Multi-task Sparse Nonnegative Matrix Factorization for Joint Spectral-Spatial Hyperspectral Imagery Denoising

Minchao Ye, Yuntao Qian, *Member, IEEE* and Jun Zhou, *Senior Member, IEEE*

Abstract

Hyperspectral imagery (HSI) denoising is a challenging problem because of the difficulty in preserving both spectral and spatial structures simultaneously. In recent years, sparse coding, among many methods dedicated to the problem, has attracted much attention and showed state-of-the-art performance. Due to the low-rank property of natural images, an assumption can be made that the latent clean signal is a linear combination of a minority of basis atoms in a dictionary, while noise component is not. Based on this assumption, denoising can be explored as a sparse signal recovery task with the support of a dictionary. In this paper, we propose to solve the HSI denoising problem by sparse nonnegative matrix factorization (SNMF) which is an integrated model that combines parts-based dictionary learning and sparse coding. The noisy image is used as the training data to learn a dictionary, and sparse coding is used to recover the image based on this dictionary. Unlike most HSI denoising approaches which treat each band image separately, we take the joint spectral-spatial structure of HSI into account. Inspired by multi-task learning, a multi-task SNMF (MTSNMF) method is developed in which band-wise denoising is linked across the spectral domain by sharing a common coefficient matrix. The intrinsic image structures are treated differently but inter-dependently within the spatial and spectral domains, which allows the physical properties of image in both spatial and spectral domains to be reflected in the denoising model. In addition, we introduce variance stabilizing transformation (VST) to provide a denoising solution for HSI which has both signal-dependent and signal-independent noise components. The experimental results show that MTSNMF has superior performance on both synthetic and real-world data compared with several other denoising methods.

Index Terms

Hyperspectral imagery, noise reduction, multi-task learning, nonnegative matrix factorization, sparse coding.

M. Ye and Y. Qian are with the Institute of Artificial Intelligence, College of Computer Science, Zhejiang University, Hangzhou 310027, P.R. China. Correspondence author: Y. Qian (ytqian@zju.edu.cn)

J. Zhou is with School of Information and Communication Technology, Griffith University, Nathan, Australia.

This work was supported by the National Basic Research Program of China (No. 2012CB316400), the National Natural Science Foundation of China (No. 61171151), and Australian Research Councils DECRA Projects funding scheme (project ID DE120102948)

I. Introduction

Hyperspectral imagery (HSI) contains hundreds of spectral entries per pixel, and is becoming an increasingly valuable source of information for scene understanding. The spaceborne, airborne, or ground-based hyperspectral imaging sensors unavoidably introduce noises into the acquired HSI data during the imaging process, so noise reduction is a necessary pre-processing step for many HSI applications including object classification, target detection and spectral unmixing. Some methods have been proposed to tackle this problem, for example, wavelet transform [1], [2], curvelet transform [3], linear/nonlinear/bilateral filtering [4]–[7], total variation [8], sparse representation [9]–[11], and tensor factorization [12]–[14].

Sparse representation has recently attracted much attention, and has demonstrated state-of-the-art performance of image denoising [15]–[17]. These approaches assume that the clean signals lie in a low-dimensional subspace spanned by only a few atoms in a dictionary. This suggests that clean signals can be represented by the linear combination of a few atoms. On the contrary, noise components do not have this property due to their randomness. Therefore, the noise components can be greatly reduced when the noisy signals are projected into a low-dimensional subspace spanned by selected active atoms.

The earliest application of sparse representation in HSI denoising can be traced back to fixed dictionary based methods. Wavelet shrinkage method uses wavelet dictionary and performs shrinkage operation over the coefficients by hard or soft thresholding to reach a sparse representation [1], [2]. The selection of dictionary is an important topic in the sparse representation [10], [11], [18]–[20]. Several orthogonal dictionaries have been proposed for image or HSI denoising, such as discrete cosine transform (DCT) and discrete wavelet transform (DWT) dictionaries [1]. The orthogonal dictionaries are insufficient to provide geometric invariant properties [21]. Consequently, overcomplete dictionaries are introduced. Examples include an undecimated wavelet transform (UWT) dictionary [19], and an overcomplete dictionary composed of DCT and DWT [10].

Fixed dictionary is independent of the task being done, i.e., it cannot be adaptively adjusted based on the latent clean signal and noise. Therefore, data-driven dictionary is introduced into sparse representation [22]. Dictionary learning algorithms range from MOD (Method of Optimal Directions) [23], GPCA (Generalized principal component analysis) [24], to K-means and K-SVD [25]. They have been applied to various areas, such as signal denoising, image super-resolution and pattern classification [15], [17], [26], [27]. In general, the training samples for dictionary learning can come from two sources: noise-free images and corrupted images [15]. For example, in [28], training samples were drawn from an auxiliary clean panchromatic image captured in the same scene as the HSI, while in [29], the training samples were selected from the noisy HSI itself. Obtaining noise-free images is not easy because they should be similar to the noisy target images. On the contrary, directly using the noisy images as the training samples is a more convenient option, and more importantly, the learned dictionary is adapted to the denoising task. The problem of this option, however, is that noisy components are inevitably more or less introduced into the dictionary. In these works, data-driven dictionaries have demonstrated some advantages over fixed dictionaries.

An alternative dictionary learning method is based on nonnegative matrix factorization (NMF), which is a parts-based model generating the atoms in a dictionary that contains various parts of the training samples. Such solution naturally favors sparse and parts-based representations [30], [31]. Matrix decomposition without additional constraint is highly ill-posed, that is, there typically exist many different but equivalent factorizations. Compared with other constraints, nonnegativity has five advantages: 1) Many real-world data such as HSI are nonnegative and the corresponding hidden components have a physical meaning only when nonnegative. 2) The space of solution is significantly reduced and the stability of solution is improved, which tackles the ill-posed problem. 3) Parts-based dictionary can well represent local and fine structures of image [30], [32]. 4) As nonnegativity always yields sparsity in matrix factorization, there is the consistency between sparsity and nonnegativity. 5) Nonnegativity has been used in matrix factorization and tensor factorization for image denoising, and has been proved to deliver favorable performance, especially the deformation problem of fine structure in true signal can be avoided after noise reduction [33], [34]. In this paper, we use sparse NMF model (SNMF) [35] to combine dictionary learning and sparse coding into a single solution, which updates the basis atoms in the dictionary and the corresponding coefficients for the sparse coding in an alternating manner. The constraint of sparsity is imposed on the coefficients to achieve a sparse linear combination of atoms. The band image is firstly divided into overlapping patches which are merged into a matrix. Then this matrix is approximated by the multiplication of a dictionary matrix and a corresponding sparse coefficient matrix via SNMF. Finally this approximate matrix is mapped into the recovered band image. SNMF is a simple yet effective image denoising approach, which not only has the advantage of data-driven dictionary in sparse representation, but also enables joint optimization of dictionary learning and sparse coding.

The problem of pixel-by-pixel or band-by-band denoising is that the joint spectral-spatial correlation is neglected. In order to make use of this intrinsic structure of HSI to improve the denoising performance, several three-dimensional (3D) denoising methods have been proposed for HSI, including 3D Tucker decomposition [13], 3D wavelets [19], 3D nonlocal means [7], 3D sparse coding [36], etc. However, these 3D methods treat the spectral and spatial domains in the same way, i.e., they simply consider an HSI as a general 3D cube with indistinguishable dimensions, ignoring the fact that spectral correlation and spatial correlation are caused by the different physical mechanism [37]. Inspired by the sparse representation and the joint spectral-spatial structure, we further develop a multi-task SNMF (MTSNMF) model for HSI denoising. Multi-task learning is a type of machine learning that learns a task together with other related tasks at the same time, using a shared representation, in which what is learned for each task can help other tasks to be learned better [38]. In our model, a single band image denoising is seen as one task, and the denoising tasks on all band images are linked by sharing a common coefficient matrix. MTSNMF takes the joint spectral-spatial information into account. The intrinsic structures within the spatial domain, within the spectral domain, and cross the spectral-spatial domain are treated differently and inter-dependently. Intuitively, the spatial information is embodied in the learned dictionaries corresponding to each individual band image, the spectral information is reflected in the shared sparse codes for the patches at the same position, and the spectral-spatial information is embedded in the joint optimization of dictionary learning and sparse coding.

Most denoising approaches including MTSNMF are built with assumption of signal-independent Gaussian noise, although they also can be used for other noise types at the cost of denoising performance. Research has suggested that both signal-dependent and signal-independent noise sources exist in hyperspectral imaging sensors [39], in which signal-independent noise is modeled by Gaussian distribution, and signal-dependent component by Poisson distribution. To make the proposed method be able to handle mixed Poisson-Gaussian noise, we extend the MTSNMF method via variance stabilizing transformation (VST) [10], [40]. VST is a non-linear transformation method that converts the mixed Poisson-Gaussian noise into signal-independent Gaussian noises, which allows Gaussian based denoising algorithms to be applicable.

The main contributions of this paper can be summarized as follows. 1) SNMF integrates dictionary learning and sparse coding into one model, which extracts training samples from the noisy image itself instead of other clean image sources. 2) MTSNMF provides a new method to embed spectral and spatial information into sparse representation model, in which spatial and spectral statistical correlations are utilized to optimize dictionary and sparse codes. 3) VST method is used to extend MTSNMF to tackle mixed Poisson-Gaussian noise. 4) Compared with state-of-the-art methods, the proposed approach demonstrates superior denoising performance on both synthetic and real HSI data sets.

The remainder of this paper is organized as follows. Section II introduces SNMF based dictionary leaning and sparse coding method for single band image denoising. Section III proposes the MTSNMF model and the spectral-spatial noise reduction scheme based on MTSNMF. The VST based extension of MTSNMF which can handle mixed Poisson-Gaussian is also discussed in this section. The multiplicative update algorithm and hierarchical alternating least squares algorithm for MTSNMF are derived in Section IV along with their accelerated versions. Experimental results on both synthetic and real-world data are described and analyzed in Section V. Finally, conclusions are drawn in Section VI.

II. SNMF BASED BAND IMAGE DENOISING

Before introducing the SNMF and MTSNMF models, the related major notations are listed in Table I. An HSI consists of numerous band images. In this section, we focus on denoising single band image, i.e., SNMF based noise reduction for a 2D image. In image sparse representation framework, a 2D image is split into patches of size $\sqrt{N} \times \sqrt{N}$ by sliding window, so that the sparse representation of image is built on the sparse representation of these patches. If a dictionary is given in advance, sparse coding of a noisy patch is modeled as ℓ_1 norm regularized sparse regression, which is also called basis pursuit denoising (BPDN) [41].

$$\min_{\mathbf{s}} \frac{1}{2}\|\mathbf{x} - \mathbf{A}\mathbf{s}\|_2^2 + \lambda\|\mathbf{s}\|_1 \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^{N\times 1}$ is a vector representing an observed noisy image patch, $\mathbf{A} \in \mathbb{R}^{N\times R}$ is a known and fixed dictionary of size R, whose columns are basis atoms, $\mathbf{s} \in \mathbb{R}^{R\times 1}$ is the coefficient vector, $\|\mathbf{s}\|_1 = \sum_i |s_i|$ is the ℓ_1 regularization for sparsity, and λ is the regularization parameter controlling the degree of sparsity. By summing up the objective functions for all patches, we can derive a unified model for all patches in the kth band image.

$$\min_{\mathbf{S}_k} \frac{1}{2}\|\mathbf{X}_k - \mathbf{A}\mathbf{S}_k\|_F^2 + \lambda_k\|\mathbf{S}_k\|_1 \tag{2}$$

TABLE I

MAJOR NOTATIONS RELATED TO SNMF AND MTSNMF

$\mathbf{H}$	3D array of the HSI data cube of size $I \times J \times K$
I, J	size of each band image in $\mathbf{H}$
K	total number of bands in $\mathbf{H}$
k	band index, $k = 1, 2, \ldots, K$
N	size of patch ($\sqrt{N} \times \sqrt{N}$)
M	number of patches in each band, typically $M = (I - \sqrt{N} + 1)(J - \sqrt{N} + 1) \approx IJ$ when the step size of overlapping patch sampling is set to 1 pixel
$\mathbf{X}_k$	$N \times M$ matrix containing all patches in the kth band, where each column stands for a patch
$\mathbf{A}_k$	$N \times R$ dictionary matrix, where each column is a basis atom
R	size of dictionary (number of atoms in a dictionary)
$\mathbf{S}_k$	$R \times M$ coefficient matrix obtained by single-task sparse coding of the kth band (only for SNMF)
$\mathbf{S}$	$R \times M$ coefficient matrix obtained by multi-task sparse coding, which is shared by all bands (only for MTSNMF)
$\|\mathbf{S}\|_1$	$\|\mathbf{S}\|_1 = \sum\limits_{i,j} \|S_{ij}\|$ is 1-norm of coefficient matrix
$\|\cdot\|_F$	Frobenius norm
λ	a parameter that controls the degree of sparsity
σ_k	the standard deviation of noise on the kth band
λ_k	regularization parameter for the kth band, $\lambda_k = \sigma_k \lambda$
C	objective function of MTSNMF model

where $\mathbf{X}_k = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M] \in \mathbb{R}^{N \times M}$ stands for all patches within the kth band image, M is the number of patches, $\mathbf{S}_k = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_M] \in \mathbb{R}^{R \times M}$ is the coefficient matrix, $\| \cdot \|_F$ is the Frobenius norm, which implies Gaussian noise assumption, and $\|\mathbf{S}_k\|_1 = \sum\limits_{i,j} |(S_k)_{ij}|$ is 1-norm regularization for sparsity.

If the dictionary $\mathbf{A}_k$ is unknown and nonnegative constraints are imposed on both $\mathbf{A}_k$ and $\mathbf{S}_k$, SNMF is used to replace the above sparse regression model of the kth band.

$$\begin{aligned} (\hat{\mathbf{A}}_k, \hat{\mathbf{S}}_k) = \arg\min_{\mathbf{A}_k, \mathbf{S}_k} \frac{1}{2} \|\mathbf{X}_k - \mathbf{A}_k \mathbf{S}_k\|_F^2 + \lambda \|\mathbf{S}_k\|_1 \\ \text{s.t.} \quad \mathbf{A}_k \geq 0, \mathbf{S}_k \geq 0 \end{aligned} \tag{3}$$

where $\mathbf{X}_k \in \mathbb{R}_+^{N \times M}$, $\mathbf{A}_k \in \mathbb{R}_+^{N \times R}$, and $\mathbf{S}_k \in \mathbb{R}_+^{R \times M}$. Different from (2), there are two unknown matrices in (3), the dictionary matrix $\mathbf{A}_k$ for dictionary learning problem and the coefficient matrix $\mathbf{S}_k$ for sparse coding problem. In SNMF, the constraints of nonnegativity and sparsity have consistency, i.e., NMF always leads to sparse representation to a certain extent, so that the optimization for SNMF becomes to be more stable and effective. Moreover, SNMF also favors to obtain a parts-based dictionary, which more easily preserves the local structure of image so that the deformation problem of fine structure in true signal can be avoided after noise reduction.

