete ($f \geq d$) or *thin*. Note that, in the column-wise sparsity case, the non-zero columns need not be sparse themselves. The applicability of these two modalities is also exhibited in our experimental analysis; see Section V for further details. Further, we specialize the theoretical results of [12], to present the conditions under which such a demixing task will succeed under the two sparsity models discussed above; see also [10] and [11].

Next, we analyze the performance of the proposed techniques via extensive experimental evaluations on real-world demixing tasks over different datasets and dictionary choices, and compare the performance of the proposed techniques with related works. This demixing task is particularly challenging since the spectral signatures of even distinct classes are highly correlated to each other, as shown in Fig. 2. The shaded region here shows the upper and lower ranges of different classes. For instance, in Fig. 2 we observe that the spectral signature of the "Stone-Steel" class is similar to that of class "Wheat". This correlation between the spectral signatures of different classes results in an approximate low-rank structure of the data, captured by the low-rank component $\mathbf{L}$, while the dictionary-sparse component $\mathbf{DS}$ is used to identify the target of interest. We specifically show that such a decomposition successfully localizes the target despite the high correlation between spectral signatures of distinct classes.

Finally, it is worth noting that although we consider *thin* dictionaries ($f \geq d$) for the purposes of this work, since it is more suitable for the current exposition, our theoretical results are also applicable for the *fat* case ($f < d$); see [10], [11], and [12] for further details.

C. Prior Art

The model shown in (1) is closely related to a number of well-known problems. To start, in the absence of the dictionary sparse part $\mathbf{DS}$, (1) reduces to the popular problem of principal component analysis (PCA) [15], [16]. The problem considered here also shares its structure with variants of PCA, such as robust-PCA [17], [18] (with $\mathbf{D} = \mathbf{I}$ for an identity matrix $\mathbf{I}$,) outlier pursuit [19] (where $\mathbf{D} = \mathbf{I}$ and $\mathbf{S}$ is column-wise sparse,) and others [20]–[28].

On the other hand, the problem can be identified as that of sparse recovery [29]–[32], in the absence of the low-rank part $\mathbf{L}$. Following which, sparse recovery methods for analysis of HS images have been explored in [33]–[36]. In addition, in a recent work [37], the authors further impose a low-rank constraint on the coefficient matrix $\mathbf{S}$ for the demixing task. Further, applications of compressive sampling have been explored in [38], while [5] analyzes the case where HS images are noisy and incomplete. The techniques discussed above focus on identifying all materials in a given HS image. However, for target localization tasks, it is of interest to identify only specific target(s) in a given HS image. As a result, there is a need for techniques which localize targets based on their *a priori* known spectral signatures.

The model described in (1) was introduced in [39] as a means to detect traffic anomalies in a network, wherein, the authors focus on a case where the dictionary $\mathbf{D}$ is *overcomplete*, i.e., *fat*, and the rows of $\mathbf{D}$ are orthogonal, e.g., $\mathbf{R}\mathbf{R}^\top = \mathbf{I}$. Here, the coefficient matrix $\mathbf{S}$ is assumed to possess at most k nonzero elements per row and column, and s nonzero elements globally. In a recent work [10] and the accompanying theoretical work [12], we analyze the extension of [39] to include a case where the dictionary has more rows than columns, i.e., is *thin*, while removing the orthogonality constraint for both the *thin* and the

[1]The code is made available at `github.com/srambhatla/Dictionary-based-Robust-PCA`.

fat dictionary cases, when s is small. This case is particularly amenable for the target localization task at hand, since often we aim to localize targets based on a few *a priori* known spectral signatures. To this end, we focus our attention on the *thin* case, although a similar analysis applies for the *fat* case [10]; see also [12].

D. Related Techniques

To study the properties of our techniques, we compare and contrast their performance with related works. First, as a sanity check, we compare the performance of the proposed techniques with matched filtering-based methods (detailed in Section V). In addition, we compare the performance of our techniques to other closely related methods based on the sparsity assumptions on the matrix $\mathbf{S}$, as described below.

For entry-wise sparse structure: The first method we compare to is based on the observation that in cases where the known dictionary $\mathbf{D}$ is thin, we can multiply (1) on the left by the pseudo-inverse of $\mathbf{D}$, say $\mathbf{D}^\dagger$, in which case, the model shown in (1) reduces to that of robust PCA, i.e.,

$$\widetilde{\mathbf{M}} = \widetilde{\mathbf{L}} + \mathbf{S}, \qquad \text{(RPCA}^\dagger\text{)}$$

where $\widetilde{\mathbf{M}} = \mathbf{D}^\dagger\mathbf{M}$ and $\widetilde{\mathbf{L}} = \mathbf{D}^\dagger\mathbf{L}$. Therefore, in this case, we can recover the sparse matrix $\mathbf{S}$ by robust PCA [17], [18], and estimate the low-rank part using the estimate of $\mathbf{DS}$. Note that this is not applicable for the *fat* case due to the non-trivial null space of its pseudo-inverse.

Although at a first glance this seems like a reasonable technique, somewhat surprisingly, it does not succeed for all *thin* dictionaries. Specifically, in cases where r, the rank of $\mathbf{L}$, is greater than the number of dictionary elements d, the pseudo-inversed component $\widetilde{\mathbf{L}}$ is no longer "low-rank." In fact, since the notion of low-rankness is relative to the potential maximum rank of the component, $\widetilde{\mathbf{L}}$ can be close to full-rank. As a result, the robust PCA model shown in RPCA† is no longer applicable and the demixing task may not succeed; see our corresponding theoretical paper [12] for details.

Moreover, even in cases where RPCA† succeeds ($r < d$), our proposed one-shot procedure guarantees the recovery of the two components under some mild conditions, while the pseudo-inverse based procedure RPCA† will require a two-step procedure – one to recover the sparse coefficient matrix and other to recover the low-rank component – in addition to a non-trivial analysis of the interaction between $\mathbf{D}^\dagger$ and the low-rank part $\mathbf{L}$. This is also apparent from our experiments shown in Section V, which indicate that optimization based on the model in (1) is more *robust* as compared to RPCA† for the classification problem at hand across different choices of the dictionaries.

For column-wise sparse structure: The column-wise sparse structure of the matrix $\mathbf{S}$ results in a column-wise sparse structure of the dictionary-sparse component $\mathbf{DS}$. As a result, the model at hand is similar to that studied in OP [19]. Specifically, the OP technique is aimed at identifying the outlier columns in a given matrix. However, it fails in cases where the target of interest is not an outlier, as in case of HS data. On the other hand, since the proposed technique uses the dictionary $\mathbf{D}$ corresponding to the spectral signatures of the target of interest to guide the demixing procedure, it results in a spectral signature-driven technique for target localization. This distinction between the two procedures is also discussed in our corresponding theoretical work [12] Section V, and is exemplified by our experimental results shown in Section V.

Further, as in the entry-wise case, one can also envision a pseudo-inverse based procedure to identify the target of interest via OP [19] on the pseudo-inversed data (referred to as OP† in our discussion) i.e.,

$$\widetilde{\mathbf{M}} = \widetilde{\mathbf{L}} + \mathbf{S}, \qquad \text{(OP}^\dagger\text{)}$$

where $\widetilde{\mathbf{M}} = \mathbf{D}^\dagger\mathbf{M}$ and $\widetilde{\mathbf{L}} = \mathbf{D}^\dagger\mathbf{L}$, with $\mathbf{S}$ admitting a column-wise sparse structure. However, this variant of OP does not succeed when the rank of the low-rank component is greater than the number of dictionary elements, i.e., $r \geq d$, as in the previous case; see our related theoretical work for details [12] Section V.

The rest of the paper is organized as follows. We formulate the problem and introduce relevant theoretical quantities in Section II, followed by specializing the theoretical results for the current application in Section III. Next, in Section IV, we present the specifics of the algorithms for the two cases. In Section V, we describe the experimental set-up and demonstrate the applicability of the proposed approaches via extensive numerical simulations on real HS datasets for a classification task. Finally, we conclude this discussion in Section VI.

Notation: Given a matrix $\mathbf{X}$, $\mathbf{X}_i$ denotes its i-th column and $\mathbf{X}_{i,j}$ denotes the (i,j) element of $\mathbf{X}$. We use $\|\mathbf{X}\| := \sigma_{\max}(\mathbf{X})$ for the spectral norm, where $\sigma_{\max}(\mathbf{X})$ denotes the maximum singular value of the matrix, $\|\mathbf{X}\|_\infty := \max_{i,\,j} |\mathbf{X}_{ij}|$, $\|\mathbf{X}\|_{\infty,\infty} := \max_i \|\mathbf{e}_i^\top \mathbf{X}\|_1$, and $\|\mathbf{X}\|_{\infty,2} := \max_i \|\mathbf{X}\mathbf{e}_i\|$, where $\mathbf{e}_i$ denotes the canonical basis vector with 1 at the i-th location and 0 elsewhere. Further, we denote the ℓ_2-norm of a vector $\mathbf{x}$ as $\|\mathbf{x}\|$. In addition, $\|.\|_*$, $\|.\|_1$, and $\|.\|_{1,2}$ refer to the nuclear norm, entry-wise ℓ_1- norm, and $\ell_{1,2}$ norm (sum of the ℓ_2 norms of the columns) of the matrix, respectively, which serve as convex relaxations of rank, sparsity, and column-wise sparsity inducing optimization, respectively.

II. Problem Formulation

In this section, we introduce the optimization problem of interest and different theoretical quantities pertinent to our analysis.

A. Optimization problems

Our aim is to recover the low-rank component $\mathbf{L}$ and the sparse coefficient matrix $\mathbf{S}$, given the dictionary $\mathbf{D}$ and samples $\mathbf{M}$ generated according to the model shown in (1). Here the coefficient matrix $\mathbf{S}$ can either have an entry-wise sparse structure or a column-wise sparse structure. We now crystallize our model assumptions to formulate appropriate convex optimization problems for the two sparsity structures.