After the learned dictionary $\hat{\mathbf{A}}_k$ and sparse coefficient matrix $\hat{\mathbf{S}}_k$ are obtained by solving the optimization problem of (3), the latent clean patches are recovered by

$$\hat{\mathbf{X}}_k = \hat{\mathbf{A}}_k \hat{\mathbf{S}}_k \tag{4}$$

Finally, we merge all denoised patches into a whole image by averaging the corresponding pixels in the denoised patches.

III. MTSNMF BASED SPECTRAL-SPATIAL HSI DENOISING

SNMF is completely dependent on the corrupted image, which makes dictionary learning and sparse coding be more or less influenced by its noise component. This is especially the case for those band images with high noise level. To further improve the denoising performance of SNMF, the correlation information between band images of HSI should be considered, which leads to the MTSNMF method in this paper. Multi-task NMF, or called simultaneous NMF, was first proposed in [42] for extraction of a common gene expression. Multi-task NMF can be seen as a combination of several strongly related NMF tasks which share a common expression, or can be seen as a special case of non-negative tensor factorization (NTF) [43].

Given K related input data matrices $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_K$, the multi-task NMF simultaneously solves K NMF problems, producing K basis matrices (dictionaries) $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_K$ and a common coefficient matrix $\mathbf{S}$.

$$\begin{aligned} \mathbf{X}_1 &= \mathbf{A}_1\mathbf{S} + \mathbf{E}_1 \\ \mathbf{X}_2 &= \mathbf{A}_2\mathbf{S} + \mathbf{E}_2 \\ &\vdots \\ \mathbf{X}_K &= \mathbf{A}_K\mathbf{S} + \mathbf{E}_K \end{aligned} \tag{5}$$

where $\mathbf{E}_k$ is the approximation error for the kth NMF task.

The multi-task NMF for HSI denoising is shown in Fig. 1. The HSI data is represented by a nonnegative three dimensional array $\mathbf{H} \in \mathbb{R}_+^{I \times J \times K}$, which contains K band images with the same spatial size of $I \times J$. In this method, image patches are extracted from band images by overlapping sliding windows respectively. For the kth band image, the observed signal matrix $\mathbf{X}_k \in \mathbb{R}_+^{N \times M}$ is formed from these patches in this band image, in which each column contains a patch, N is the patch size and M is the total number of patches. Considering the spectral correlation, we bind these related NMF tasks for all band images by sharing a common coefficient matrix, i.e., the patches in the different bands but with the same position (see Fig. 2) have the same coefficients. Similar to (3), a sparsity inducing 1-norm regularization is added to the coefficient matrix for the denoising task, thus the MTSNMF model is obtained.

$$\begin{aligned} (\hat{\mathbf{A}}_1, \ldots, \hat{\mathbf{A}}_K, \hat{\mathbf{S}}) = \underset{\mathbf{A}_1,\ldots,\mathbf{A}_K,\mathbf{S}}{\arg\min}\ & C(\mathbf{A}_1, \ldots, \mathbf{A}_K, \mathbf{S}) \\ \text{s.t.} \quad \mathbf{S} \geq 0, \quad & \text{and} \quad \forall k : \mathbf{A}_k \geq 0 \end{aligned} \tag{6}$$

$$C(\mathbf{A}_1, \ldots, \mathbf{A}_K, \mathbf{S}) = \sum_{k=1}^{K} \left(\frac{1}{2} \|\mathbf{X}_k - \mathbf{A}_k\mathbf{S}\|_F^2 + \lambda_k \|\mathbf{S}\|_1 \right) \tag{7}$$

where $\mathbf{A}_1, \ldots, \mathbf{A}_K \in \mathbb{R}_+^{N \times R}$ are the dictionaries for different bands, R is the dictionary size, N is the atom size (equal to patch size), and $\mathbf{S} \in \mathbb{R}_+^{R \times M}$ is the common coefficient matrix shared by all NMF tasks. The first term in the right side of (7) stands for the error between the observed image patches and their recovered ones, in which the

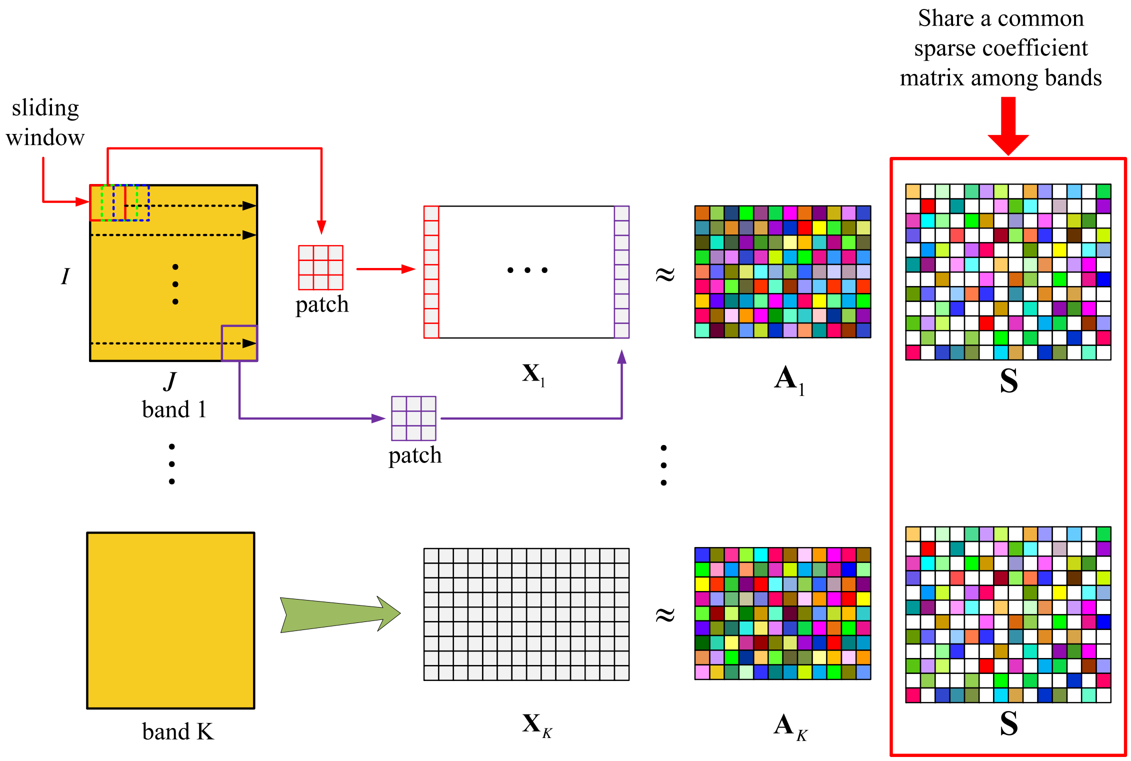


Fig. 1. The framework of multi-task NMF.

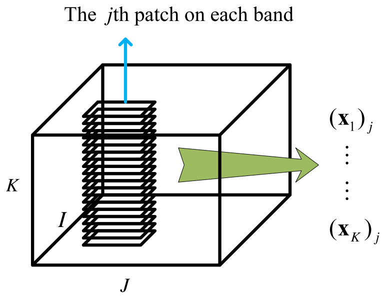


Fig. 2. Patches from different bands at the same spatial position.

Frobenius norm implies Gaussian noise assumption. The second term is sparsity inducing 1-norm regularization, where $\|\mathbf{S}\|_1 = \sum\limits_{i,j} S_{ij}$ when $\mathbf{S} \geq 0$, and $\lambda_1, \ldots, \lambda_K$ are regularization parameters for different bands, determining the degrees of sparsity. After the solution of (6), i.e., $(\hat{\mathbf{A}}_1, \ldots, \hat{\mathbf{A}}_K, \hat{\mathbf{S}})$, is obtained, the band images are recovered in a same manner as described in section II.

It has been approved that for sparse representation based signal denoising, the degree of sparsity is very critical to the performance [41]. The signal with high noise level always needs a high degree of sparsity, and vice versa. Therefore, in MTSNMF, $\lambda_k = \sigma_k \lambda$ is determined by the noise level of the kth band and the global sparsity across all bands, where σ_k is the standard deviation of noise in the kth band, and λ is a parameter to control the global sparsity. In general, the choice of λ is dependent on the empirical knowledge or validation experiment. In our experiments, following [41], the regularization parameters are set to $\lambda_k = \sigma_k\sqrt{2\log(R)}$ that depends on the dictionary size and noise level of each band, where $\lambda = \sqrt{2\log(R)}$ is considered as an approximate optimal value in practice. As the noises vary band by band in HSI, in this paper a multiple linear regression (MLR) based noise estimation algorithm is used for each band image individually. MLR method assumes that clean signal within a band image can be linearly represented by all remaining bands due to the spectral correlations, i.e, the regression error is defined as noise component [44]. Of course, other noise estimation methods also can be used. For example, [10] presented a homogeneous area based method which treats the standard deviations of pixels in the homogeneous areas as the standard deviations of noises.

In this model, the size of dictionary R is another parameter that needs to be determined. The choice of dictionary size R is related to the size of patch N. An overcomplete dictionary ($R > N$) is preferred in practice, since the overcompleteness of dictionary provides some geometric invariant properties during the dictionary learning [26]. In [15], the dictionary size is suggested to be empirically set to $R = 4N$. In our experiments, we found that the denoising performance is acceptable in the range of $R \geq 2N$, but when R goes too large, e.g. $R > 4N$, the performance does not increase any more. Taking the computational cost into further account, it is recommended that the size of dictionary is set as $2N \leq R \leq 4N$.

Like most signal denoising approaches, MTSNMF also makes assumption that there are only signal-independent Gaussian noises in HSI. However, many researches suggested that both signal-dependent and signal-independent noises co-exist in HSI [10], [39]. The signal-dependent noise is mainly caused by photons and is therefore called photon noise, which is usually represented by Poisson (or Poisson-like) distribution. The signal-independent noise is caused by electronic devices and is called thermal noise or read-out noise, which follows Gaussian distribution. A Poisson-Gaussian noise model was proposed in [10] for HSI.

$$x = \tilde{x} + n_P + n_G \tag{8}$$

where x is the noisy voxel, $\tilde{x}$ is the clean voxel, $n_G \sim \mathcal{N}(0, b)$ is the Gaussian noise component, and the Poisson noise n_P follows Poisson distribution $\frac{1}{a}(\tilde{x} + n_P) \sim \mathcal{P}(\frac{1}{a}\tilde{x})$. a and b are the parameters of Poisson and Gaussian distributions respectively. Like aforementioned variance estimation of Gaussian noise, these two parameters of the mixed noise model are also estimated by MLR method in this paper.

To make the Gaussian noise based denoising algorithms applicable to the mixed Poisson-Gaussian noise, variance stabilizing transform (VST) has been proposed in [10], [40], [45]. VST is a nonlinear mapping function that converts Poisson noise or Poisson-Gaussian noise into Gaussian noise, then various denoising algorithms based on assumption of Gaussian noise can be used. The details of VST based HSI denoising method for Poisson-Gaussian noise can be found in [10].

IV. Implementation of MTSNMF

Several algorithms have been proposed for solving NMF and related problems, for example, multiplicative update (MU) algorithm [46], alternating least squares [47], and projected gradient method [48]. In this section, we propose an MU algorithm and a hierarchical alternating least squares (HALS) algorithm for MTSNMF, together with their accelerated versions which greatly reduce the computational cost.

A. Multiplicative Update (MU) Algorithm for MTSNMF

Firstly we define two matrix operations $\circledast$ and $\oslash$, which stand for the element-wise multiplication and division, respectively. Consider the Karush-Kuhn-Tucker (KKT) conditions of (6),

$$\begin{cases} \mathbf{S} \geq 0 & (9) \\ \mathbf{A}_k \geq 0, \quad k = 1, \ldots, K & (10) \\ \nabla_{\mathbf{S}} C \geq 0 & (11) \\ \nabla_{\mathbf{A}_k} C \geq 0, \quad k = 1, \ldots, K & (12) \\ \mathbf{S} \circledast \nabla_{\mathbf{S}} C = 0 & (13) \\ \mathbf{A}_k \circledast \nabla_{\mathbf{A}_k} C = 0, \quad k = 1, \ldots, K & (14) \end{cases}$$

where

$$\nabla_{\mathbf{S}} C = \sum_{k=1}^{K} \left(\mathbf{A}_k^{\mathrm{T}} \mathbf{A}_k \mathbf{S} - \mathbf{A}_k^{\mathrm{T}} \mathbf{X}_k + \lambda_k \mathbf{1} \right) \tag{15}$$

and

$$\nabla_{\mathbf{A}_k} C = \mathbf{A}_k \mathbf{S} \mathbf{S}^{\mathrm{T}} - \mathbf{X}_k \mathbf{S}^{\mathrm{T}}, \quad k = 1, \ldots, K \tag{16}$$

Substituting (15) into (13), we have

$$\mathbf{S} \circledast \left(\sum_{k=1}^{K} \left(\mathbf{A}_k^{\mathrm{T}} \mathbf{A}_k \mathbf{S} + \lambda_k \mathbf{1} \right) \right) = \mathbf{S} \circledast \left(\sum_{k=1}^{K} \left(\mathbf{A}_k^{\mathrm{T}} \mathbf{X}_k \right) \right) \tag{17}$$

Then the multiplicative update rule for $\mathbf{S}$ is derived,

$$\mathbf{S} \leftarrow \mathbf{S} \circledast \left(\sum_{k=1}^{K} \left(\mathbf{A}_k^{\mathrm{T}} \mathbf{X}_k \right) \right) \oslash \left(\sum_{k=1}^{K} \left(\mathbf{A}_k^{\mathrm{T}} \mathbf{A}_k \mathbf{S} + \lambda_k \mathbf{1} \right) \right) \tag{18}$$

Likewise, we can obtain the multiplicative update rule for $\mathbf{A}_k$ via (14) and (16).

$$\mathbf{A}_k \leftarrow \mathbf{A}_k \circledast \left(\mathbf{X}_k \mathbf{S}^{\mathrm{T}} \right) \oslash \left(\mathbf{A}_k \mathbf{S} \mathbf{S}^{\mathrm{T}} \right), \quad k = 1, \ldots, K \tag{19}$$

The solution of MTSNMF can be obtained via alternatingly applying update rules (18) and (19). The detailed algorithm of MU is given in Alg. 1. It can be easily prove that the objective function in (7) is non-increasing under the update rules (18) and (19), which guarantees the convergence of MU algorithm.