Specifically, depending upon the priors about the sparsity structure of $\mathbf{S}$, and the low-rank property of the component $\mathbf{L}$,

we aim to solve the following convex optimization problems, i.e.,

$$\min_{\mathbf{L},\mathbf{S}} \|\mathbf{L}\|_* + \lambda_e \|\mathbf{S}\|_1 \quad \text{s.t.} \quad \mathbf{M} = \mathbf{L} + \mathbf{DS} \qquad \text{(D-RPCA(E))}$$

for the entry-wise sparsity case, and

$$\min_{\mathbf{L},\mathbf{S}} \|\mathbf{L}\|_* + \lambda_c \|\mathbf{S}\|_{1,2} \text{ s.t. } \mathbf{M} = \mathbf{L} + \mathbf{DS} \qquad \text{(D-RPCA(C))}$$

for the column-wise sparse case, to recover $\mathbf{L}$ and $\mathbf{S}$ with regularization parameters $\lambda_e \geq 0$ and $\lambda_c \geq 0$, given the data $\mathbf{M}$ and the dictionary $\mathbf{D}$. Here, the *a priori* known dictionary $\mathbf{D}$ is assumed to be undercomplete (*thin*, i.e., $d \leq f$) for the application at hand. Analysis of a more general case can be found in [12]. Further, here "D-RPCA" refers to "dictionary based robust principal component analysis", while the qualifiers "E" and "C" indicate the entry-wise and column-wise sparsity patterns, respectively.

Note that, in the column-wise sparse case there is an inherent ambiguity regarding the recovery of the true component pairs $(\mathbf{L}, \mathbf{S})$ corresponding to the low-rank part and the dictionary sparse component, respectively. Specifically, any pair $(\mathbf{L}_0, \mathbf{S}_0)$ satisfying $\mathbf{M} = \mathbf{L}_0 + \mathbf{DS}_0 = \mathbf{L} + \mathbf{DS}$, where $\mathbf{L}_0$ and $\mathbf{L}$ have the same column space, and $\mathbf{S}_0$ and $\mathbf{S}$ have the identical column support, is a solution of D-RPCA(C). To this end, we define the following *oracle model* to characterize the optimality of any solution pair $(\mathbf{L}_0, \mathbf{S}_0)$.

Definition D.1 (Oracle Model for Column-wise Sparse Case)**.** Let the pair $(\mathbf{L}, \mathbf{S})$ be the matrices forming the data $\mathbf{M}$ as per (1), define the corresponding oracle model $\{\mathbf{M}, \mathcal{U}, \mathcal{I}_{\mathcal{S}_c}\}$. Then, any pair $(\mathbf{L}_0, \mathbf{S}_0)$ is in the *Oracle Model* $\{\mathbf{M}, \mathcal{U}, \mathcal{I}_{\mathcal{S}_c}\}$, if $\mathcal{P}_{\mathcal{U}}(\mathbf{L}_0) = \mathbf{L}$, $\mathcal{P}_{\mathcal{S}_c}(\mathbf{DS}_0) = \mathbf{DS}$ and $\mathbf{L}_0 + \mathbf{DS}_0 = \mathbf{L} + \mathbf{DS} = \mathbf{M}$ hold simultaneously, where $\mathcal{P}_{\mathcal{U}}$ and $\mathcal{P}_{\mathcal{S}_c}$ are projections onto the column space $\mathcal{U}$ of $\mathbf{L}$ and column support $\mathcal{I}_{\mathcal{S}_c}$ of $\mathbf{S}$, respectively.

For this case, we then first establish the sufficient conditions for the existence of a solution based on some incoherence conditions. Following which, our main result for the column-wise case states the sufficient conditions under which solving a convex optimization problem recovers a solution pair $(\mathbf{L}_0, \mathbf{S}_0)$ in the oracle model.

B. Conditions on the Dictionary

For our analysis, we require that the dictionary $\mathbf{D}$ follows the *generalized frame property* (GFP) defined as follows.

Definition D.2. A matrix $\mathbf{D}$ satisfies the *generalized frame property* (GFP), on vectors $\mathbf{v} \in \mathcal{R}$, if for any fixed vector $\mathbf{v} \in \mathcal{R}$ where $\mathbf{v} \neq \mathbf{0}$, we have

$$\alpha_\ell \|\mathbf{v}\|_2^2 \leq \|\mathbf{Dv}\|_2^2 \leq \alpha_u \|\mathbf{v}\|_2^2,$$

where α_ℓ and α_u are the lower and upper *generalized frame bounds* with $0 < \alpha_\ell \leq \alpha_u < \infty$.

The GFP is met as long as the vector $\mathbf{v}$ is not in the null-space of the matrix $\mathbf{D}$, and $\|\mathbf{D}\|$ is bounded. Therefore, for the *thin* dictionary setting $d < n$ for both entry-wise and column-wise sparsity cases, this condition is satisfied as long as $\mathbf{D}$ has a full column rank, and $\mathcal{R}$ can be the entire space. For example, $\mathbf{D}$ being a *frame* [40] suffices; see [41] for a brief overview of frames.

C. Relevant Subspaces

Before we define the relevant subspaces for this discussion, we define a few preliminaries. First, let the pair $(\mathbf{L_0}, \mathbf{S_0})$ be the solution to D-RPCA(E) (the entry-wise sparse case), and for the column-wise sparse case, let the pair $(\mathbf{L_0}, \mathbf{S_0})$ be in the oracle model $\{\mathbf{M}, \mathcal{U}, \mathcal{I}_{\mathcal{S}_c}\}$; see Definition **D.1**.

Next, for the low-rank matrix $\mathbf{L}$, let the compact singular value decomposition (SVD) be represented as

$$\mathbf{L} = \mathbf{U\Sigma V}^\top,$$

where $\mathbf{U} \in \mathbb{R}^{f \times r}$ and $\mathbf{V} \in \mathbb{R}^{nm \times r}$ are the left and right singular vectors of $\mathbf{L}$, respectively, and $\mathbf{\Sigma}$ is a diagonal matrix with singular values arranged in a descending order on the diagonal. Here, matrices $\mathbf{U}$ and $\mathbf{V}$ each have orthogonal columns. Further, let $\mathcal{L}$ be the linear subspace consisting of matrices spanning the same row or column space as $\mathbf{L}$, i.e.,

$$\mathcal{L} := \{\mathbf{UW}_1^\top + \mathbf{W}_2\mathbf{V}^\top, \mathbf{W}_1 \in \mathbb{R}^{nm \times r}, \mathbf{W}_2 \in \mathbb{R}^{f \times r}\}.$$

Next, let $\mathcal{S}_e$ ($\mathcal{S}_c$) be the space spanned by $d \times nm$ matrices with the same non-zero support (column support, denoted as csupp) as $\mathbf{S}$, and let $\mathcal{D}$ be defined as

$$\mathcal{D} := \{\mathbf{DH}\}, \text{where} \begin{cases} \mathbf{H} \in \mathcal{S}_e \text{ for entry-wise case,} \\ \text{csupp}(\mathbf{H}) \subseteq \mathcal{I}_{\mathcal{S}_c} \text{ for column-wise case.} \end{cases}$$

Here, $\mathcal{I}_{\mathcal{S}_c}$ denotes the index set containing the non-zero column indices of $\mathbf{S}$ for the column-wise sparsity case. In addition, we denote the corresponding complements of the spaces described above by appending '$\perp$'.

We use calligraphic '$\mathcal{P}(\cdot)$' to denote the projection operator onto a subspace defined by the subscript, and '$\mathbf{P}$' to denote the corresponding projection matrix with the appropriate subscripts. Therefore, using these definitions the projection operators onto and orthogonal to the subspace $\mathcal{L}$ are defined as

$$\mathcal{P}_{\mathcal{L}}(\mathbf{L}) = \mathbf{P_U L} + \mathbf{L P_V} - \mathbf{P_U L P_V}$$

and

$$\mathcal{P}_{\mathcal{L}^\perp}(\mathbf{L}) = (\mathbf{I} - \mathbf{P_U})\mathbf{L}(\mathbf{I} - \mathbf{P_V}),$$

respectively.

D. Incoherence Measures

We also employ various notions of incoherence to identify the conditions under which our procedures succeed. To this end, we first define the incoherence parameter μ that characterizes the relationship between the low-rank part $\mathbf{L}$ and the dictionary sparse part $\mathbf{DS}$, as

$$\mu := \max_{\mathbf{Z} \in \mathcal{D} \setminus \{\mathbf{0}_{d \times nm}\}} \frac{\|\mathcal{P}_{\mathcal{L}}(\mathbf{Z})\|_{\text{F}}}{\|\mathbf{Z}\|_{\text{F}}}. \tag{2}$$

The parameter $\mu \in [0, 1]$ is the measure of degree of similarity between the low-rank part and the dictionary sparse component. Here, a larger μ implies that the dictionary sparse component is close to the low-rank part. In addition, we also define the parameter β_U as

$$\beta_{\mathbf{U}} := \max_{\|\mathbf{u}\|=1} \frac{\|(\mathbf{I}-\mathbf{P_U})\mathbf{Du}\|^2}{\|\mathbf{Du}\|^2}, \tag{3}$$

which measures the similarity between the orthogonal complement of the column-space $\mathcal{U}$ and the dictionary $\mathbf{D}$.

The next two measures of incoherence can be interpreted as a way to identify the cases where for $\mathbf{L}$ with SVD as

$\mathbf{L} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$: (a) $\mathbf{U}$ resembles the dictionary $\mathbf{D}$, and (b) $\mathbf{V}$ resembles the sparse coefficient matrix $\mathbf{S}$. In these cases, the low-rank part may resemble the dictionary sparse component. To this end, similar to [39], we define the following measures to identify these cases as

$$\text{(a) } \gamma_{\mathbf{U}} := \max_i \frac{\|\mathbf{P}_{\mathbf{U}}\mathbf{D}\mathbf{e}_i\|^2}{\|\mathbf{D}\mathbf{e}_i\|^2} \text{ and (b) } \gamma_{\mathbf{V}} := \max_i \|\mathbf{P}_{\mathbf{V}}\mathbf{e}_i\|^2. \quad (4)$$

Here, $0 \leq \gamma_{\mathbf{U}} \leq 1$ achieves the upper bound when a dictionary element is exactly aligned with the column space $\mathcal{U}$ of the $\mathbf{L}$, and lower bound when all of the dictionary elements are orthogonal to $\mathcal{U}$. Moreover, $\gamma_{\mathbf{V}} \in [r/nm, 1]$ achieves the upper bound when the row-space of $\mathbf{L}$ is "spiky", i.e., a certain row of $\mathbf{V}$ is 1-sparse, meaning that a column of $\mathbf{L}$ is supported by (can be expressed as a linear combination of) a column of $\mathbf{U}$. The lower bound here is attained when it is "spread-out", i.e., each column of $\mathbf{L}$ is a linear combination of all columns of $\mathbf{U}$. In general, our recovery of the two components is easier when the incoherence parameters $\gamma_{\mathbf{U}}$ and $\gamma_{\mathbf{V}}$ are closer to their lower bounds. In addition, for notational convenience, we define constants

$$\xi_e := \|\mathbf{D}^\top\mathbf{U}\mathbf{V}^\top\|_\infty \text{ and } \xi_c := \|\mathbf{D}^\top\mathbf{U}\mathbf{V}^\top\|_{\infty,2}. \quad (5)$$

Here, ξ_e is the maximum absolute entry of $\mathbf{D}^\top\mathbf{U}\mathbf{V}^\top$, which measures how close columns of $\mathbf{D}$ are to the singular vectors of $\mathbf{L}$. Similarly, for the column-wise case, ξ_c measures the closeness of columns of $\mathbf{D}$ to the singular vectors of $\mathbf{L}$ under a different metric (column-wise maximal ℓ_2-norm).