The computational cost of each iteration in Alg. 1 is analyzed in Algs. 2 and 3 by the number of floating point operations (flops).

Algorithm 1 The MU algorithm for MTSNMF

```
Input:
    Observed HSI data $\mathbf{X}_1, \ldots, \mathbf{X}_K \in \mathbb{R}_+^{N \times M}$,
    Dictionary size $R$,
    Regularization parameters $\lambda_1, \ldots, \lambda_K$.
Output:
    Dictionary matrices $\mathbf{A}_1, \ldots, \mathbf{A}_K \in \mathbb{R}_+^{N \times R}$,
    Coefficient matrix $\mathbf{S} \in \mathbb{R}_+^{R \times M}$.
Initialize $\mathbf{A}_1, \ldots, \mathbf{A}_K$ and $\mathbf{S}$ with random numbers in $[0, 1]$.
repeat
    Fix $\mathbf{A}_1, \ldots, \mathbf{A}_K$, update $\mathbf{S}$ with MU rule (18).
    for $k = 1$ to $K$ do
        Fix $\mathbf{S}$, update $\mathbf{A}_k$ with MU rule (19).
    end for
until the maximum number of iterations has been reached, or the change of objective function cost (7) in this iteration is less than a predefined threshold
```

Algorithm 2 The MU for $\mathbf{S}$ within an iteration

```
$\mathbf{U} = \sum_{k=1}^{K} \left(\mathbf{A}_k^{\mathrm{T}} \mathbf{X}_k\right)$                          // $(2N+1)KMR$ flops
$\mathbf{V} = \sum_{k=1}^{K} \left(\mathbf{A}_k^{\mathrm{T}} \mathbf{A}_k\right)$                          // $(2N+1)KR^2$ flops
$\mathbf{W} = \mathbf{V}\mathbf{S} + \lambda_{\text{sum}} \mathbf{1}$,                            // $MR(2R+1)$ flops
    where $\lambda_{\text{sum}} = \sum_{k=1}^{K} \lambda_k$
$\mathbf{S} \leftarrow \mathbf{S} \circledast \mathbf{U} \oslash \mathbf{W}$                                  // $2MR$ flops
// Total: $(2N+1)(M+R)KR + (2R+3)MR$ flops
```

Algorithm 3 The MU for $\mathbf{A}_k$ within an iteration

```
$\mathbf{U} = \mathbf{X}_k \mathbf{S}^{\mathrm{T}}$                                          // $2NMR$ flops
$\mathbf{V} = \mathbf{S}\mathbf{S}^{\mathrm{T}}$                                            // $2MR^2$ flops
$\mathbf{W} = \mathbf{A}_k \mathbf{V}$                                           // $2NR^2$ flops
$\mathbf{A}_k \leftarrow \mathbf{A}_k \circledast \mathbf{U} \oslash \mathbf{W}$                              // $2NR$ flops
// Total: $2R(NM + MR + NR + N)$ flops
```

B. Accelerated MU (A-MU) Algorithm for MTSNMF

In order to make the MU algorithm more suitable for large HSI data, an accelerated version of MU algorithm is presented here. Based on the MU rules of standard NMF [30], [46], Gillis and Glineur proposed an accelerated multiplicative update (A-MU) algorithm [49]. The A-MU algorithm performs inner iterations to reuse the calculation results of the time-consuming steps. Their experimental results demonstrate that the A-MU converges much faster than the standard MU.

Following this idea, we propose an A-MU algorithm for MTSNMF. Note that $M \gg N$ and $M \gg R$. In our denoising model, step 1 is the most time-consuming step in Alg. 2. Therefore, to reduce the computational cost of Alg. 2, this time-consuming step shall be less executed, i.e., we should take full advantage of having computed the relatively expensive $\mathbf{U}$. It can be done by repeating the low cost steps 3-4 for several times as an inner loop while $\mathbf{U}$ and $\mathbf{V}$ are only updated once. Similarly, steps 3-4 in Alg. 3 are also repeated several times to form an inner loop.

For the A-MU algorithm, the most important issue is how to decide the inner iteration number, i.e., how many times should the low cost updating steps be repeated. Though reusing results of time-consuming steps can save the computational cost, too large inner iteration numbers will lead to an increase in flops when compared with the standard MU. Hence, we should limit the inner iteration number. In the algorithm, $\rho_{\mathbf{S}}$ and $\rho_{\mathbf{A}}$ are designed as the safeguards on inner iteration numbers for $\mathbf{S}$ and $\mathbf{A}_k$ respectively, which show when the computational cost of inner iterations in the A-MU is equal to that of a standard MU step. Denoting the computational cost on updating $\mathbf{S}$ by $T_{\mathbf{S}}$, we have $T_{\mathbf{S}}^{\text{MU}} = (2N+1)(M+R)KR + (2R+3)MR$ and $T_{\mathbf{S}}^{\text{A-MU(inner)}} = \rho_{\mathbf{S}}(2R+9)MR$, in which checking inner iteration error (Frobenius norm) costs $6MR$ flops (see line 8 of Alg. 4). Assuming that $T_{\mathbf{S}}^{\text{MU}} = T_{\mathbf{S}}^{\text{A-MU(inner)}}$, we obtain

$$\rho_{\mathbf{S}} = \frac{(2N+1)(M+R)K + (2R+3)M}{(2R+9)M} \tag{20}$$

In a similar way, $\rho_{\mathbf{A}}$ can be derived

$$\rho_{\mathbf{A}} = \frac{NM + MR + NR + N}{NR + 4N} \tag{21}$$

The stopping criteria for the inner iteration is determined by whether the maximum number of iterations ($\alpha\rho_{\mathbf{S}}$ or $\alpha\rho_{\mathbf{A}}$ where $\alpha > 0$) has been reached or the updating change in this iteration is smaller than a parameter τ. The detailed steps of the A-MU for MTSNMF are listed in Alg. 4. In the experiments of this paper, the parameters are set as $\alpha = 1$ and $\tau = 0.2$. The optimal parameter setting is dependent on the data set being used, and very difficult to be estimated, so we only give an experiential setting here.

Finally, it should be pointed out that although the A-MU is designed to reduce the computational cost, the algorithm does not theoretically guarantee to speed up the convergence rate. In other words, the A-MU can only be used as a friendly scheme in terms of computational load. Nevertheless, the lack of theoretical guarantee does not affect its applications, e.g., the A-MU has provided significant acceleration in most cases as described in [49]. We will also validate the accelerating capability of the A-MU in our experiments.

Algorithm 4 The A-MU algorithm for MTSNMF

```
Input:
    Observed HSI data $\mathbf{X}_1, \ldots, \mathbf{X}_K \in \mathbb{R}_+^{N \times M}$,
    Dictionary size $R$,
    Regularization parameters $\lambda_1, \ldots, \lambda_K$,
    Inner loop parameters $\tau$, $\rho_{\mathbf{S}}$, $\rho_{\mathbf{A}}$, and $\alpha$.
Output:
    Dictionary matrices $\mathbf{A}_1, \ldots, \mathbf{A}_K \in \mathbb{R}_+^{N \times R}$,
    Coefficient matrix $\mathbf{S} \in \mathbb{R}_+^{R \times M}$.
Initialize $\mathbf{A}_1^{(0)}, \ldots, \mathbf{A}_K^{(0)}$ and $\mathbf{S}^{(0)}$ with random numbers in $[0, 1]$.
for $t = 0, 1, 2, \ldots$ do
    $\mathbf{U} = \sum_{k=1}^{K} (\mathbf{A}_k^{(t)})^{\mathrm{T}} \mathbf{X}_k$
    $\mathbf{V} = \sum_{k=1}^{K} (\mathbf{A}_k^{(t)})^{\mathrm{T}} \mathbf{A}_k^{(t)}$
    for $l = 1$ to $\lfloor 1 + \alpha \rho_{\mathbf{S}} \rfloor$ do
        $\mathbf{W} = \mathbf{V}\mathbf{S}^{(t,l)} + \lambda_{\text{sum}} \mathbf{1}$
        $\mathbf{S}^{(t,l+1)} = \mathbf{S}^{(t,l)} \circledast \mathbf{U} \oslash \mathbf{W}$
        if $\|\mathbf{S}^{(t,l+1)} - \mathbf{S}^{(t,l)}\|_F \leq \tau \|\mathbf{S}^{(t,l+1)} - \mathbf{S}^{(t,0)}\|_F$ then
            break
        end if
    end for
    $\mathbf{S}^{(t+1)} = \mathbf{S}^{(t,l+1)}$
    $\mathbf{V} = \mathbf{S}^{(t+1)} (\mathbf{S}^{(t+1)})^{\mathrm{T}}$
    for $k = 1$ to $K$ do
        $\mathbf{U} = \mathbf{X}_k (\mathbf{S}^{(t+1)})^{\mathrm{T}}$
        for $l = 1$ to $\lfloor 1 + \alpha \rho_{\mathbf{A}} \rfloor$ do
            $\mathbf{W} = \mathbf{A}_k^{(t,l)} \mathbf{V}$
            $\mathbf{A}_k^{(t,l+1)} = \mathbf{A}_k^{(t,l)} \circledast \mathbf{U} \oslash \mathbf{W}$
            if $\|\mathbf{A}_k^{(t,l+1)} - \mathbf{A}_k^{(t,l)}\|_F \leq \tau \|\mathbf{A}_k^{(t,l+1)} - \mathbf{A}_k^{(t,0)}\|_F$ then
                break
            end if
        end for
        $\mathbf{A}_k^{(t+1)} = \mathbf{A}_k^{(t,l+1)}$
    end for
end for
```

C. Hierarchical alternating least squares (HALS) Algorithm for MTSNMF

Besides the MU algorithm, HALS algorithm [50] is also an applicable algorithm to solve MTSNMF problem. In HALS, the whole problem (7) is divided into R sub-problems of least squares, which are sequentially optimized. The cost function $C^{(j)}((\mathbf{A}_1)_{[j]}, \ldots, (\mathbf{A}_K)_{[j]}, \mathbf{S}^{[j]})$ for jth ($j = 1, 2, \ldots, R$) sub-problems is defined as

$$\begin{gathered} C^{(j)} = \sum_{k=1}^{K} \left(\frac{1}{2} \| \mathbf{X}_k^{(j)} - (\mathbf{A}_k)_{[j]} \mathbf{S}^{[j]} \|_F^2 + \lambda_k \| \mathbf{S} \|_1 \right) \\ \text{s.t.} \quad \mathbf{S} \geq 0, \quad \text{and} \quad \forall k : \mathbf{A}_k \geq 0 \end{gathered} \tag{22}$$

where $(\mathbf{A}_k)_{[j]}$ is the jth column of $\mathbf{A}_k$, $\mathbf{S}^{[j]}$ is the jth row of $\mathbf{S}$, and

$$\mathbf{X}_k^{(j)} = \mathbf{X}_k - \sum_{p \neq j} (\mathbf{A}_k)_{[p]} \mathbf{S}^{[p]} \tag{23}$$

is the residual matrix of kth task in the jth sub-problem. When we optimize one sub-problem, other parameters are fixed, e.g., when $\mathbf{S}^{[j]}$ is being optimized, other variables $\mathbf{S}^{[p]}, (p \neq j)$ and $(\mathbf{A}_k)_{[j]}, (k = 1, 2, \ldots, K, j = 1, 2, \ldots, R)$ are fixed.

The gradients of (22) with respect to the vectors $\mathbf{S}^{[j]}$ and $(\mathbf{A}_k)_{[j]}$ are expressed by

$$\nabla_{\mathbf{S}^{[j]}} C^{(j)} = \sum_{k=1}^{K} \left((\mathbf{A}_k)_{[j]}^{\mathrm{T}} (\mathbf{A}_k)_{[j]} \mathbf{S}^{[j]} - (\mathbf{A}_k)_{[j]}^{\mathrm{T}} \mathbf{X}_k^{(j)} + \lambda_k \mathbf{1} \right) \tag{24}$$

$$\nabla_{(\mathbf{A}_k)_{[j]}} C^{(j)} = (\mathbf{A}_k)_{[j]} \mathbf{S}^{[j]} (\mathbf{S}^{[j]})^{\mathrm{T}} - \mathbf{X}_k^{(j)} (\mathbf{S}^{[j]})^{\mathrm{T}} \tag{25}$$

By setting $\nabla_{\mathbf{S}^{[j]}} C^{(j)} = 0$ and $\nabla_{(\mathbf{A}_k)_{[j]}} C^{(j)} = 0$, we get the update rules,

$$\mathbf{S}^{[j]} \leftarrow \left[\frac{\sum_{k=1}^{K} \left((\mathbf{A}_k)_{[j]}^{\mathrm{T}} \mathbf{X}_k^{(j)} - \lambda_k \mathbf{1} \right)}{\sum_{k=1}^{K} (\mathbf{A}_k)_{[j]}^{\mathrm{T}} (\mathbf{A}_k)_{[j]}} \right]_+ \tag{26}$$

$$(\mathbf{A}_k)_{[j]} \leftarrow \left[\frac{\mathbf{X}_k^{(j)} (\mathbf{S}^{[j]})^{\mathrm{T}}}{\mathbf{S}^{[j]} (\mathbf{S}^{[j]})^{\mathrm{T}}} \right]_+ \tag{27}$$

where $[\xi]_+ = \max\{\xi, \epsilon\}$, and ϵ is a very small positive value to avoid numerical problems (usually $\epsilon = 10^{-16}$).

Either (26) or (27) is one step of exact block coordinate descent, which guarantees the objective function (7) to decrease [49]. Therefore, by alternatingly applying (26) and (27), HALS ensures the convergency. The detailed algorithm steps of HALS are given in Alg. 5.

We expand the details in Algs. 6 and 7 and evaluate the number of flops. It can be found that the computational costs of each iteration of MU and HALS algorithms are of the same level, but their convergence rates have little or large difference for a specific data set.

Algorithm 5 The HALS algorithm for MTSNMF

```
Input:
    Observed HSI data $\mathbf{X}_1, \dots, \mathbf{X}_K \in \mathbb{R}_+^{N \times M}$,
    Dictionary size $R$,
    Regularization parameters $\lambda_1, \dots, \lambda_K$.
Output:
    Dictionary matrices $\mathbf{A}_1, \dots, \mathbf{A}_K \in \mathbb{R}_+^{N \times R}$,
    Coefficient matrix $\mathbf{S} \in \mathbb{R}_+^{R \times M}$.
Initialize $\mathbf{A}_1, \dots, \mathbf{A}_K$ and $\mathbf{S}$ with random numbers in $[0, 1]$.
repeat
    for $j = 1$ to $R$ do
        Fix $\mathbf{A}_1, \dots, \mathbf{A}_K$ and $\mathbf{S}^{[p]}, (p \neq j)$, update $\mathbf{S}^{[j]}$ with (26).
    end for
    for $k = 1$ to $K$ do
        for $j = 1$ to $R$ do
            Fix $\mathbf{S}$ and $(\mathbf{A}_k)_{[p]}, (p \neq j)$, update $(\mathbf{A}_k)_{[j]}$ with (27).
        end for
    end for
until the maximum number of iterations has been reached, or the change of objective function cost (7) in this
    iteration is less than a predefined threshold
```

Algorithm 6 The HALS for $\mathbf{S}$ within an iteration

```
$\mathbf{U} = \sum_{k=1}^{K} \left( \mathbf{A}_k^{\mathrm{T}} \mathbf{X}_k \right)$                      // $(2N+1)KMR$ flops
$\mathbf{V} = \sum_{k=1}^{K} \left( \mathbf{A}_k^{\mathrm{T}} \mathbf{A}_k \right)$                      // $(2N+1)KR^2$ flops
for $j = 1$ to $R$ do
    $\mathbf{S}^{[j]} \leftarrow \left[ \frac{\mathbf{U}^{[j]} - \sum_{p \neq j} \mathbf{V}_{jp} \mathbf{S}^{[p]} - \lambda_{\text{sum}} \mathbf{1}}{\mathbf{V}_{jj}} \right]_+$,
                                                                             // $2M(R+1)$ flops
    where $\lambda_{\text{sum}} = \sum_{k=1}^{K} \lambda_k$
end for
// Total: $(2N+1)(M+R)KR + 2(R+1)MR$ flops
```

Algorithm 7 The HALS for $\mathbf{A}_k$ within an iteration

```
1: $\mathbf{U} = \mathbf{X}_k \mathbf{S}^{\mathrm{T}}$    // $2NMR$ flops
2: $\mathbf{V} = \mathbf{S}\mathbf{S}^{\mathrm{T}}$    // $2MR^2$ flops
3: for $j = 1$ to $R$ do
4:     $(\mathbf{A}_k)_{[j]} \leftarrow \left[ \frac{\mathbf{U}_{[j]} - \sum_{p \neq j} \mathbf{V}_{pj} (\mathbf{A}_k)_{[p]}}{\mathbf{V}_{jj}} \right]_+$
       // $(2R+1)N$ flops
5: end for
   // Total: $R(2NM + 2MR + 2NR + N)$ flops
```

D. Accelerated HALS (A-HALS) Algorithm for MTSNMF

With the same idea of A-MU, an accelerated HALS (A-HALS) can be designed [49]. In HALS, step 1 is the most time-consuming step in Alg. 6, so we repeat the updating of $\mathbf{S}$ (lines 3-5) several times while the already computed $\mathbf{U}$ and $\mathbf{V}$ are unchanged. Likewise, the updating of $\mathbf{A}_k$ (lines 3-5) in Alg. 7 can be repeated. Taking these factors into consideration, we developed the A-HALS algorithm, whose details are listed in Alg. 8.

The maximal number of inner iterations are defined as

$$\rho_{\mathbf{S}} = \frac{(2N+1)(M+R)K + 2(R+1)M}{2M(R+4)} \tag{28}$$

$$\rho_{\mathbf{A}} = \frac{2NM + 2MR + 2NR + N}{N(2R+7)} \tag{29}$$

which are derived in the same way as A-MU. All parameters in A-HALS are set as in A-MU algorithm.

V. Experimental Results

Having presented our method in the previous sections, we now turn our attention to demonstrating its utility for noise reduction of HSI. Both synthetic and real-world HSI data are used to evaluate the performance of the methods.

A. Results on Synthetic Data

The synthetic data were generated from an HSI acquired by a SpecTIR airborne system that covers the urban area on Reno, Nevada, USA [51]. This HSI data can be treated as a clean data or the ground truth as its noise level is low. The original data size is $600 \times 320 \times 356$ in which the last dimension is the number of bands. A subset of size $200 \times 200 \times 356$ was used for our experiments. The reflectance values of this image were linearly mapped to the range of $[0, 1]$. Some bands of this clean HSI are shown in Figs. 5a, 6a, and 7a.

In order to evaluate the denoising performance of the proposed method, both signal-independent Gaussian noise and signal-dependent Poisson-Gaussian noise were added to the clean HSI data. When the signal-independent Gaussian noise was added, different band images were given different noise levels, which satisfies the assumption

Algorithm 8 The A-HALS algorithm for MTSNMF

```
Input:
    Observed HSI data $\mathbf{X}_1, \ldots, \mathbf{X}_K \in \mathbb{R}_+^{N \times M}$,
    Dictionary size $R$,
    Regularization parameters $\lambda_1, \ldots, \lambda_K$,
    Inner loop parameters $\tau$, $\rho_{\mathbf{S}}$, $\rho_{\mathbf{A}}$, and $\alpha$.
Output:
    Dictionary matrices $\mathbf{A}_1, \ldots, \mathbf{A}_K \in \mathbb{R}_+^{N \times R}$,
    Coefficient matrix $\mathbf{S} \in \mathbb{R}_+^{R \times M}$.
Initialize $\mathbf{A}_1^{(0)}, \ldots, \mathbf{A}_K^{(0)}$ and $\mathbf{S}^{(0)}$ with random numbers in $[0, 1]$.
for $t = 0, 1, 2, \ldots$ do
    $\mathbf{U} = \sum_{k=1}^{K} (\mathbf{A}_k^{(t)})^{\mathrm{T}} \mathbf{X}_k$
    $\mathbf{V} = \sum_{k=1}^{K} (\mathbf{A}_k^{(t)})^{\mathrm{T}} \mathbf{A}_k^{(t)}$
    for $l = 1$ to $\lfloor 1 + \alpha \rho_{\mathbf{S}} \rfloor$ do
        for $j = 1$ to $R$ do
            $(\mathbf{S}^{(t,l+1)})^{[j]} = \left[ \frac{\mathbf{U}^{[j]} - \sum_{p \neq j} \mathbf{V}_{jp} (\mathbf{S}^{(t,l)})^{[p]} - \lambda_{\text{sum}} \mathbf{1}}{\mathbf{V}_{jj}} \right]_+$
        end for
        if $\|\mathbf{S}^{(t,l+1)} - \mathbf{S}^{(t,l)}\|_F \leq \tau \|\mathbf{S}^{(t,l+1)} - \mathbf{S}^{(t,0)}\|_F$ then
            break
        end if
    end for
    $\mathbf{S}^{(t+1)} = \mathbf{S}^{(t,l+1)}$
    $\mathbf{V} = \mathbf{S}^{(t+1)} (\mathbf{S}^{(t+1)})^{\mathrm{T}}$
    for $k = 1$ to $K$ do
        $\mathbf{U} = \mathbf{X}_k (\mathbf{S}^{(t+1)})^{\mathrm{T}}$
        for $l = 1$ to $\lfloor 1 + \alpha \rho_{\mathbf{A}} \rfloor$ do
            for $j = 1$ to $R$ do
                $(\mathbf{A}_k)_{[j]}^{(t,l+1)} = \left[ \frac{\mathbf{U}_{[j]} - \sum_{p \neq j} \mathbf{V}_{pj} (\mathbf{A}_k)_{[p]}^{(t,l)}}{\mathbf{V}_{jj}} \right]_+$
            end for
            if $\|\mathbf{A}_k^{(t,l+1)} - \mathbf{A}_k^{(t,l)}\|_F \leq \tau \|\mathbf{A}_k^{(t,l+1)} - \mathbf{A}_k^{(t,0)}\|_F$ then
                break
            end if
        end for
        $\mathbf{A}_k^{(t+1)} = \mathbf{A}_k^{(t,l+1)}$
    end for
end for
```

TABLE II

ISNR VALUES (dB) ACHIEVED BY DIFFERENT ALGORITHMS OF MTSNMF ON SYNTHETIC DATA UNDER GAUSSIAN NOISE

Algorithm	MU	A-MU	HALS	A-HALS
ISNR	16.10	16.53	16.72	16.64

that the noise levels vary band by band. The standard deviations of Gaussian distribution for different bands were selected as random numbers within interval $[0.01, 0.1]$. To facilitate the visualization of the experimental results, the selected random numbers were sorted so that $\sigma_1, \ldots, \sigma_{356}$ were set in a descending order. In other words, from the 1st band image to the 356th band image, their noise levels decreased orderly. Three noisy band images are displayed in Figs. 5b, 6b, and 7b.

The quantitative evaluation metric of denoising performance used in synthetic experiments is the Improvement in Signal-to-Noise Ratio (ISNR)

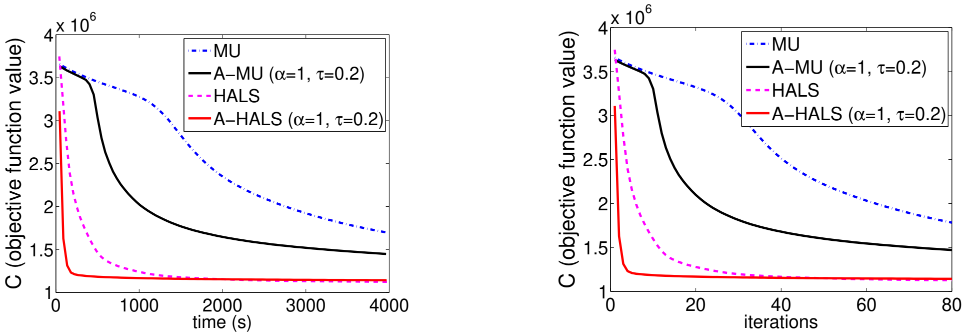
$$\begin{aligned} \mathrm{ISNR} &= \mathrm{SNR}^{\text{denoised}} - \mathrm{SNR}^{\text{noisy}} \\ &= 10\log_{10}\frac{\sum\limits_{i,j,k}(\mathbf{H}_{ijk}^{\text{noisy}} - \mathbf{H}_{ijk}^{\text{clean}})^2}{\sum\limits_{i,j,k}(\mathbf{H}_{ijk}^{\text{denoised}} - \mathbf{H}_{ijk}^{\text{clean}})^2} \end{aligned} \tag{30}$$

where $\mathbf{H}$ is the 3D data cube of the HSI, and the subscript ijk indexes the voxel position in the HSI. A higher ISNR value indicates better denoising performance.

Four algorithms MU, A-MU, HALS, A-HALS are proposed for solving MTSNMF problem. We firstly evaluate these four algorithms from the aspects of denoising performance and convergence rate respectively. With the same parameter setting $R = 255$, $\lambda = \sqrt{2\log(R)}$, $\alpha = 1$ and $\tau = 0.2$, the ISNR values obtained from these four algorithms of MTSNMF are listed in Table II. The results show that these four algorithms have the similar denoising performance. To test their convergence rate, we recorded the objective function values of (7) with respect to the time elapsed and the number of iterations in Fig. 3. This experiment is performed on a computer with Intel® Xeon® CPU E5606 @ 2.13 GHz and 24.0 GB RAM. It can be seen that whichever MU or HALS is considered, the convergence rate of the accelerated algorithm A-MU or A-HALS is faster than the original version, and the A-HALS shows the best convergence rate. Many experiments on other data sets and with other parameter settings also support this conclusion that these four optimization algorithms have the very similar denoising performance and A-HALS always has the fastest convergence rate. Based on this reason, A-HALS is adopted as the MTSNMF solver in the all following experiments.

To evaluate the proposed MTSNMF method, six state-of-the-art denoising methods are used for comparison:

- **2D K-SVD**: K-SVD is a popular sparse representation based noise reduction method for 2D images [15], where sparse coding and dictionary updating are alternatingly performed, aiming at training an adaptive dictionary which can better recover the underlying clean signal via sparse representation. 2D K-SVD is performed band by band separately, the patch size is set to 7 × 7, and various dictionary sizes (R) are used to evaluate its



(a) Function values with respect to time elapsed

(b) Function values with respect to number of iterations

Fig. 3. The convergence rate of the MU, A-MU, HALS, and A-HALS algorithms.

performance.

- **3D DWT-H** and **3D DWT-S**: Discrete wavelet transform (DWT) has been widely used as a denoising tool. When a noisy signal is transformed into wavelet domain, the coefficients with small magnitudes are considered as the coefficients of noise components. Hard and soft thresholding algorithms apply the kill and kill-or-shrink schemes, respectively, to eliminate the small coefficients using a threshold λ [52].

$$\eta_{\mathrm{H}}(w,\lambda) = \begin{cases} w & \text{if } |w| > \lambda \\ 0 & \text{otherwise} \end{cases} \tag{31}$$

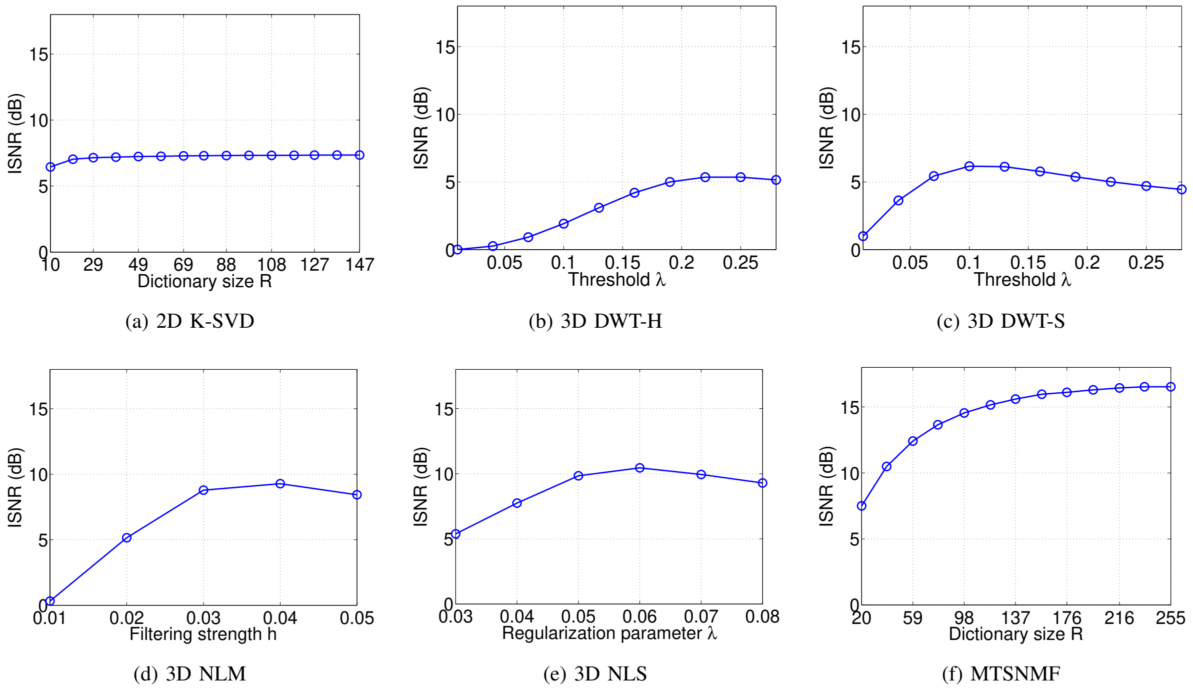
$$\eta_{\mathrm{S}}(w,\lambda) = \begin{cases} \mathrm{sgn}(w)(|w|-\lambda) & \text{if } |w| > \lambda \\ 0 & \text{otherwise} \end{cases} \tag{32}$$

In order to take the joint spectral-spatial structure of HSI into account, 3D DWT is applied for HSI denoising [19]. Here 3D DWT-H and 3D-DWT-S represent 3D DWT methods with hard and soft thresholding, respectively. Following [19], Daubechies wavelet with 4 coefficients (D4) is adopted in the experiments. A two-level decomposition is chosen because it performs better than other number of levels on the synthetic data. The different threshold values are set in order to test the denoising performance.