III. Theoretical Results

In this section, we specialize our theoretical results [12] for the HS demixing task. Specifically, we provide the main results corresponding to each sparsity structure of $\mathbf{S}$ for the thin dictionary case considered here. We start with the theoretical results for the entry-wise sparsity case, and then present the corresponding theoretical guarantees for the column-wise sparsity structure; see [12] for detailed proofs.

A. Exact Recovery for Entry-wise Sparsity Case

For the entry-wise case, our main result establishes the existence of a regularization parameter λ_e, for which solving the optimization problem D-RPCA(E) will recover the components $\mathbf{L}$ and $\mathbf{S}$ exactly. To this end, we will show that such a λ_e belongs to a non-empty interval $[\lambda_e^{\min}, \lambda_e^{\max}]$, where $\lambda_e^{\min}$ and $\lambda_e^{\max}$ are defined as

$$\lambda_e^{\min} := \tfrac{1+C_e}{1-C_e}\,\xi_e \text{ and } \lambda_e^{\max} := \tfrac{\sqrt{\alpha_\ell}(1-\mu)-\sqrt{r\alpha_u}\mu}{\sqrt{s_e}}. \quad (6)$$

Here, $C_e(\alpha_u, \alpha_\ell, \gamma_{\mathbf{U}}, \gamma_{\mathbf{V}}, s_e, d, k, \mu)$ where $0 \leq C_e < 1$ is a constant that captures the relationship between different model parameters, and is defined as

$$C_e := \tfrac{c}{\alpha_\ell(1-\mu)^2 - c},$$

where $c = \frac{\alpha_u}{2}((1 + 2\gamma_{\mathbf{U}})(\min(s_e, d) + s_e\gamma_{\mathbf{V}}) + 2\gamma_{\mathbf{V}}\min(s_e, nm)) - \frac{\alpha_\ell}{2}(\min(s_e, d) + s_e\gamma_{\mathbf{V}})$. Given these definitions, we have the following result for the entry-wise sparsity structure.

Theorem 1. *Suppose* $\mathbf{M} = \mathbf{L} + \mathbf{D}\mathbf{S}$, *where* $\operatorname{rank}(\mathbf{L}) = r$ *and* $\mathbf{S}$ *has at most* s_e *non-zeros, i.e.,* $\|\mathbf{S}\|_0 \leq s_e \leq s_e^{\max} := \frac{(1-\mu)^2}{2}\frac{nm}{r}$, *and the dictionary* $\mathbf{D} \in \mathbb{R}^{f\times d}$ *for* $d \leq f$ *obeys the generalized frame property* (2) *with frame bounds* $[\alpha_\ell, \alpha_u]$, *where* $0 < \alpha_\ell \leq \frac{1}{(1-\mu)^2}$, *and* $\gamma_{\mathbf{U}}$ *follows*

$$\gamma_{\mathbf{U}} \leq \begin{cases} \frac{(1-\mu)^2 - 2s_e\gamma_{\mathbf{V}}}{2s_e(1+\gamma_{\mathbf{V}})}, & \text{for } s_e \leq \min\ (d, s_e^{\max}) \\ \frac{(1-\mu)^2 - 2s_e\gamma_{\mathbf{V}}}{2(d+s_e\gamma_{\mathbf{V}})}, & \text{for } d < s_e \leq s_e^{\max}. \end{cases} \quad (7)$$

Then given $\mu \in [0, 1]$, $\gamma_{\mathbf{U}}$ *and* $\gamma_{\mathbf{V}} \in [r/nm, 1]$, *and* ξ_e *defined in* (2), (4), (5), *respectively,* $\lambda_e \in [\lambda_e^{\min}, \lambda_e^{\max}]$ *with* $\lambda_e^{\max} > \lambda_e^{\min} \geq 0$ *defined in* (6), *solving D-RPCA(E) will recover matrices* $\mathbf{L}$ *and* $\mathbf{S}$.

We observe that the conditions for the recovery of $(\mathbf{L}, \mathbf{S})$ are closely related to the incoherence measures (μ, $\gamma_{\mathbf{V}}$, and $\gamma_{\mathbf{U}}$) between the low-rank part, $\mathbf{L}$, the dictionary, $\mathbf{D}$, and the sparse component $\mathbf{S}$. In general, smaller sparsity, rank, and incoherence parameters are sufficient for ensuring the recovery of the components for a particular problem. This is in line with our intuition that the more distinct the two components, the easier it should be to tease them apart. For our HS demixing problem, this indicates that a target of interest can be localized as long as its the spectral signature is appropriately different from the other materials in the scene.

B. Recovery for Column-wise Sparsity Case

For the column-wise sparsity model, recall that any pair in the oracle model described in **D.1** is considered optimal. To this end, we first establish the sufficient conditions for the existence of such an optimal pair $(\mathbf{L}_0, \mathbf{S}_0)$ by the following lemma.

Lemma 2. *Given* $\mathbf{M}$, $\mathbf{D}$, *and* $(\mathcal{L}, \mathcal{S}_c, \mathcal{D})$, *any pair* $(\mathbf{L}_0, \mathbf{S}_0) \in \{\mathbf{M}, \mathcal{U}, \mathcal{I}_{\mathcal{S}_c}\}$ *satisfies* $\operatorname{span}\{col(\mathbf{L}_0)\} = \mathcal{U}$ *and* $csupp(\mathbf{S}_0) = \mathcal{I}_{\mathcal{S}_c}$ *if* $\mu < 1$.

In essence, we need the incoherence parameter μ to be strictly smaller than 1. Next, analogous to the entry-wise case, we show that λ_c belongs to a non-empty interval $[\lambda_c^{\min}, \lambda_c^{\max}]$, using which solving D-RPCA(C) recovers an optimal pair in the oracle model **D.1** in accordance with Lemma 2. Here, for a constant $C_c := \frac{\alpha_u}{\alpha_\ell}\frac{1}{(1-\mu)^2}\gamma_{\mathbf{V}}\beta_{\mathbf{U}}$, $\lambda_c^{\min}$ and $\lambda_c^{\max}$ are defined as

$$\lambda_c^{\min} := \tfrac{\xi_c + \sqrt{rs_c\alpha_u}\mu C_c}{1 - s_c C_c} \text{ and } \lambda_c^{\max} := \tfrac{\sqrt{\alpha_\ell}(1-\mu) - \sqrt{r\alpha_u}\mu}{\sqrt{s_c}}. \quad (8)$$

This leads us to the following result for the column-wise case.

Theorem 3. *Suppose* $\mathbf{M} = \mathbf{L} + \mathbf{D}\mathbf{S}$ *with* $(\mathbf{L}, \mathbf{S})$ *defining the oracle model* $\{\mathbf{M}, \mathcal{U}, \mathcal{I}_{\mathcal{S}_c}\}$, *where* $\operatorname{rank}(\mathbf{L}) = r$, $|\mathcal{I}_{\mathcal{S}_c}| = s_c$ *for* $s_c \leq s_c^{\max} := \frac{\alpha_\ell}{\alpha_u\gamma_{\mathbf{V}}} \cdot \frac{(1-\mu)^2}{\beta_{\mathbf{U}}}$. *Given* $\mu \in [0, 1)$, $\beta_{\mathbf{U}}$, $\gamma_{\mathbf{V}} \in [r/nm, 1]$, ξ_c *as defined in* (2), (3), (4), (5), *respectively, and any* $\lambda_c \in [\lambda_c^{\min}, \lambda_c^{\max}]$, *for* $\lambda_c^{\max} > \lambda_c^{\min} \geq 0$ *defined in* (8), *solving D-RPCA(C) will recover a pair of components* $(\mathbf{L}_0, \mathbf{S}_0) \in \{\mathbf{M}, \mathcal{U}, \mathcal{I}_{\mathcal{S}_c}\}$, *if the dictionary* $\mathbf{D} \in \mathbb{R}^{f\times d}$ *obeys the generalized frame property* ***D.2*** *with frame bounds* $[\alpha_\ell, \alpha_u]$, *for* $\alpha_\ell > 0$.