- **3D NLM**: To exploit the non-local similarity (self-similarity) and spectral-spatial correlation, 3D non-local means (NLM) is proposed for HSI denoising in [7]. The reflectance value of a denoised voxel $u(i)$ is calculated as a weighted average of all voxels $v(j), j \in \mathbf{H}$ in an HSI data cube $\mathbf{H}$, where their weights $w(i,j)$ are dependent on the similarities between the 3D neighborhoods $\mathcal{N}_i$ and $\mathcal{N}_j$.

$$u(i) = \sum_{j \in \mathbf{H}} w(i,j) v(j) \tag{33}$$

$$w(i,j) = \frac{1}{Z(i)} e^{-\frac{\|\mathcal{N}_i - \mathcal{N}_j\|_2^2}{h^2}} \tag{34}$$



(a) 2D K-SVD

(b) 3D DWT-H

(c) 3D DWT-S

(d) 3D NLM

(e) 3D NLS

(f) MTSNMF

Fig. 4. ISNR values with different key parameters.

where $Z(i)$ is the normalizing constant.

$$Z(i) = \sum_{j \in \mathbf{H}} e^{-\frac{\|\mathcal{N}_i - \mathcal{N}_j\|_2^2}{h^2}} \tag{35}$$

The 3D neighborhood size is set to 7 × 7 × 7, and the parameter h is evaluated in the experiments.

- **3D NLS**: The non-local sparse representation (NLS) was first proposed for image restoration [53] and then extends to 3D version for HSI denoising [10], [36]. The 3D block size is set to 7 × 7 × 7, and regularization parameter λ is tested in the experiments.
- **MTSNMF**: MTSNMF is proposed in this paper, in which the patch size is set to 7 × 7, and different dictionary sizes are evaluated in the experiments. Following [41], the regularization parameters are set to $\lambda_k = \sigma_k \sqrt{2\log(R)}$ depending on the dictionary size and noise level of each band, where R is the dictionary size and σ_k is the standard deviation of noise in the kth band.

The experimental results of all six methods with different key parameters are plotted in Fig. 4. Their highest ISNRs in the ranges of parameters are listed in Table III. It can be seen that the proposed MTSNMF outperforms the other methods by at least 5dB in ISNR. We also find from Fig. 4f that the ISNR of MTSNMF firstly increases along with the dictionary size R, and then becomes almost stable. This demonstrates that the overcompleteness of sparse representation provides some geometric invariant properties during the dictionary learning [26], which is very beneficial to true signal recovery.

Figs. 5-7 give the denoised band images (bands 10, 50, 100) of all methods with the optimal parameter values.

TABLE III
ISNR VALUES (dB) AT THE OPTIMAL PARAMETERS

2D K-SVD	3D DWT-H	3D DWT-S
7.35	5.39	6.21
3D NLM	3D NLS	MTSNMF
9.28	10.44	**16.64**

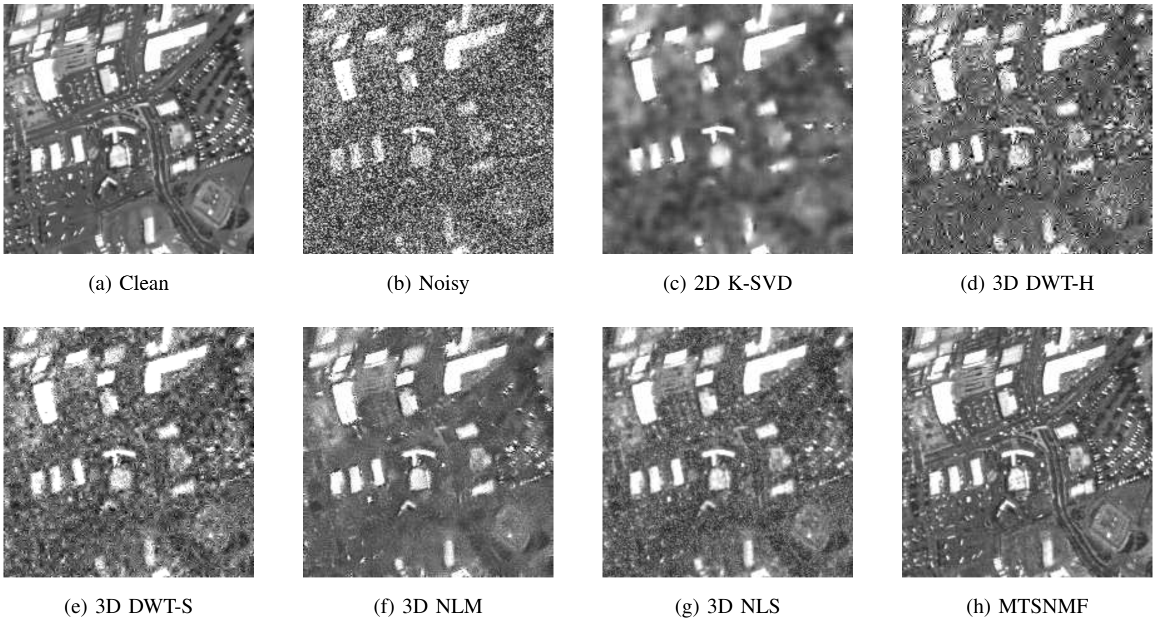
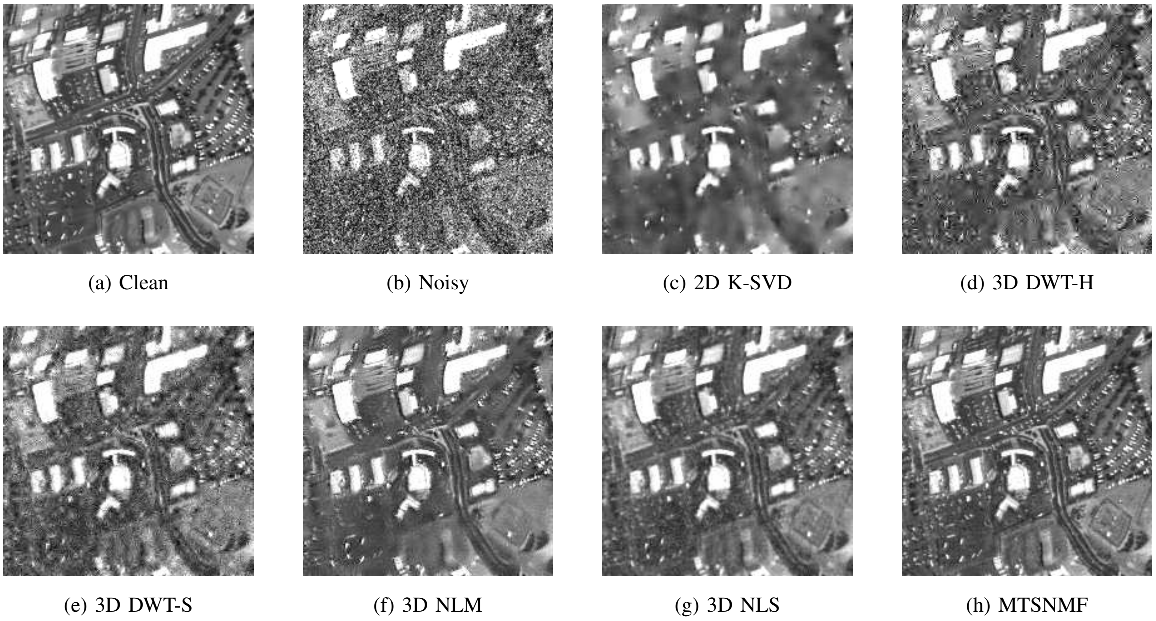

(a) Clean (b) Noisy (c) 2D K-SVD (d) 3D DWT-H
(e) 3D DWT-S (f) 3D NLM (g) 3D NLS (h) MTSNMF

Fig. 5. Band 10 of synthetic data before and after denoising under Gaussian noise.

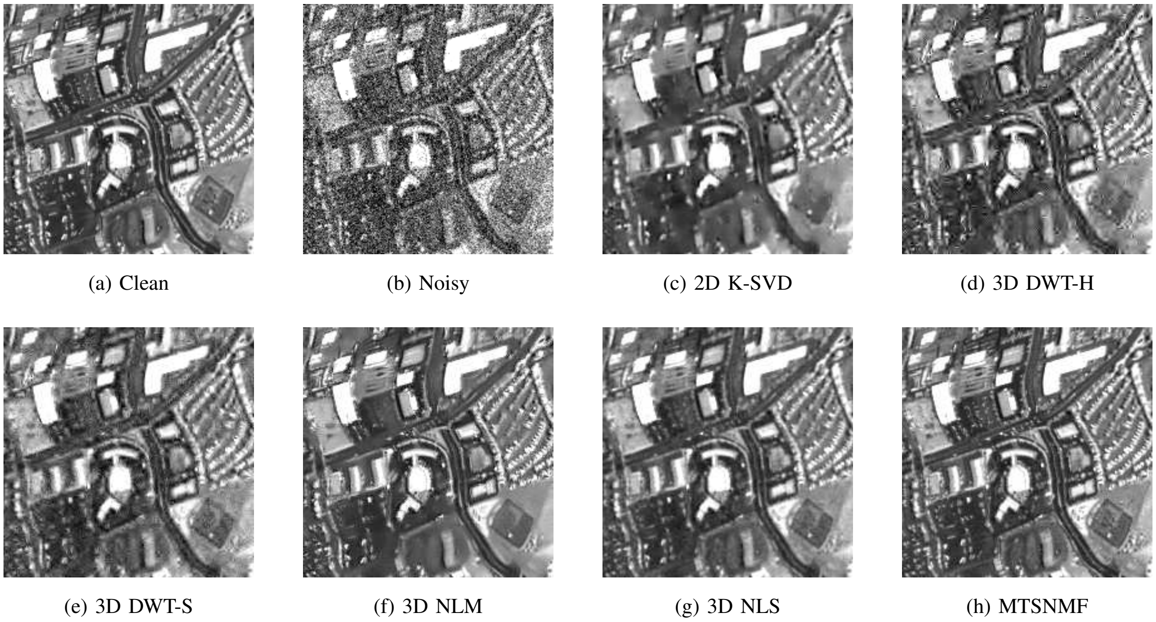
For the slightly noisy bands like band 100, all methods show competitive performance. But for heavily noisy bands like band 10, there are large variations on performance. 2D K-SVD gives over-blurred result; 3D DWT-H and 3D DWT-S deliver obvious artifacts; 3D NLM loses a lot of fine objects such as small trees and narrow roads; and 3D NLS produces acceptable recovery results, however, with slight noise and unclear fine objects. MTSNMF gives extremely favorable results which not only leave no visible noise but also keeps the fine objects clear. The excellent denoising performance of MTSNMF on heavily noisy bands is mainly due to its combination of sparse representation and joint spectral-spatial structure. To gain further intuition, we display the visual results of some spectral profiles before and after denoising in Figs. 8-9, which demonstrate that MTSNMF is able to preserve spectral details and reduce most noises.

Now we turn to mixed Poisson-Gaussian noise. We add noises to all bands with different intensities, in which the Poisson parameters for all band are generated with random numbers $a \in [0.02, 0.2]$ and sorted in descending order, while the standard deviations of the Gaussian noise are set in descending order with random numbers $b \in [0.005, 0.05]$. This experiment aims to demonstrate that MTSNMF and other denoising methods based on the


(a) Clean (b) Noisy (c) 2D K-SVD (d) 3D DWT-H

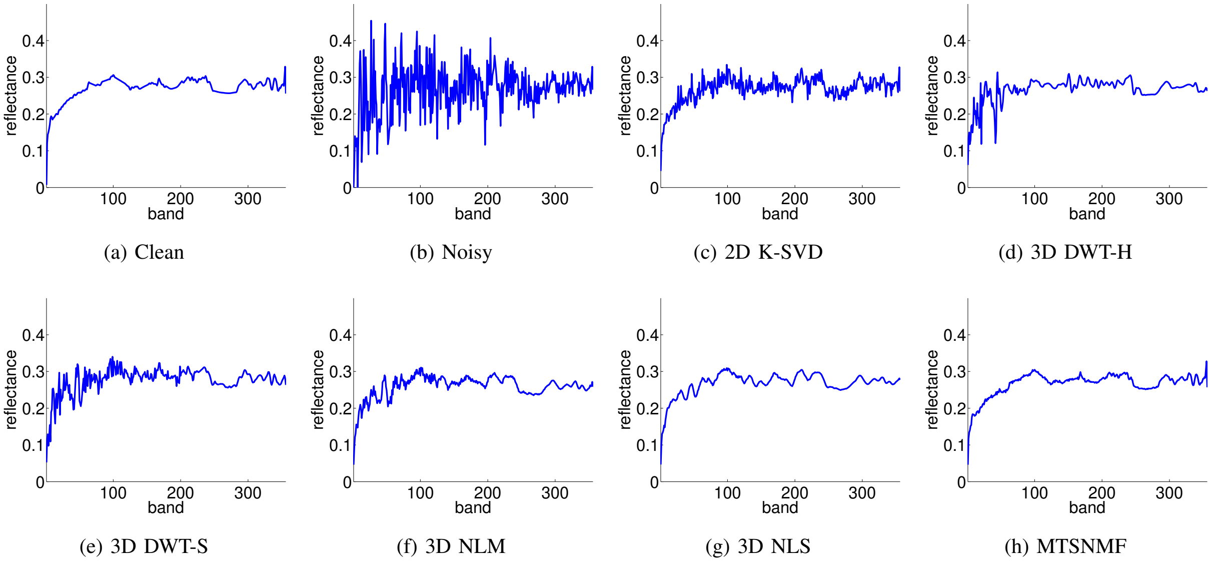
(e) 3D DWT-S (f) 3D NLM (g) 3D NLS (h) MTSNMF

Fig. 6. Band 50 of synthetic data before and after denoising under Gaussian noise.


(a) Clean (b) Noisy (c) 2D K-SVD (d) 3D DWT-H

(e) 3D DWT-S (f) 3D NLM (g) 3D NLS (h) MTSNMF

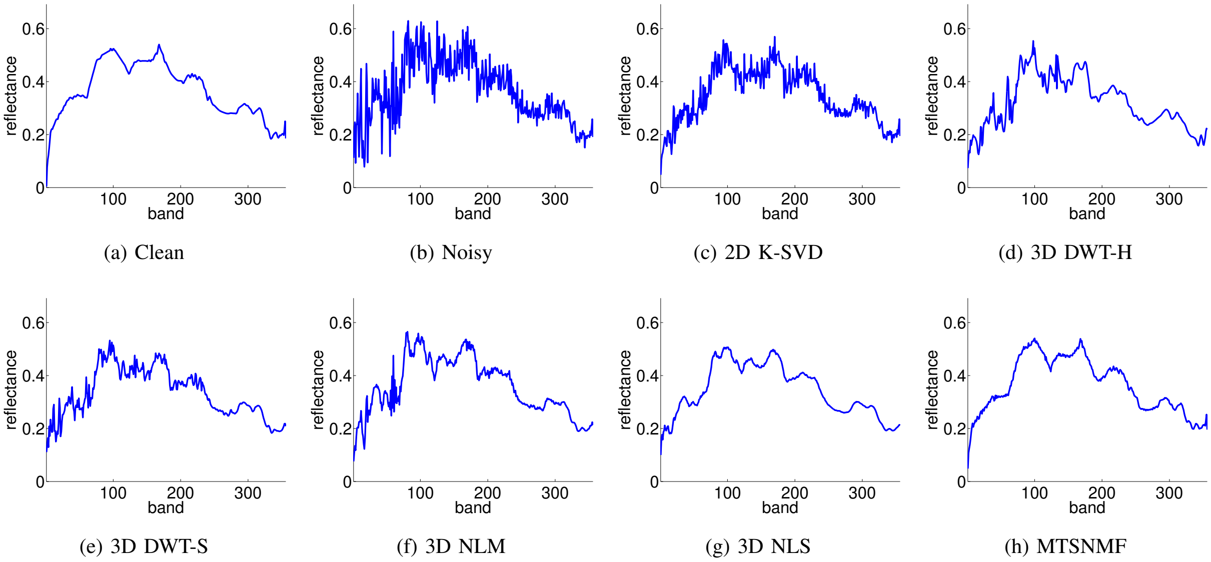
Fig. 7. Band 100 of synthetic data before and after denoising under Gaussian noise.



(a) Clean (b) Noisy (c) 2D K-SVD (d) 3D DWT-H

(e) 3D DWT-S (f) 3D NLM (g) 3D NLS (h) MTSNMF

Fig. 8. Spectral profiles of pixel $(30, 70)$ in the synthetic data before and after denoising under Gaussian noise.

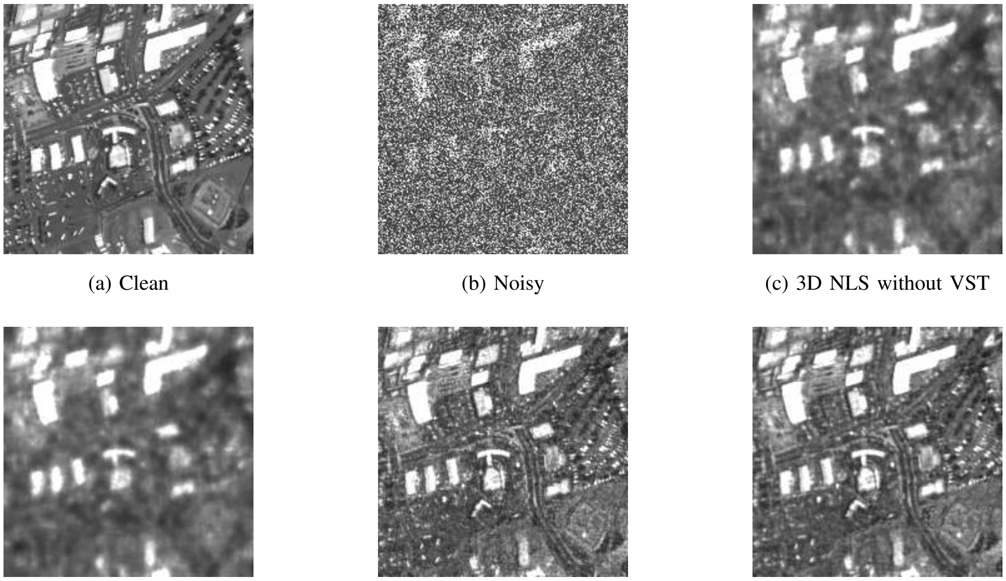


(a) Clean (b) Noisy (c) 2D K-SVD (d) 3D DWT-H

(e) 3D DWT-S (f) 3D NLM (g) 3D NLS (h) MTSNMF

Fig. 9. Spectral profiles of pixel $(100, 180)$ in the synthetic data before and after denoising under Gaussian noise.

assumption of Gaussian noise can handle the mixed Poisson-Gaussian noise via VST, so we only give the results of 3D NLS without and with VST, and MTSNMF without and with VST. The best ISNR achieved by these four methods are listed in Table IV respectively, and their band images and spectral profiles are compared in Figs. 10-13. All these results show that with the help of VST the denoising performance can be improved under Poisson-Gaussian mixed noise environment.



(a) Clean

(b) Noisy

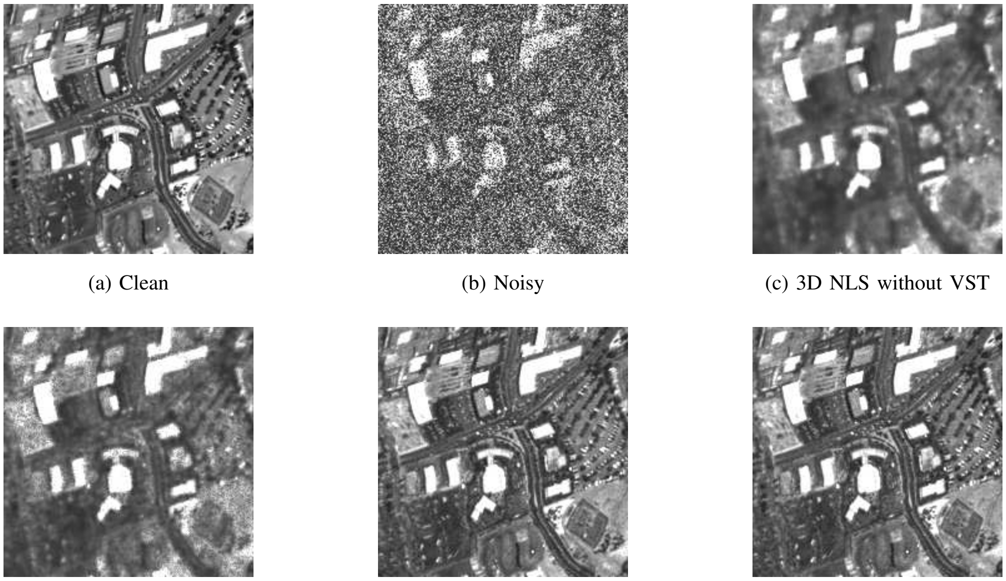
(c) 3D NLS without VST

(d) 3D NLS with VST

(e) MTSNMF without VST

(f) MTSNMF with VST

Fig. 10. Band 10 of synthetic data before and after denoising under Poisson-Gaussian noise.



(a) Clean

(b) Noisy

(c) 3D NLS without VST

(d) 3D NLS with VST

(e) MTSNMF without VST

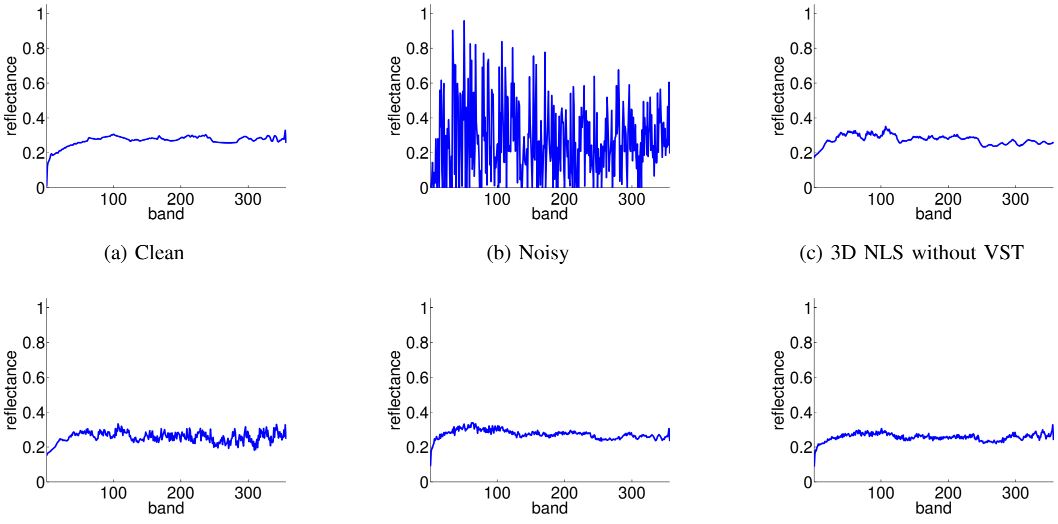
(f) MTSNMF with VST

Fig. 11. Band 200 of synthetic data before and after denoising under Poisson-Gaussian noise.

TABLE IV

ISNR VALUES (dB) AT THE OPTIMAL PARAMETERS UNDER POISSON-GAUSSIAN NOISE.

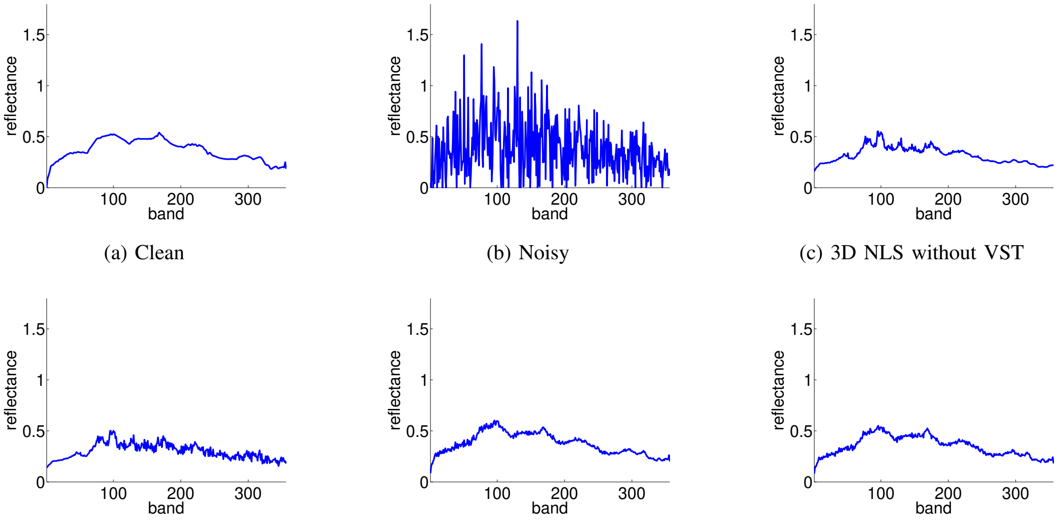
Algorithm	ISNR
3D NLS without VST	12.18
3D NLS with VST	13.37
MTSNMF without VST	15.37
MTSNMF with VST	**18.31**



(a) Clean (b) Noisy (c) 3D NLS without VST

(d) 3D NLS with VST (e) MTSNMF without VST (f) MTSNMF with VST

Fig. 12. Spectral profiles of pixel $(30, 70)$ in the synthetic data before and after denoising under Poisson-Gaussian noise.



(a) Clean (b) Noisy (c) 3D NLS without VST

(d) 3D NLS with VST (e) MTSNMF without VST (f) MTSNMF with VST

Fig. 13. Spectral profiles of pixel $(100, 180)$ in the synthetic data before and after denoising under Poisson-Gaussian noise.