Theorem 3 outlines the sufficient conditions under which the solution to the optimization problem D-RPCA(C) will be in the oracle model defined in **D.1**. Here, for a case where $1 \lesssim \alpha_l \leq \alpha_u \lesssim 1$, which can be easily met by a tight frame when $f > d$, constant $\frac{(1-\mu)^2}{\beta_U}$, and $\gamma_{\mathbf{V}} = \Theta(\frac{r}{nm})$, we have

Algorithm 1 APG Algorithm for D-RPCA(E) and D-RPCA(C), adapted from [39]

Input: $\mathbf{M}$, $\mathbf{D}$, λ, v, ν_0, $\bar{\nu}$, and $L_f = \lambda_{max}\left([\mathbf{I}\ \mathbf{D}]^\top[\mathbf{I}\ \mathbf{D}]\right)$
Initialize: $\mathbf{L}[0] = \mathbf{L}[-1] = \mathbf{0}_{L\times T}$, $\mathbf{S}[0] = \mathbf{S}[-1] = \mathbf{0}_{F\times T}$, $t[0] = t[-1] = 1$, and set $k = 0$.
while not converged **do**

Generate points $\mathbf{T}_L[k]$ and $\mathbf{T}_S[k]$ using momentum:

$$\mathbf{T}_L[k] = \mathbf{L}[k] + \tfrac{t[k-1]-1}{t[k]}(\mathbf{L}[k] - \mathbf{L}[k-1]),$$
$$\mathbf{T}_S[k] = \mathbf{S}[k] + \tfrac{t[k-1]-1}{t[k]}(\mathbf{S}[k] - \mathbf{S}[k-1]).$$

Take a gradient step using these points :

$$\mathbf{G}_L[k] = \mathbf{T}_L[k] + \tfrac{1}{L_f}(\mathbf{M} - \mathbf{T}_L[k] - \mathbf{D}\mathbf{T}_S[k]),$$
$$\mathbf{G}_S[k] = \mathbf{T}_S[k] + \tfrac{1}{L_f}\mathbf{D}^\top(\mathbf{M} - \mathbf{T}_L[k] - \mathbf{D}\mathbf{T}_S[k]).$$

Update Low-rank part via singular value thresholding:

$$\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top = \text{svd}(\mathbf{G}_L[k]),$$
$$\mathbf{L}[k+1] = \mathbf{U}\mathcal{S}_{\nu[k]/L_f}(\mathbf{\Sigma})\mathbf{V}^\top.$$

Update the Dictionary Sparse part:

$$\mathbf{S}[k+1] = \begin{cases} \mathcal{S}_{\nu[k]\lambda_e/L_f}(\mathbf{G}_S[k]), & \text{for D-RPCA(E)},\\ \mathcal{C}_{\nu[k]\lambda_c/L_f}(\mathbf{G}_S[k]), & \text{for D-RPCA(C)}. \end{cases}$$

Update the momentum term parameter $t[k+1]$:

$$t[k+1] = \tfrac{1+\sqrt{4t^2[k]+1}}{2}.$$

Update the continuation parameter $\nu[k+1]$:

$$\nu[k+1] = \max\{v\nu[k], \bar{\nu}\}.$$

$k \leftarrow k+1$
end while
return $\mathbf{L}[k]$, $\mathbf{S}[k]$

$s_c^{\max} = \mathcal{O}(\frac{nm}{r})$, which is of same order as in the Outlier Pursuit (OP) [19]. Moreover, our numerical results in [12] show that D-RPCA(C) can be much more robust than OP, and may recover $\{\mathcal{U}, \mathcal{I}_\mathbf{C}\}$ even when the rank of $\mathbf{L}$ is high and the number of outliers s_c is a constant proportion of m. This implies that, D-RPCA(C) will succeed as long as the dictionary $\mathbf{D}$ can successfully represent the target of interest while rejecting the columns of the data matrix $\mathbf{M}$ corresponding to materials other than the target.

IV. Algorithmic Considerations

The optimization problems of interest, D-RPCA(E) and D-RPCA(C), for the entry-wise and column-wise case, respectively, are convex but non-smooth. To solve for the components of interest, we adopt the accelerated proximal gradient (APG) algorithm, as shown in Algorithm 1. Note that [39] also applied the APG algorithm for D-RPCA(E), and we present a unified algorithm for both sparsity cases for completeness.

A. Background

The APG algorithm is motivated from a long line of work starting with [42], which showed the existence of a first order algorithm with a convergence rate of $\mathcal{O}(1/k^2)$ for a smooth convex objective, where k denotes the iterations. Following this, [43] developed the popular fast iterative shrinkage-thresholding algorithm (FISTA) which achieves this convergence rate for convex non-smooth objectives by accelerating the proximal gradient descent algorithm using a *momentum term* (the term $\frac{t[k-1]-1}{t[k]}$ in Algorithm 1) as prescribed by [42]. As a result, it became a staple to solve a wide range of convex non-smooth tasks including matrix completion [44], and robust PCA [45] and its variants [19], [39]. Also, recently [46] has shown further improvements in the rate of convergence.

In addition to the momentum term, the APG procedure operates by evaluating the gradient at a point further in the direction pointed by the negative gradient. Along with faster convergence, this insight about the next point minimizes the oscillations around the optimum point; see [43] and references therein.

B. Discussion of Algorithm 1

For the optimization problem of interest, we solve an unconstrained problem by transforming the equality constraint to a least-square term which penalizes the fit. In particular, the problems of interest we will solve via the APG algorithm are given by

$$\min_{\mathbf{L},\mathbf{S}} \ \nu\|\mathbf{L}\|_* + \nu\lambda_e\|\mathbf{S}\|_1 + \tfrac{1}{2}\|\mathbf{M} - \mathbf{L} - \mathbf{D}\mathbf{S}\|_\text{F}^2 \qquad (9)$$

for the entry-wise sparsity case, and

$$\min_{\mathbf{L},\mathbf{S}} \ \nu\|\mathbf{L}\|_* + \nu\lambda_c\|\mathbf{S}\|_{1,2} + \tfrac{1}{2}\|\mathbf{M} - \mathbf{L} - \mathbf{D}\mathbf{S}\|_\text{F}^2, \qquad (10)$$

for the column-wise sparsity case. We note that although for the application at hand, the thin dictionary case with ($f \geq d$) might be more useful in practice, Algorithm 1 allows for the use of fat dictionaries ($f < d$) as well.

Algorithm 1 also employs a continuation technique [45], which can be viewed as a "warm start" procedure. Here, we initialize the parameter ν_0 at some large value and geometrically reduced until it reaches a value $\bar{\nu}$. A smaller choice of $\bar{\nu}$ results in a solution which is closer to the optimal solution of the constrained problem. Further, as ν approaches zero, (9) and (10) recover the optimal solution of D-RPCA(E) and D-RPCA(C), respectively. Moreover, Algorithm 1 also utilizes the knowledge of the smoothness constant L_f (the Lipschitz constant of gradient) to set the step-size parameter.

Specifically, the APG algorithm requires that the gradient of the smooth part,

$$f(\mathbf{L},\mathbf{S}) := \tfrac{1}{2}\|\mathbf{M} - \mathbf{L} - \mathbf{D}\mathbf{S}\|_\text{F}^2 = \tfrac{1}{2}\|\mathbf{M} - \begin{bmatrix}\mathbf{I} & \mathbf{D}\end{bmatrix}\begin{bmatrix}\mathbf{L}\\ \mathbf{S}\end{bmatrix}\|_\text{F}^2$$

of the convex objectives shown in (9) and (10) is Lipschitz continuous with minimum Lipschitz constant L_f. Now, since the gradient $\nabla f(\mathbf{L},\mathbf{S})$ with respect to $\begin{bmatrix}\mathbf{L} & \mathbf{S}\end{bmatrix}^\top$ is given by

$$\nabla f(\mathbf{L},\mathbf{S}) = \begin{bmatrix}\mathbf{I} & \mathbf{D}\end{bmatrix}^\top(\mathbf{M} - \begin{bmatrix}\mathbf{I} & \mathbf{D}\end{bmatrix}\begin{bmatrix}\mathbf{L}\\ \mathbf{S}\end{bmatrix}),$$

we have that the gradient ∇f is Lipschitz continuous as

$$\|\nabla f(\mathbf{L}_1,\mathbf{S}_1) - \nabla f(\mathbf{L}_2,\mathbf{S}_2)\| \leq L_f\|\begin{bmatrix}\mathbf{L}_1\\ \mathbf{S}_1\end{bmatrix} - \begin{bmatrix}\mathbf{L}_2\\ \mathbf{S}_2\end{bmatrix}\|,$$

where

$$L_f = \|\begin{bmatrix}\mathbf{I} & \mathbf{D}\end{bmatrix}^\top\begin{bmatrix}\mathbf{I} & \mathbf{D}\end{bmatrix}\| = \lambda_{\max}(\begin{bmatrix}\mathbf{I} & \mathbf{D}\end{bmatrix}^\top\begin{bmatrix}\mathbf{I} & \mathbf{D}\end{bmatrix}),$$

TABLE I: Entry-wise sparsity model for the Indian Pines Dataset. Simulation results are presented for our proposed approach (D-RPCA(E)), robust-PCA based approach on transformed data $\mathbf{D}^\dagger\mathbf{M}$ (RPCA†), matched filtering (MF) on original data $\mathbf{M}$, and matched filtering on transformed data $\mathbf{D}^\dagger\mathbf{M}$ (MF†), across dictionary elements d, and the regularization parameter for initial dictionary learning procedure ρ; see Algorithm 2 . Threshold selects columns with column-norm greater than threshold such that AUC is maximized. For each case, the best performing metrics are reported in bold for readability. Further, " $*$ " denotes the case where ROC curve was "flipped" (i.e. classifier output was inverted to achieve the best performance).

(a) Learned dictionary, $d = 4$

d	ρ	Method	Threshold	Performance at best operating point		AUC
				TPR	FPR	
4	0.01	**D-RPCA(E)**	0.300	**0.979**	**0.023**	**0.989**
		RPCA†	0.650	0.957	0.049	0.974
		MF$_*$	N/A	0.957	0.036	0.994
		MF$_*^\dagger$	N/A	0.914	0.104	0.946
	0.1	**D-RPCA(E)**	0.800	**0.989**	0.017	0.997
		RPCA†	0.800	0.989	0.014	0.997
		MF	N/A	0.989	0.016	0.998
		MF†	N/A	0.989	**0.010**	**0.998**
	0.5	**D-RPCA(E)**	0.600	**0.968**	**0.031**	**0.991**
		RPCA†	0.600	0.935	0.067	0.988
		MF	N/A	0.548	0.474	0.555
		MF$_*^\dagger$	N/A	0.849	0.119	0.939

(b) Learned dictionary, $d = 10$

d	ρ	Method	Threshold	Performance at best operating point		AUC
				TPR	FPR	
10	0.01	**D-RPCA(E)**	0.600	0.935	0.060	0.972
		RPCA†	0.700	**0.978**	**0.023**	**0.990**
		MF$_*$	N/A	0.624	0.415	0.681
		MF$_*^\dagger$	N/A	0.569	0.421	0.619
	0.1	**D-RPCA(E)**	0.500	**0.968**	**0.029**	**0.993**
		RPCA†	0.500	0.871	0.144	0.961
		MF$_*$	N/A	0.688	0.302	0.713
		MF†	N/A	0.527	0.469	0.523
	0.5	**D-RPCA(E)**	1.000	**0.978**	**0.031**	**0.996**
		RPCA†	2.200	0.849	0.113	0.908
		MF	N/A	0.807	0.309	0.781
		MF$_*^\dagger$	N/A	0.527	0.465	0.539

(c) Dictionary by sampling voxels, $d = 15$

d	Method	Threshold	Performance at best operating point		AUC
			TPR	FPR	
15	**D-RPCA(E)**	0.300	**0.989**	**0.021**	**0.998**
	RPCA†	3.000	0.849	0.146	0.900
	MF	N/A	0.957	0.085	0.978
	MF†	N/A	0.796	0.217	0.857

(d) Average performance

Method	TPR		FPR		AUC	
	Mean	St.Dev.	Mean	St.Dev.	Mean	St.Dev.
D-RPCA(E)	**0.972**	**0.019**	**0.030**	**0.014**	**0.991**	**0.009**
RPCA†	0.919	0.061	0.079	0.055	0.959	0.040
MF	0.796	0.179	0.234	0.187	0.814	0.178
MF†	0.739	0.195	0.258	0.192	0.775	0.207

as shown in Algorithm 1.