B. Results on Real-world Data

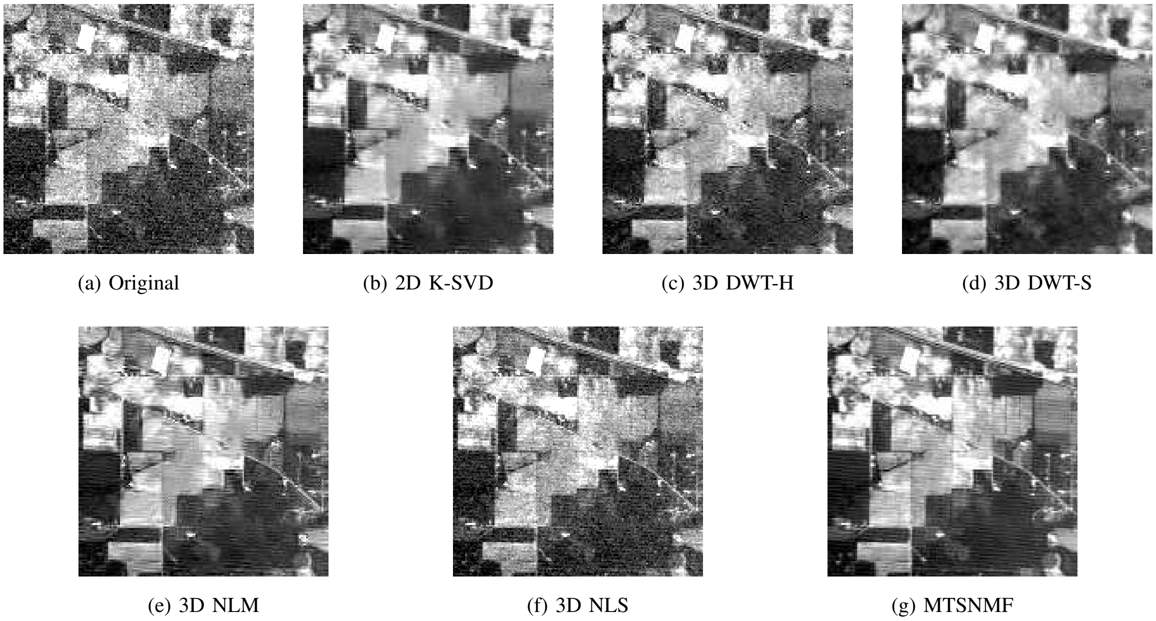
Two real-world remote sensing HSI data sets are used to evaluate the denoising performance, i.e., the Indian Pines data [54] and the Pavia University data [55]. As the underlying clean signal of real-world data is very difficult to obtain, the measure of ISNR is unavailable. Therefore, we perform qualitative evaluation based on the visual effect of band images and spectral profiles before and after denoising. Sometimes the classification results before and after denoising are used as an alternative quantitative evaluation measurement for noise reduction [8], [56]–[58], even though classification and noise reduction are two different tasks in HSI processing. Two different schemes of classification are designed here to evaluate denoising performance. Firstly, classification with all bands is used to show the advantage of denoising for classification. Then a new classification scheme based on the heavily noisy bands is proposed to further evaluate denoising performance in heavily noisy conditions. As there exist strong redundancy in spectral domain of HSI and some band images have very little noise, classification results with all band images cannot always well reflect the denoising performance. Therefore, a new approach is presented here that only uses some heavily noisy bands and their denoised versions for classification. This approach can remove the impact of those nearly clean band images, and decrease the redundancy of band images. Support vector machine (SVM) classifier with Gaussian RBF kernel is used as classifier, and one-vs-one scheme is used for multi-class problem. All parameters of the compared algorithms are set as optimal values, among which λ_k is determined by the estimated σ_k, and $R = 3N = 137$ is set for MTSNMF.

Indian Pines data set was acquired by AVIRIS hyperspectral sensor over the Indian Pine Test Site in Northwestern Indiana. The data set consists of 145×145 pixels with 220 bands. The denoising results of are presented in Figs. 14-17. For a less noisy band, e.g., band 3, the compared methods produce similar results, while for a seriously corrupted band, e.g., band 106 or 220, MTSNMF gives much more favorable results than the rest methods.

For classification based evaluation, two experiments are implemented, one uses all bands, and the other is based on only 20 heavily noisy bands including 104-108, 150-163, 220. There are 16 land cover classes and the corresponding labeled samples in the Indian Pines data set [54]. The number of training samples is set as 50 per class, except very small classes, to which we assign 15 training samples as in [59]. The training samples are randomly selected, and the rest labeled samples are treated as test samples. The experiment is repeated 20 times. The mean values and standard deviations of overall accuracy (OA), average accuracy (AA), and kappa coefficient (κ) obtained from these two classification experiments are reported in Tables V and VI, respectively. The classification results demonstrate that MTSNMF can recover the structural information (which is necessary for classification) even if the bands are heavily noisy.

Pavia University data set was captured with ROSIS hyperspectral sensor during a flight campaign over the Pavia University, Pavia city, Italy. The HSI data has 103 spectral bands, and each band has 610×340 pixels. The geometric resolution is 1.3 meters. Bands 1-4 have high noise levels. Figs. 18-20 show the denoising results of different methods.

Results on the Pavia University data are similar to those on the synthetic data: 2D K-SVD produces over-blurred



(a) Original

(b) 2D K-SVD

(c) 3D DWT-H

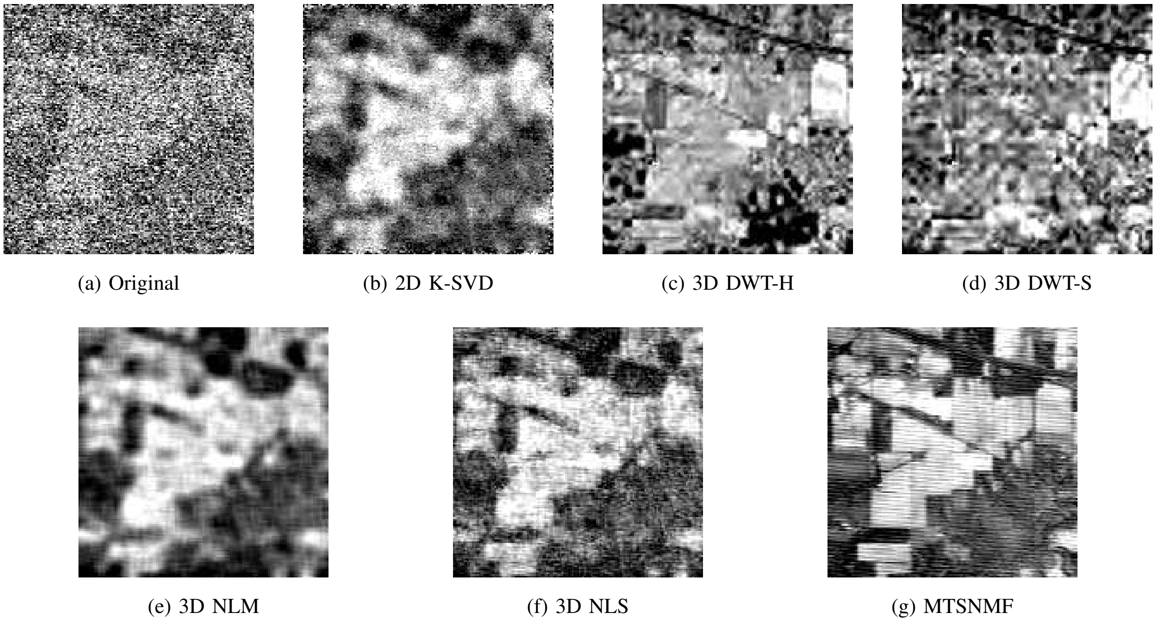
(d) 3D DWT-S

(e) 3D NLM

(f) 3D NLS

(g) MTSNMF

Fig. 14. Band 3 of Indian Pines data before and after denoising.



(a) Original

(b) 2D K-SVD

(c) 3D DWT-H

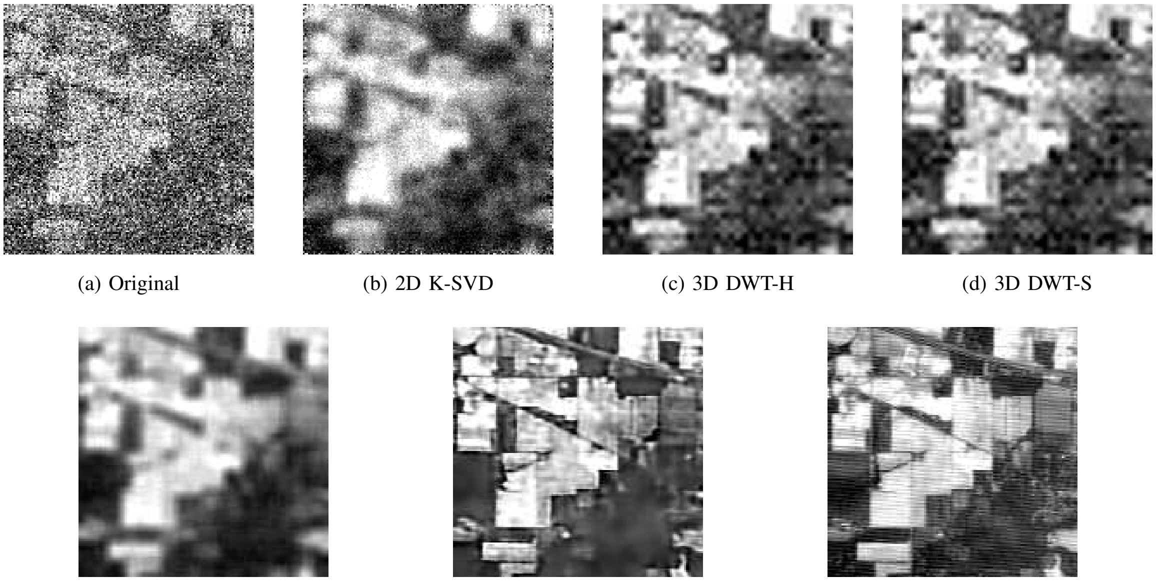
(d) 3D DWT-S

(e) 3D NLM

(f) 3D NLS

(g) MTSNMF

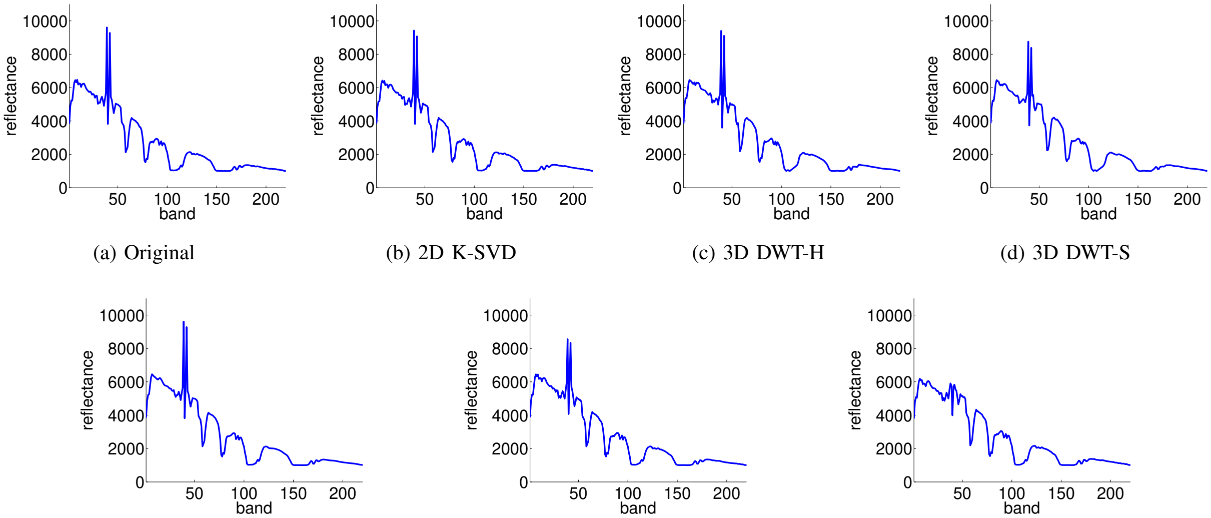
Fig. 15. Band 106 of Indian Pines data before and after denoising.



(a) Original (b) 2D K-SVD (c) 3D DWT-H (d) 3D DWT-S

(e) 3D NLM (f) 3D NLS (g) MTSNMF

Fig. 16. Band 220 of Indian Pines data before and after denoising.



(a) Original (b) 2D K-SVD (c) 3D DWT-H (d) 3D DWT-S

(e) 3D NLM (f) 3D NLS (g) MTSNMF

Fig. 17. Spectral profiles of pixel $(18, 52)$ in the Indian Pines data before and after denoising.

TABLE V

MEANS AND STANDARD DEVIATIONS OF OA, AA, AND κ ON ALL BANDS OF INDIAN PINES DATA

Algorithm	OA ($\mathbf{std}$(OA))	AA ($\mathbf{std}$(AA))	κ ($\mathbf{std}(\kappa)$)
Original (bands 104-108,150-163,220 removed)	75.19% (1.10×10^{-2})	83.52% (1.22×10^{-2})	71.99% (1.19×10^{-2})
2D K-SVD	83.89% (8.91×10^{-3})	91.06% (6.41×10^{-3})	81.73% (9.73×10^{-3})
3D DWT-H	75.86% (1.47×10^{-2})	84.62% (1.48×10^{-2})	72.75% (1.62×10^{-2})
3D DWT-S	80.28% (1.35×10^{-2})	88.58% (1.21×10^{-2})	77.69% (1.52×10^{-2})
3D NLM	85.70% (1.17×10^{-2})	**92.18%** (8.28×10^{-3})	83.77% (1.31×10^{-2})
3D NLS	85.20% (1.22×10^{-2})	92.10% (8.02×10^{-3})	83.21% (1.36×10^{-2})
MTSNMF	**86.62%** (8.90×10^{-3})	90.72% (9.96×10^{-3})	**84.79%** (9.94×10^{-3})

TABLE VI

MEANS AND STANDARD DEVIATIONS OF OA, AA, AND κ ON HEAVILY NOISY BANDS (104-108,150-163,220) OF INDIAN PINES DATA

Algorithm	OA ($\mathbf{std}$(OA))	AA ($\mathbf{std}$(AA))	κ ($\mathbf{std}(\kappa)$)
Original	15.52% (1.05×10^{-2})	13.42% (8.03×10^{-3})	8.35% (8.78×10^{-3})
2D K-SVD	54.20% (1.55×10^{-2})	64.55% (1.99×10^{-2})	48.93% (1.62×10^{-2})
3D DWT-H	58.48% (1.25×10^{-2})	71.91% (1.44×10^{-2})	53.64% (1.29×10^{-2})
3D DWT-S	62.35% (1.03×10^{-2})	74.76% (1.55×10^{-2})	57.90% (1.18×10^{-2})
3D NLM	64.73% (1.23×10^{-2})	79.19% (1.68×10^{-2})	60.55% (1.33×10^{-2})
3D NLS	69.41% (1.63×10^{-2})	81.33% (1.36×10^{-2})	65.69% (1.76×10^{-2})
MTSNMF	**75.33%** (1.10×10^{-2})	**84.29%** (1.15×10^{-2})	**72.11%** (1.19×10^{-2})

image; recognizable recovery artifacts remain in the results of 3D DWT-H and 3D DWT-S; 3D NLM, 3D NLS, and MTSNMF performs well on both noise reduction and detail preservation, but MTSNMF behaves even better in preserving some very fine details. As most bands in the Pavia University data have very weak noise, if a denoising method takes both of spectral correlation and spatial correlation information into account, a few band images with high noise level can be well recovered via using the related clean signal in other bands. 2D K-SVD processes the HSI data band by band so that the spectral correlation between bands is not taken into account. Although 3D DWT methods use the spectral and spatial neighboring information of voxel, other correlation information beyond neighborhood is ignored. As the heavily noisy bands concentrating upon from 1 to 4, their neighboring information is not sufficient to provide enough reference about clean signal for noise reduction. Therefore, both 3D DWT-H and 3D DWT-S cannot produce excellent results either. Non-local methods can use similarity information in a whole HSI cube, so 3D NLM and 3D NLS are able to generate good denoised results. In MTSNMF, the shared coefficients are imposed on all bands, hence, it also can use the spectral-spatial information across all bands rather than the neighboring bands.