The update of the low-rank component and the sparse matrix $\mathbf{S}$ for the entry-wise case, both involve a soft thresholding step, $\mathcal{S}_\tau(.)$, where for a matrix $\mathbf{Y}$, $\mathcal{S}_\tau(\mathbf{Y}_{ij})$ is defined as

$$\mathcal{S}_\tau(\mathbf{Y}_{ij}) = \text{sgn}(\mathbf{Y}_{ij}) \max(|\mathbf{Y}_{ij} - \tau|, 0).$$

In case of the low-rank part we apply this function to the singular values (therefore referred to as *singular value thresholding*) [44], while for the update of the dictionary sparse component, we apply it to the sparse coefficient matrix $\mathbf{S}$.

The low-rank update step for the column-wise case remains the same as for the entry-wise case. However, for the update of the column-wise case we threshold the columns of $\mathbf{S}$ based on their column norms, i.e., for a column $\mathbf{Y}_j$ of a matrix $\mathbf{Y}$, the column-norm based soft-thresholding function, $\mathcal{C}_\tau(.)$ is defined as

$$\mathcal{C}_\tau(\mathbf{Y}_j) = \max(\mathbf{Y}_j - \tau\mathbf{Y}_j/\|\mathbf{Y}_j\|).$$

C. Parameter Selection

Since the choice of regularization parameters by our main theoretical results contain quantities (such as incoherence etc.) that cannot be evaluated in practice, we employ a grid-search strategy over the range of admissible values for the low-rank and dictionary sparse component to find the best values of the regularization parameters. We now discuss the specifics of the grid-search for each sparsity case.

1) Selecting parameters for the entry-wise case: The choice of parameters ν and λ_e in Algorithm 1 is based on the optimality conditions of the optimization problem shown in (9). As presented in [39], the range of parameters ν and $\nu\lambda_e$ associated with the low-rank part $\mathbf{L}$ and the sparse coefficient matrix $\mathbf{S}$, respectively, lie in $\nu \in \{0, \|\mathbf{M}\|\}$ and $\nu\lambda_e \in \{0, \|\mathbf{D}^\top\mathbf{M}\|_\infty\}$, i.e., for Algorithm 1 $\nu_0 = \|\mathbf{M}\|$.

These ranges for ν and $\nu\lambda_e$ are derived using the optimization problem shown in (9). Specifically, we find the largest values of these regularization parameters which yield a $(\mathbf{0}, \mathbf{0})$ solution for the pair $(\mathbf{L}_0, \mathbf{S}_0)$ by analyzing the optimality conditions of (9). This value of the regularization parameter then defines the upper bound on the range. For instance, let $\lambda_* := \nu$ and $\lambda_1 := \nu\lambda_e$, then the optimality condition is given by

$$\lambda_* \partial_{\mathbf{L}} \|\mathbf{L}\|_* - (\mathbf{M} - \mathbf{L} - \mathbf{DS}) = 0,$$

where the sub-differential set $\partial_{\mathbf{L}}\|\mathbf{L}\|_*$ is defined as

$$\partial_{\mathbf{L}}\|\mathbf{L}\|_*\Big|_{\mathbf{L}=\mathbf{L}_0} = \{\mathbf{U}\mathbf{V}^\top + \mathbf{W} : \|\mathbf{W}\| \leq 1, \mathcal{P}_{\mathcal{L}}(\mathbf{W}) = \mathbf{0}\}.$$

Therefore, for a zero solution pair $(\mathbf{L}_0, \mathbf{S}_0)$ we have that

$$\{\lambda_*\mathbf{W} = \mathbf{M} : \|\mathbf{W}\| \leq 1, \mathcal{P}_{\mathcal{L}}(\mathbf{W}) = \mathbf{0}\},$$

which yields the condition that $\|\mathbf{M}\| \leq \lambda_*$. Therefore, the maximum value of λ_* which drives the low-rank part to an all-zero solution is $\|\mathbf{M}\|$.

Similarly, for the dictionary sparse component the optimality condition for choosing λ_1 is given by

$$\lambda_1 \partial_{\mathbf{S}} \|\mathbf{S}\|_1 - \mathbf{D}^\top(\mathbf{M} - \mathbf{L} - \mathbf{DS}) = 0,$$

where the the sub-differential set $\partial_{\mathbf{S}}\|\mathbf{S}\|_1$ is defined as

$$\partial_{\mathbf{S}}\|\mathbf{S}\|_1\Big|_{\mathbf{S}=\mathbf{S}_0} = \{\text{sign}(\mathbf{S}_0) + \mathbf{F} : \|\mathbf{F}\|_\infty \leq 1, \mathcal{P}_{\mathcal{S}_e}(\mathbf{F}) = \mathbf{0}\}.$$

Again, for a zero solution pair $(\mathbf{L}_0, \mathbf{S}_0)$ we need that

$$\{\lambda_1 \mathbf{F} = \mathbf{D}^\top \mathbf{M} : \|\mathbf{F}\|_\infty \leq 1, \mathcal{P}_{\mathcal{S}_e}(\mathbf{F}) = \mathbf{0}\},$$

which implies that $\|\mathbf{D}^\top \mathbf{M}\|_\infty \leq \lambda_1$. Meaning, that the maximum value of λ_1 that drives the dictionary sparse part to zero is $\|\mathbf{D}^\top \mathbf{M}\|_\infty$.

2) Selecting parameters for the column-wise case: Again, the choice of parameters ν and λ_c is derived from the optimization problem shown in (10). In this case, the range of parameters ν and $\nu\lambda_c$ associated with the low-rank part $\mathbf{L}$ and the sparse coefficient matrix $\mathbf{S}$, respectively, lie in $\nu \in \{0, \|\mathbf{M}\|\}$ and $\nu\lambda_e \in \{0, \|\mathbf{D}^\top \mathbf{M}\|_{\infty,2}\}$, i.e., for Algorithm 1 $\nu_0 = \|\mathbf{M}\|$. The range of regularization parameters are evaluated using the analysis similar to the entry-wise case, by analyzing the optimality conditions for the optimization problem shown in (10), instead of (9).

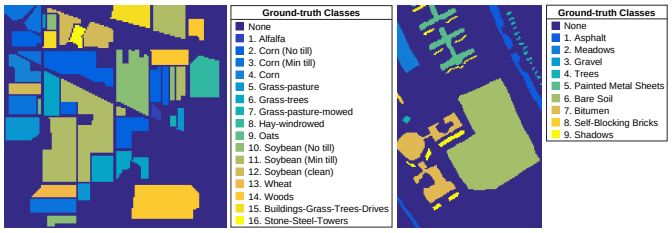
V. Experimental Evaluation

We now evaluate the performance of the proposed technique on real HS data. We begin by introducing the dataset used for the simulations, following which we describe the experimental set-up and present the results.

A. Data

Indian Pines Dataset: We first consider the "Indian Pines" dataset [14], which was collected over the Indian Pines test site in North-western Indiana in the June of 1992 using the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) [47] sensor, a popular choice for collecting HS images for various remote sensing applications. This dataset consists of spectral reflectances across 224 bands in wavelength of ranges $400 - 2500$ nm from a scene which is composed mostly of agricultural land along with two major dual lane highways, a rail line and some built structures, as shown in Fig. 3(a). The dataset is further processed by removing the bands corresponding to those of water absorption, which results in a HS data-cube with dimensions $\{145 \times 145 \times 200\}$ is as visualized in Fig. 1. Here, $n = m = 145$ and $f = 200$. This modified dataset is available as "corrected Indian Pines" dataset [14], with the ground-truth containing 16 classes; Henceforth, referred to as the "Indian Pines Dataset". We form the data matrix $\mathbf{M} \in \mathbb{R}^{f \times nm}$ by stacking each voxel of the image side-by-side, which results in a $\{200 \times 145^2\}$ data matrix $\mathbf{M}$. We will analyze the performance of the proposed technique for the identification of the stone-steel towers (class 16 in the dataset), shown in Fig. 3(a), which constitutes about 93 voxels in the dataset.

Pavia University Dataset: Acquired using Reflective Optics System Imaging Spectrometer (ROSIS) sensor, the Pavia University Dataset [48] consists of spectral reflectances across 103 bands (in the range $430 - 860$ nm) of an urban landscape over northern Italy. The selected subset of the scene, a $\{201 \times 131 \times 103\}$ data-cube, mainly consists of buildings, roads, painted metal sheets and trees, as shown in Fig. 3(b).



(a) Indian Pines (b) Pavia University

Fig. 3: Ground-truth classes in the datasets. Panels (a) and (b) show the ground truth classes for the Indian Pines dataset [14] and Pavia University dataset [48], respectively.

Note that class-3 corresponding to "Gravel" is not present in the selected data-cube considered here. For our demixing task, we will analyze the localization of target class 5, corresponding to the painted metal sheets, which constitutes 707 voxels in the scene. Note that for this dataset $n = 201$, $m = 131$ and $f = 103$.

B. Dictionary

We form the known dictionary $\mathbf{D}$ two ways: 1) where a (thin) dictionary is learned based on the voxels using Algorithm 2, and 2) when the dictionary is formed by randomly sampling voxels from the target class. This is to emulate the ways in which we can arrive at the dictionary corresponding to a target – 1) where the *exact signatures* are not available, and/or there is noise, and 2) where we have access to the exact signatures of the target, respectively. Note that, the optimization procedures for D-RPCA(E) and D-RPCA(C) are agnostic to the selection of the dictionary.

In our experiments for case 1), we learn the dictionary using the target class data $\mathbf{Y} \in \mathbb{R}^{f \times p}$ via Algorithm 2, which (approximately) solves the following optimization problem,

$$\hat{\mathbf{D}} = \underset{\mathbf{D}:\|\mathbf{D}_i\|=1, \mathbf{A}}{\arg.\min} \|\mathbf{Y} - \mathbf{D}\mathbf{A}\|_{\text{F}}^2 + \rho\|\mathbf{A}\|_1,$$

Algorithm 2 Dictionary Learning [8], [9]

Input: Data $\mathbf{Y} \in \mathbb{R}^{f \times p}$, regularization parameter ρ, and the number of dictionary elements d.
Output: The dictionary $\mathbf{D} \in \mathbb{R}^{f \times d}$
Initialize: $\hat{\mathbf{A}} \leftarrow \mathbf{0}_{d \times p}$, $\hat{\mathbf{D}}$ with $\mathcal{N}(0,1)$ entries and columns normalized to have norm 1, $\hat{\mathbf{Y}} = \hat{\mathbf{D}}\hat{\mathbf{A}}$, and tolerance ϵ .
while $\frac{\|\mathbf{Y} - \hat{\mathbf{Y}}\|_{\text{F}}}{\|\mathbf{Y}\|_{\text{F}}} \geq \epsilon$ **do**
  Update Coefficient Matrix $\mathbf{A}$:

$$\hat{\mathbf{A}} = \underset{\mathbf{A}}{\arg.\min} \|\mathbf{Y} - \hat{\mathbf{D}}\mathbf{A}\|_{\text{F}}^2 + \rho\|\mathbf{A}\|_1 \quad (11)$$

  Update Dictionary $\mathbf{D}$:

$$\hat{\mathbf{D}} = \underset{\mathbf{D}:\|\mathbf{D}_i\|=1}{\arg.\min} \|\mathbf{Y} - \mathbf{D}\hat{\mathbf{A}}\|_{\text{F}}^2 \quad (12)$$

  Form Estimate of Data $\hat{\mathbf{Y}}$:
    $\hat{\mathbf{Y}} = \hat{\mathbf{D}}\hat{\mathbf{A}}$
end while
return $\hat{\mathbf{D}}$

TABLE II: Entry-wise sparsity model and Pavia University Dataset. Simulation results are presented for the proposed approach (D-RPCA(E)), robust-PCA based approach on transformed data (RPCA†), matched filtering (MF) on original data $\mathbf{M}$, and matched filtering on transformed data $\mathbf{D}^\dagger\mathbf{M}$ (MF†), across dictionary elements d, and the regularization parameter for initial dictionary learning step ρ. Threshold selects columns with column-norm greater than threshold such that AUC is maximized. For each case, the best performing metrics are reported in bold for readability. Further, " $*$ " denotes the case where ROC curve was "flipped" (i.e. classifier output was inverted to achieve the best performance).

(a) Learned dictionary, $d = 30$

d	ρ	Method	Threshold	Performance at best operating point		AUC
				TPR	FPR	
30	0.01	**D-RPCA(E)**	0.150	**0.989**	**0.015**	**0.992**
		RPCA†	0.700	0.849	0.146	0.925
		MF	N/A	0.929	0.073	0.962
		MF†	N/A	0.502	0.498	0.498
	0.1	**D-RPCA(E)**	0.050	**0.982**	**0.019**	**0.992**
		RPCA†	3.000	0.638	0.374	0.664
		MF	N/A	0.979	0.053	0.986
		MF†	N/A	0.620	0.381	0.660
	0.5	**D-RPCA(E)**	0.080	**0.982**	**0.019**	**0.992**
		RPCA†	2.500	0.635	0.381	0.671
		MF	N/A	0.980	0.159	0.993
		MF$^\dagger_*$	N/A	0.555	0.447	0.442

(b) Dictionary by sampling voxels, $d = 60$

d	Method	Threshold	Performance at best operating point		AUC
			TPR	FPR	
60	**D-RPCA(E)**	0.060	**0.986**	0.016	**0.995**
	RPCA†	1.000	0.799	0.279	0.793
	MF	N/A	0.980	**0.011**	0.994
	MF†	N/A	0.644	0.355	0.700

(c) Average performance

Method	TPR		FPR		AUC	
	Mean	St.Dev.	Mean	St.Dev.	Mean	St.Dev.
D-RPCA(E)	**0.984**	**0.003**	**0.014**	**0.002**	**0.993**	**0.001**
RPCA†	0.730	0.110	0.295	0.110	0.763	0.123
MF	0.967	0.025	0.074	0.062	0.983	0.0149
MF†	0.580	0.064	0.420	0.065	0.575	0.125

Algorithm 2 operates by alternating between updating the sparse coefficients (11) via FISTA [43] and dictionary (12) via the Newton method [49].

For case 2), the columns of the dictionary are set as the known data voxels of the target class. Specifically, instead of learning a dictionary based on a target class of interest, we set it as the exact signatures observed previously. Note that for this case, the dictionary is not normalized at this stage since the specific normalization depends on the particular demixing problem of interest, discussed shortly. In practice, we can store the un-normalized dictionary $\mathbf{D}$ (formed from the voxels), consisting of actual *signatures* of the target material, and can normalize it after the HS image has been acquired.

C. Experimental Setup

Normalization of data and the dictionary: For normalizing the data, we divide each element of the data matrix $\mathbf{M}$ by $\|\mathbf{M}\|_\infty$ to preserve the inter-voxel scaling. For the dictionary, in the learned dictionary case, i.e., case 1), the dictionary already has unit-norm columns as a result of Algorithm 2. Further, when the dictionary is formed from the data directly, i.e., for case 2), we divide each element of $\mathbf{D}$ by $\|\mathbf{M}\|_\infty$, and then normalize the columns of $\mathbf{D}$, such that they are unit-norm.

Dictionary selection for the Indian Pines Dataset: For the learned dictionary case, we evaluate the performance of the aforementioned techniques for both entry-wise and column-wise settings for two dictionary sizes, $d = 4$ and $d = 10$, for three values of the regularization parameter ρ, used for the initial dictionary learning step, i.e., $\rho = 0.01,\ 0.1$ and 0.5. Here, the parameter ρ controls the sparsity during the initial dictionary learning step; see Algorithm 2. For the case when dictionary is selected from the voxels directly, we randomly select 15 voxels from the target class-16 to form our dictionary.

Dictionary selection for the Pavia University Dataset: Here, for the learned dictionary case, we evaluate the performance of the aforementioned techniques for both entry-wise and column-wise settings for a dictionary of size $d = 30$ for three values of the regularization parameter ρ, used for the initial dictionary learning step, i.e., $\rho = 0.01,\ 0.1$ and 0.5. Further, we randomly select 60 voxels from the target class-5, when the dictionary is formed from the data voxels.

Comparison with matched filtering (MF)-based approaches: In addition to the robust PCA-based and OP-based techniques introduced in Section I-D, we also compare the performance of our techniques with two MF-based approaches. These MF-based techniques are agnostic to our model assumptions, i.e., entry-wise or column-wise sparsity cases. Therefore, the following description of these techniques applies to both sparsity cases.

For the first MF-based technique, referred to as MF, we form the inner-product of the column-normalized data matrix $\mathbf{M}$, denoted as $\mathbf{M}_n$, with the dictionary $\mathbf{D}$, i.e., $\mathbf{D}^\top\mathbf{M}_n$, and select the maximum absolute inner-product per column. For the second MF-based technique, MF†, we perform matched filtering on the pseudo-inversed data $\widetilde{\mathbf{M}} = \mathbf{D}^\dagger\mathbf{M}$. Here, the matched filtering corresponds to finding maximum absolute entry for each column of the column-normalized $\widetilde{\mathbf{M}}$. Next, in both cases we scan through 1000 threshold values between $(0, 1]$ to generate the results.

Performance Metrics: We evaluate the performance of these techniques via the receiver operating characteristic (ROC) plots. ROC plots are a staple for analysis of classification performance of a binary classifier in machine learning; see [50] for details. Specifically, it is a plot between the true positive rate (TPR) and the false positive rate (FPR), where a higher TPR (close to 1) and a lower FPR (close to 0) indicate that the classifiier performs detects all the elements in the class while rejecting those outside the class.

A natural metric to gauge good performance is the area under the curve (AUC) metric. It indicates the area under the ROC curve, which is maximized when TPR $= 1$ and FPR $= 0$, therefore, a higher AUC is preferred. Here, an AUC of 0.5 indicates that the performance of the classifier is roughly as

TABLE III: Column-wise sparsity model and Indian Pines Dataset. Simulation results are presented for the proposed approach (D-RPCA(C)), Outlier Pursuit (OP) based approach on transformed data (OP†), matched filtering (MF) on original data $\mathbf{M}$, and matched filtering on transformed data $\mathbf{D}^\dagger\mathbf{M}$ (MF†), across dictionary elements d, and the regularization parameter for initial dictionary learning step ρ. Threshold selects columns with column-norm greater than threshold such that AUC is maximized. For each case, the best performing metrics are reported in bold for readability. Further, " $*$ " denotes the case where ROC curve was "flipped" (i.e. classifier output was inverted to achieve the best performance).

(a) Learned dictionary, $d = 4$

d	ρ	Method	Threshold	Performance at best operating point		AUC
				TPR	FPR	
4	0.01	**D-RPCA(C)**	0.905	**0.989**	**0.014**	**0.998**
		OP†	0.895	0.989	0.015	0.998
		MF$_*$	N/A	0.656	0.376	0.611
		MF$_*^\dagger$	N/A	0.624	0.373	0.639
	0.1	**D-RPCA(C)**	0.805	**0.989**	**0.013**	**0.998**
		OP$_*^\dagger$	1.100	0.720	0.349	0.682
		MF$_*$	N/A	0.742	0.256	0.780
		MF†	N/A	0.828	0.173	0.905
	0.5	**D-RPCA(C)**	1.800	**0.989**	**0.010**	**0.998**
		OP†	1.300	0.989	0.012	0.998
		MF	N/A	0.548	0.474	0.556
		MF$_*^\dagger$	N/A	0.849	0.146	0.939

(b) Learned dictionary, $d = 10$

d	ρ	Method	Threshold	Performance at best operating point		AUC
				TPR	FPR	
10	0.01	**D-RPCA(C)**	0.800	**0.946**	**0.016**	**0.993**
		OP†	1.300	0.946	0.060	0.988
		MF$_*$	N/A	0.946	0.060	0.987
		MF$_*^\dagger$	N/A	0.527	0.468	0.511
	0.1	**D-RPCA(C)**	0.550	**0.979**	**0.029**	**0.997**
		OP†	0.800	0.893	0.112	0.928
		MF$_*$	N/A	0.688	0.302	0.714
		MF†	N/A	0.527	0.470	0.523
	0.5	**D-RPCA(C)**	1.400	**0.989**	**0.037**	**0.997**
		OP†	0.800	0.807	0.148	0.847
		MF	N/A	0.807	0.309	0.781
		MF$_*^\dagger$	N/A	0.527	0.468	0.539

(c) Dictionary by sampling voxels, $d = 15$

d	Method	Threshold	Performance at best operating point		AUC
			TPR	FPR	
15	**D-RPCA(C)**	0.800	0.989	0.018	0.998
	OP†	2.200	0.882	0.126	0.900
	MF	N/A	0.957	0.085	0.978
	MF†	N/A	0.796	0.217	0.857

(d) Average performance

Method	TPR		FPR		AUC	
	Mean	St.Dev.	Mean	St.Dev.	Mean	St.Dev.
D-RPCA(C)	**0.981**	**0.016**	**0.020**	**0.010**	**0.997**	**0.002**
OP†	0.889	0.099	0.117	0.115	0.906	0.114
MF	0.763	0.151	0.266	0.149	0.772	0.166
MF†	0.668	0.151	0.331	0.148	0.702	0.192

good as a coin flip. As a result, if a classifier has an AUC < 0.5, one can improve the performance by simply inverting the result of the classifier. This effectively means that AUC is evaluated after "flipping" the ROC curve. In other words, this means that the classifier is good at rejecting the class of interest, and taking the complement of the classifier decision can be used to identify the class of interest.

In our experiments, MF-based techniques often exhibit this phenomenon. Specifically, when the dictionary contains element(s) which resemble the average behavior of the spectral signatures, the inner-product between the normalized data columns and these dictionary elements may be higher as compared to other distinguishing dictionary elements. Since, MF-based techniques rely on the maximum inner-product between the normalized data columns and the dictionary, and further since the spectral signatures of even distinct classes are highly correlated; see, for instance Fig. 2, where MF-based approaches in these cases can effectively reject the class of interest. This leads to an AUC < 0.5. Therefore, as discussed above, we invert the result of the classifier (indicated as $(\cdot)_*$ in the tables) to report the best performance. If using MF-based techniques, this issue can potentially be resolved in practice by removing the dictionary elements which tend to resemble the average behavior of the spectral signatures.

D. Parameter Setup for the Algorithms

Entry-wise sparsity case: We evaluate and compare the performance of the proposed method D-RPCA(E) with RPCA† (described in Section I-C), MF, and MF†. Specifically, we evaluate the performance of these techniques via the receiver operating characteristic (ROC) plot for the Indian Pines dataset and the Pavia University dataset, with the results shown in Table I(a)-(d) and Table II(a)-(c), respectively.

For the proposed technique, we employ the accelerated proximal gradient (APG) algorithm shown in Algorithm 1 and discussed in Section IV to solve the optimization problem shown in D-RPCA(E). Similarly, for RPCA† we employ the APG algorithm with transformed data matrix $\widetilde{\mathbf{M}}$, while setting $\mathbf{D} = \mathbf{I}$.

With reference to selection of tuning parameters for the APG solver for (D-RPCA(E)) (RPCA†, respectively), we choose $\upsilon = 0.95$, $\nu = \|\mathbf{M}\|$ ($\nu = \|\widetilde{\mathbf{M}}\|$), $\bar{\nu} = 10^{-4}$, and scan through 100 values of λ_e in the range $\lambda_e \in (0, \|\mathbf{D}^\top\mathbf{M}\|_\infty/\|\mathbf{M}\|]$ ($\lambda_e \in (0, \|\widetilde{\mathbf{M}}\|_\infty/\|\widetilde{\mathbf{M}}\|]$), to generate the ROCs. We threshold the resulting estimate of the sparse part $\mathbf{S} \in \mathbb{R}^{d\times nm}$ based on its column norm. We choose the threshold such that the AUC metric is maximized for both cases (D-RPCA(E) and RPCA†).

Column-wise sparsity case: For this case, we evaluate and compare the performance of the proposed method D-RPCA(C) with OP† (as described in Section I-C), MF, and MF†. The results for the Indian Pines dataset and the Pavia University dataset as shown in Table III(a)-(d) and Table IV(a)-(c), respectively.

As in the entry-wise sparsity case, we employ the accelerated proximal gradient (APG) algorithm presented in Algorithm 1 to solve the optimization problem shown in D-RPCA(C).

TABLE IV: Column-wise sparsity model and Pavia University Dataset. Simulation results for the proposed approach (D-RPCA(C)), Outlier Pursuit (OP) based approach (OP†), matched filtering (MF) on original data $\mathbf{M}$, and matched filtering on transformed data $\mathbf{D}^\dagger\mathbf{M}$ (MF†), across dictionary elements d, and the regularization parameter for initial dictionary learning step ρ. Threshold selects columns with column-norm greater than threshold such that AUC is maximized. For each case, the best performing metrics are reported in bold for readability. Further, "$*$" denotes the case where ROC curve was "flipped" (i.e. classifier output was inverted to achieve the best performance).

(a) Learned dictionary, $d = 30$

d	ρ	Method	Threshold	Performance at best operating point		AUC
				TPR	FPR	
30	0.01	**D-RPCA(C)**	0.065	**0.990**	**0.015**	**0.991**
		OP†	0.800	0.7581	0.3473	0.705
		MF	N/A	0.929	0.073	0.962
		MF†	N/A	0.502	0.50	0.498
	0.1	**D-RPCA(C)**	0.070	**0.996**	**0.022**	**0.994**
		OP†	0.100	0.989	0.3312	0.904
		MF	N/A	0.979	0.053	0.986
		MF†	N/A	0.62	0.3814	0.66
	0.5	**D-RPCA(C)**	0.035	**0.983**	**0.017**	**0.995**
		OP†	0.200	0.940	0.264	0.887
		MF	N/A	0.980	0.160	0.993
		MF$^\dagger_*$	N/A	0.555	0.447	0.442

(b) Dictionary by sampling voxels, $d = 60$

d	Method	Threshold	Performance at best operating point		AUC
			TPR	FPR	
60	**D-RPCA(C)**	0.020	**0.993**	0.022	**0.994**
	OP†	0.250	0.963	0.264	0.907
	MF	N/A	0.980	**0.011**	0.994
	MF†	N/A	0.644	0.355	0.700

(c) Average performance

Method	TPR		FPR		AUC	
	Mean	St.Dev.	Mean	St.Dev.	Mean	St.Dev.
D-RPCA(C)	**0.990**	**0.006**	**0.015**	**0.003**	**0.993**	**0.002**
OP†	0.912	0.105	0.302	0.044	0.850	0.098
MF	0.97	0.025	0.074	0.063	0.984	0.015
MF†	0.580	0.064	0.4208	0.065	0.575	0.124

Similarly, for OP† we employ the APG with transformed data matrix $\widetilde{\mathbf{M}}$, while setting $\mathbf{D} = \mathbf{I}$. For the tuning parameters for the APG solver for (D-RPCA(C)) (OP†, respectively), we choose $\upsilon = 0.95$, $\nu = \|\mathbf{M}\|$ ($\nu = \|\widetilde{\mathbf{M}}\|$), $\bar{\nu} = 10^{-4}$, and scan through 100 λ_cs in the range $\lambda_c \in (0, \|\mathbf{D}^\top\mathbf{M}\|_{\infty,2}/\|\mathbf{M}\|]$ ($\lambda_c \in (0, \|\widetilde{\mathbf{M}}\|_{\infty,2}/\|\widetilde{\mathbf{M}}\|]$), to generate the ROCs. As in the previous case, we threshold the resulting estimate of the sparse part $\mathbf{S} \in \mathbb{R}^{d\times nm}$ based on its column norm.

E. Analysis

Table I–II and Table III–IV show the ROC characteristics and the classification performance of the proposed techniques D-RPCA(E) and D-RPCA(C), for two datasets under consideration, respectively, under various choices of the dictionary $\mathbf{D}$ and regularization parameter ρ for Algorithm 2. We note that both proposed techniques D-RPCA(E) and D-RPCA(C) on an average outperform competing techniques, emerging as the most reliable techniques across different dictionary choices for the demixing task at hand; see Tables I(d), II(c), III(d), and IV(c).

Further, the performance of D-RPCA(C) is slightly better than D-RPCA(E). This can be attributed to the fact that the column-wise sparsity model does not require the columns of $\mathbf{S}$ to be sparse themselves. As alluded to in Section I-B, this allows for higher flexibility in the choice of the dictionary elements for the thin dictionary case.

In addition, we see that the matched filtering-based techniques (and even OP† based technique for $d = 4$ and $\rho = 0.1$ in Table III) exhibit "flip" or inversion of the ROC curve. As described in Section V-C, this phenomenon is an indicator that a classifier is better at rejecting the target class. In case of MF-based technique, this is a result of a dictionary that contains an element that resembles the average behavior of the spectral responses. A similar phenomenon is at play in case of the OP† for $d = 4$ and $\rho = 0.1$ in Table III. Specifically, here the inversion indicates that the dictionary is capable of representing the columns of the data $\mathbf{M}$ effectively, which leads to an increase in the corresponding column norms in their representation $\widehat{\mathbf{M}}$. Coupled with the fact that the component $\mathbf{L}$ is no longer low-rank for this thin dictionary case (see our discussion in Section I-D), this results in rejection of the target class. On the other hand, our techniques D-RPCA(E) and D-RPCA(C) do not suffer from this issue. Moreover, note that across all the experiments, the thresholds for RPCA† and OP† are higher than their D-RPCA counterparts. This can also be attributed to the pre-multiplication by the pseudo-inverse of the dictionary $\mathbf{D}^\dagger$, which increases column norms based on the leading singular values of $\mathbf{D}$. Therefore, using D-RPCA(E), when the target spectral response admits a sparse representation, and D-RPCA(C), otherwise, yield consistent and superior results as compared to related techniques considered in this work.

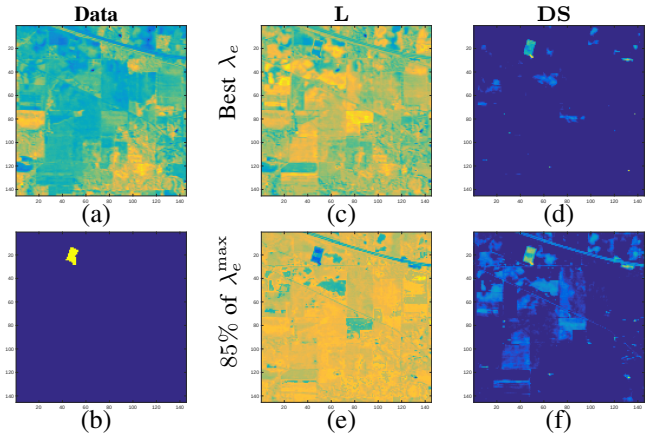


Fig. 4: Recovery of the low-rank component $\mathbf{L}$ and the dictionary sparse component $\mathbf{DS}$ for different values of λ for the proposed technique at $f = 50$-th channel of the [14] (shown in panel (a)) corresponding to the results shown in Table I(c). Panel (b) corresponds to the ground truth for class-16. Panel (c) and (d) show the recovery of the low-rank part and dictionary sparse part for a λ at the best operating point. While, panels (e) and (f) show the recovery of these components at $\lambda_e = 85\%$ of $\lambda_e^{\max}$. Here, $\lambda_e^{\max}$ denotes the maximum value λ_e can take; see Section IV-C1.

There are other interesting recovery results which warrant our attention. Fig. 4 shows the low-rank and the dictionary sparse component recovered by D-RPCA(E) for two different values of λ_e, for the case where we form the dictionary by randomly sampling the voxels (Table I(c)) for the Indian Pines Dataset [14]. Interestingly, we recover the rail tracks/roads running diagonally on the top-right corner, along with some low-density housing; see Fig 4 (f). This is because the *signatures* we seek (stone-steel towers) are similar to the signatures of the materials used in these structures. This further corroborates the applicability of the proposed approach in detecting the presence of a particular spectral *signature* in a HS image. However, this also highlights potential drawback of this technique. As D-RPCA(E) and D-RPCA(C) are based on identifying materials with similar composition, it may not be effective in distinguishing between very closely related classes, say two agricultural crops, also indicated by our theoretical results.

VI. Conclusions

We present a generalized robust PCA-based technique to localize a target in a HS image, based on the *a priori* known spectral *signature* of the material we wish to localize. We model the data as being composed of a low-rank component and a dictionary-sparse component, and consider two different sparsity patterns corresponding to different structural assumptions on the data, where the dictionary contains the *a priori* known spectral *signatures* of the target. We adapt the theoretical results of our previous work [10]–[12], to present the conditions under which such decompositions recover the two components for the HS demixing task. Further, we evaluate and compare the performance of the proposed method via experimental evaluations for a classification task for different choices of the dictionary on real HS image datasets, and demostrate the applicability of the proposed techniques for a target localization in HS images.

References

[1] M. Borengasser, W. S. Hungate, and R. Watkins, *Hyperspectral remote sensing: principles and applications*, CRC press, 2007.

[2] B. Park and R. Lu, *Hyperspectral imaging technology in food and agriculture*, Springer, 2015.

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

[4] J. B. Greer, "Sparse demixing of hyperspectral images," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 219–228, Jan 2012.

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

[6] B. A. Olshausen and D. J. Field, "Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1?," *Vision Research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[7] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: Design of Dictionaries for Sparse Representation," *In Proceedings of SPARS*, pp. 9–12, 2005.

[8] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online Learning for Matrix Factorization and Sparse Coding," *Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[9] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in Neural Information Processing Systems*, 2007, pp. 801–808.

[10] S. Rambhatla, X. Li, and J. Haupt, "A dictionary based generalization of robust PCA," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2016.

[11] X. Li, J. Ren, S. Rambhatla, Y. Xu, and J. Haupt, "Robust pca via dictionary based outlier pursuit," in *2018 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP),*. IEEE, 2018.

[12] S. Rambhatla, X. Li, J. Ren, and J. Haupt, "A Dictionary-Based Generalization of Robust PCA Part I: Study of Theoretical Properties," *Journal In Review*, 2018.

[13] S. Rambhatla, X. Li, and J. Haupt, "Target-based hyperspectral demixing via generalized robust PCA," in *51st Asilomar Conference on Signals, Systems, and Computers, ACSSC 2017, Pacific Grove, CA, USA, October 29 - November 1, 2017*, 2017, pp. 420–424.

[14] M. F. Baumgardner, L. L. Biehl, and D. A. Landgrebe, "220 Band AVIRIS Hyperspectral Image Data Set: June 12, 1992 Indian Pine Test Site 3, dataset available via http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes," Sept 2015.

[15] K. Pearson, "On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[16] I. Jolliffe, *Principal component analysis*, Wiley Online Library, 2002.

[17] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 11, 2011.

[18] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM Journal on Optimization*, vol. 21, no. 2, pp. 572–596, 2011.

[19] H. Xu, C. Caramanis, and S. Sanghavi, "Robust pca via outlier pursuit," in *Advances in Neural Information Processing Systems*, 2010, pp. 2496–2504.

[20] Z. Zhou, X. Li, J. Wright, E. J. Candès, and Y. Ma, "Stable principal component pursuit," in *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*. IEEE, 2010, pp. 1518–1522.

[21] X. Ding, L. He, and L. Carin, "Bayesian robust principal component analysis," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3419–3430, 2011.

[22] J. Wright, A. Ganesh, K. Min, and Y. Ma, "Compressive principal component pursuit," *Information and Inference*, vol. 2, no. 1, pp. 32–68, 2013.

[23] Y. Chen, A. Jalali, S. Sanghavi, and C. Caramanis, "Low-rank matrix recovery from errors and erasures," *IEEE Transactions on Information Theory*, vol. 59, no. 7, pp. 4324–4337, 2013.

[24] X. Li and J. Haupt, "Identifying outliers in large matrices via randomized adaptive compressive sampling," *Trans. Signal Processing*, vol. 63, no. 7, pp. 1792–1807, 2015.

[25] X. Li and J. Haupt, "Locating salient group-structured image features via adaptive compressive sensing," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2015.

[26] X. Li and J. Haupt, "Outlier identification via randomized adaptive compressive sampling," in *IEEE International Conference on Acoustic, Speech and Signal Processing*, 2015.

[27] X. Li and J. Haupt, "A refined analysis for the sample complexity of adaptive compressive outlier sensing," in *IEEE Workshop on Statistical Signal Processing*, 2016.

[28] X. Li, J. Ren, Y. Xu, and J. Haupt, "An efficient dictionary based robust pca via sketching," *Technical Report*, 2016.

[29] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227–234, 1995.

[30] D. L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2845–2862, 2001.

[31] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[32] S. Rambhatla and J. Haupt, "Semi-blind source separation via sparse representations and online dictionary learning," in *Signals, Systems and Computers, 2013 Asilomar Conference on*. IEEE, 2013, pp. 1687–1691.

[33] Y. Moudden, J. Bobin, J. L. Starck, and J. M. Fadili, "Dictionary learning with spatio-spectral sparsity constraints," in *Signal Processing with Adaptive Sparse Structured Representations(SPARS)*, 2009.

[34] J. Bobin, Y. Moudden, J. L. Starck, and J. Fadili, "Sparsity constraints for hyperspectral data analysis: Linear mixture model and beyond," 2009.

[35] R. Kawakami, Y. Matsushita, J. Wright, M. Ben-Ezra, Y. W. Tai, and K. Ikeuchi, "High-resolution hyperspectral imaging via matrix factorization," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2011, pp. 2329–2336.

[36] A. S. Charles, B. A. Olshausen, and C. J. Rozell, "Learning sparse codes for hyperspectral imagery," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 963–978, Sept 2011.

[37] P. V. Giampouras, K. E. Themelis, A. A. Rontogiannis, and K. D. Koutroumbas, "Simultaneously sparse and low-rank abundance matrix estimation for hyperspectral image unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4775–4789, Aug 2016.

[38] M. Golbabaee, S. Arberet, and P. Vandergheynst, "Distributed compressed sensing of hyperspectral images via blind source separation," in *Forty Fourth Asilomar Conference on Signals, Systems and Computers*, Nov 2010, pp. 196–198.

[39] M. Mardani, G. Mateos, and G. B. Giannakis, "Recovery of low-rank plus compressed sparse matrices with application to unveiling traffic anomalies," *IEEE Transactions on Information Theory*, vol. 59, no. 8, pp. 5186–5205, 2013.

[40] R. J. Duffin and A. C. Schaeffer, "A class of nonharmonic fourier series," *Transactions of the American Mathematical Society*, vol. 72, no. 2, pp. 341–366, 1952.

[41] C. Heil, "What is ... a frame?," *Notices of the American Mathematical Society*, vol. 60, no. 6, June/July 2013.

[42] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $\mathcal{O}(1/k^2)$," *Soviet Mathematics Doklady 27*, pp. 72–376, 1983.

[43] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[44] K.C.Toh and S. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems," *Pacific Journal of optimization*, vol. 6, no. 615-640, pp. 15, 2010.

[45] M. Chen, A. Ganesh, Z. Lin, Y. Ma, J. Wright, and L. Wu, "Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix," *Coordinated Science Laboratory Report no. UILU-ENG-09-2214*, 2009.

[46] H. Karimi, J. Nutini, and M. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the Polyak-łojasiewicz condition," in *Machine Learning and Knowledge Discovery in Databases*, Cham, 2016, pp. 795–811, Springer International Publishing.

[47] Jet Propulsion Laboratory, NASA and California Institute of Technology, "Airborne Visible/Infrared Imaging Spectrometer," 1987, Available at http://aviris.jpl.nasa.gov/.

[48] P. Gamba, "Pavia centre and university, dataset available via http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes#Pavia_Centre_and_University," .

[49] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, NY, USA, second edition, 2006.

[50] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, vol. 112, Springer, 2013.