Two classification experiments are performed respectively, one is based on full bands and the other is based on four heavily noisy bands 1-4. There are nine land cover classes and the corresponding labeled samples in Pavia University data set [55], which can be found in [60]. Like the classification on Indian Pines data set, the number of

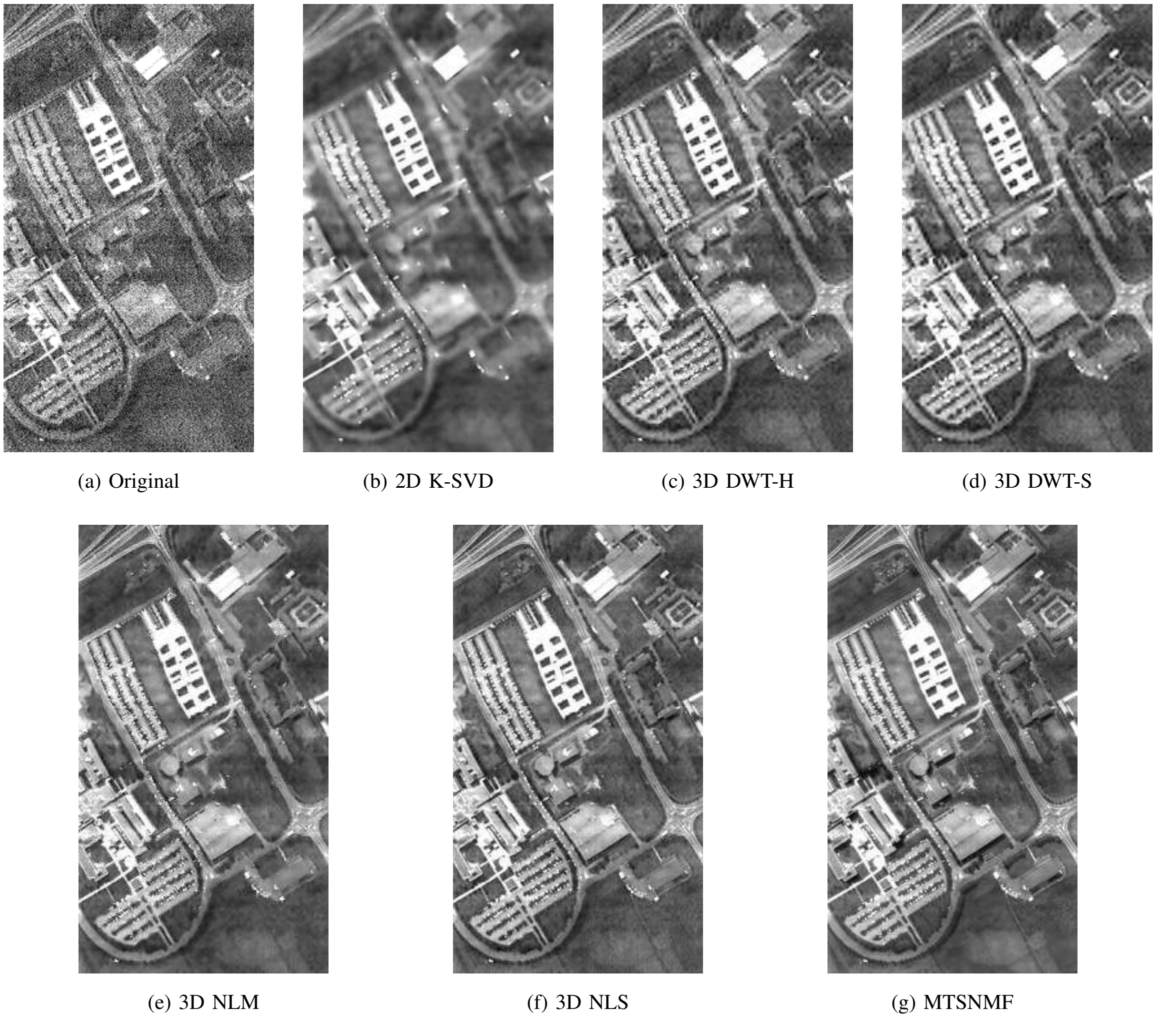
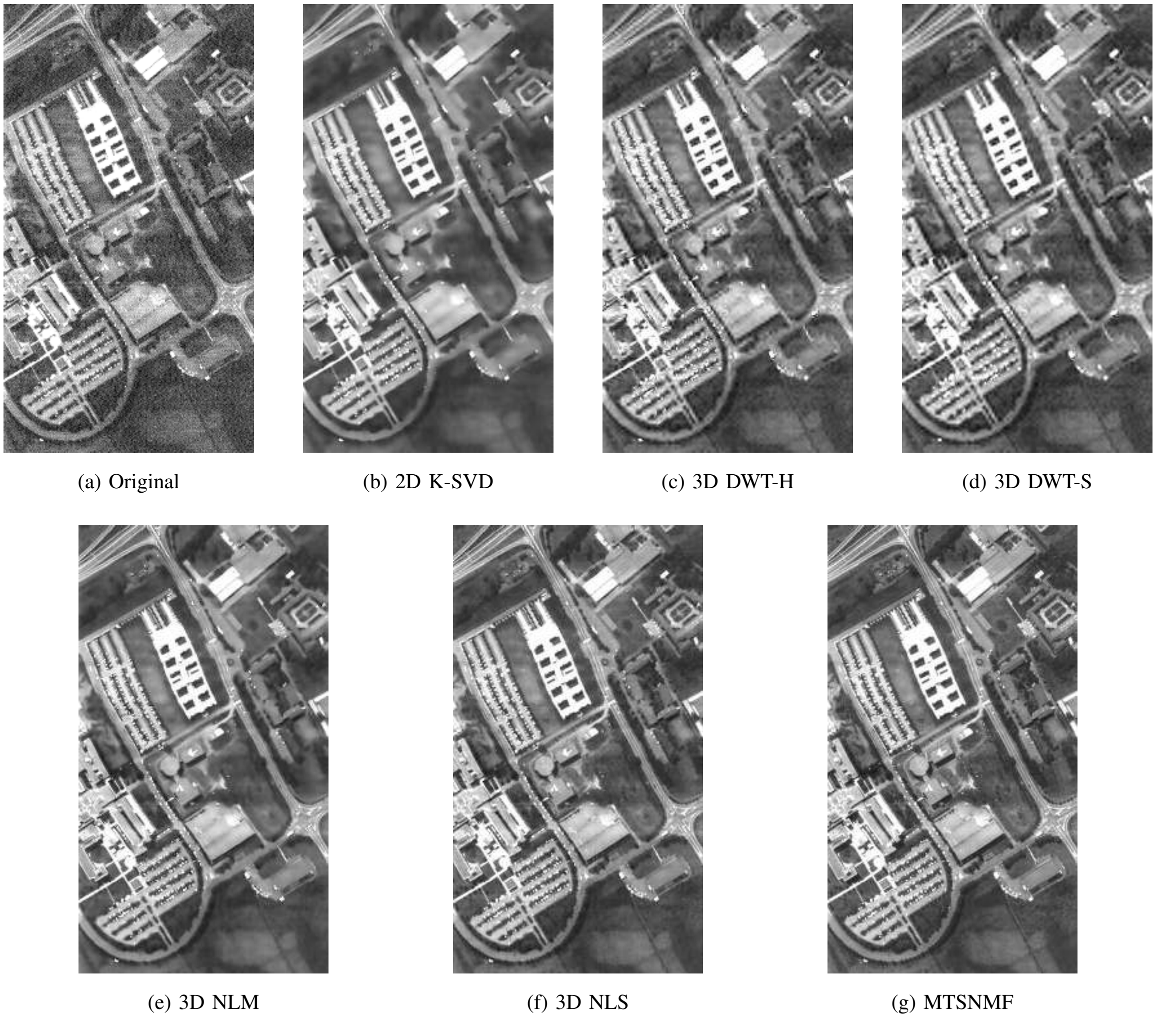

(a) Original
(b) 2D K-SVD
(c) 3D DWT-H
(d) 3D DWT-S
(e) 3D NLM
(f) 3D NLS
(g) MTSNMF

Fig. 18. Band 1 of Pavia University data before and after denoising.

training samples is set as 50 per class. The OA, AA, κ of two classification experiments are reported in Tables VII and VIII, respectively. It can be seen that MTSNMF shows excellent classification performance that outperforms most of other compared denoising methods.

Finally, we discuss the computational cost of each denoising method and the corresponding running time on the real-world data. The computation complexities of all compared methods are listed in the Table IX. The size of 2D patches/3D blocks is represented as N. R is the dictionary size. L is the average number of nonzero elements in each coefficient vector in 2D K-SVD algorithm. M is the number of patches, and $M \approx IJ$ when the step size of overlapping patch sampling is set to 1 pixel. It should be noted that, all proposed MTSNMF solvers including MU, A-MU, HALS, and A-HALS have the same computational complexity, though they have different convergence rates.

Time costs on Indian Pines data and Pavia University are shown in Table X. Seen from Tables IX and X, there



(a) Original (b) 2D K-SVD (c) 3D DWT-H (d) 3D DWT-S

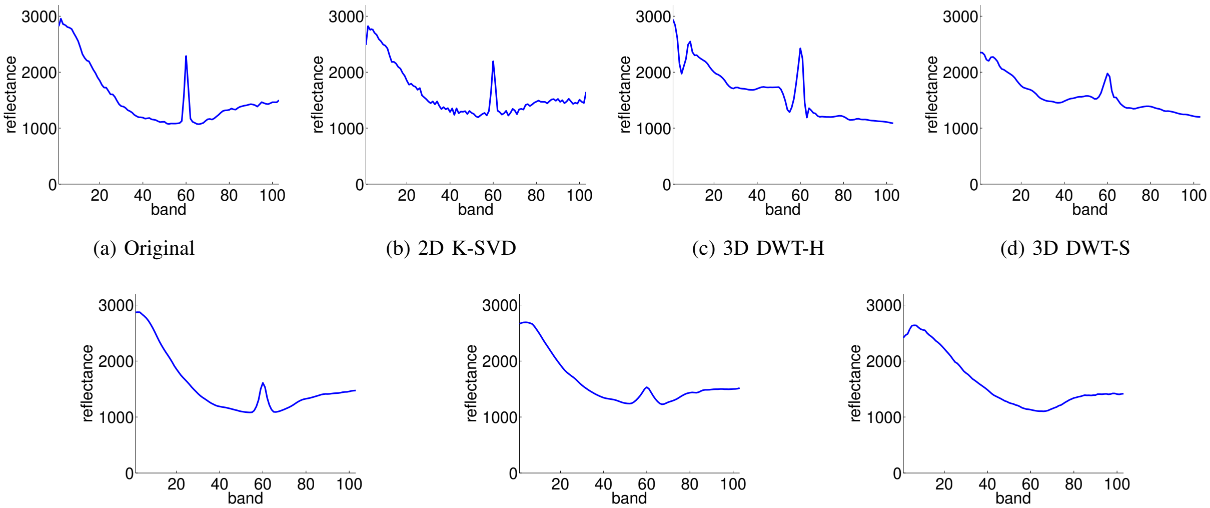
(e) 3D NLM (f) 3D NLS (g) MTSNMF

Fig. 19. Band 3 of Pavia University data before and after denoising.

TABLE VII

MEANS AND STANDARD DEVIATIONS OF OA, AA, AND κ ON ALL BANDS OF PAVIA UNIVERSITY DATA

Algorithm	OA ($\mathbf{std}$(OA))	AA ($\mathbf{std}$(AA))	κ ($\mathbf{std}(\kappa)$)
Original (bands 1-4 removed)	93.28% (3.68×10^{-3})	93.66% (2.15×10^{-3})	90.90% (4.81×10^{-3})
2D K-SVD	**98.37%** (1.57×10^{-3})	**98.05%** (1.25×10^{-3})	**97.77%** (2.14×10^{-3})
3D DWT-H	92.45% (2.38×10^{-3})	91.20% (1.59×10^{-3})	89.74% (3.15×10^{-3})
3D DWT-S	96.26% (1.98×10^{-3})	95.42% (1.81×10^{-3})	94.90% (2.67×10^{-3})
3D NLM	97.25% (2.06×10^{-3})	97.01% (1.74×10^{-3})	96.24% (2.78×10^{-3})
3D NLS	97.27% (1.44×10^{-3})	96.97% (1.56×10^{-3})	96.27% (1.94×10^{-3})
MTSNMF	97.31% (1.83×10^{-3})	97.17% (1.77×10^{-3})	96.34% (2.48×10^{-3})



(a) Original

(b) 2D K-SVD

(c) 3D DWT-H

(d) 3D DWT-S

(e) 3D NLM

(f) 3D NLS

(g) MTSNMF

Fig. 20. Spectral profiles of pixel $(296, 91)$ in Pavia University data before and after denoising.

TABLE VIII

MEANS AND STANDARD DEVIATIONS OF OA, AA, AND κ ON HEAVILY NOISY BANDS (1-4) OF PAVIA UNIVERSITY DATA

Algorithm	OA ($\mathbf{std}$(OA))	AA ($\mathbf{std}$(AA))	κ ($\mathbf{std}(\kappa)$)
Original	29.99% (3.01×10^{-2})	40.04% (1.12×10^{-2})	19.19% (1.88×10^{-2})
2D K-SVD	48.52% (3.33×10^{-2})	56.46% (1.12×10^{-2})	37.49% (2.74×10^{-2})
3D DWT-H	44.57% (3.42×10^{-2})	52.19% (1.40×10^{-2})	33.47% (2.89×10^{-2})
3D DWT-S	45.66% (3.23×10^{-2})	53.31% (1.17×10^{-2})	34.48% (2.45×10^{-2})
3D NLM	46.66% (3.94×10^{-2})	56.65% (9.56×10^{-3})	36.00% (3.28×10^{-2})
3D NLS	54.13% (2.30×10^{-2})	61.69% (8.60×10^{-3})	43.62% (2.11×10^{-2})
MTSNMF	**72.66%** (2.13×10^{-2})	**73.25%** (7.79×10^{-3})	**64.92%** (2.34×10^{-2})

TABLE IX

COMPUTATIONAL COMPLEXITY OF DIFFERENT METHODS

2D K-SVD	3D DWT-H	3D DWT-S
$\mathcal{O}(RNMKL)$	$\mathcal{O}(IJK)$	

3D NLM	3D NLS	MTSNMF
$\mathcal{O}(N(IJK)^2)$	$\mathcal{O}(RNIJK)$	$\mathcal{O}(RMK(2N+R))$

TABLE X

TIME COST ON INDIAN PINES AND PAVIA UNIVERSITY DATA WITH DIFFERENT ALGORITHMS

Algorithm	Indian Pines	Pavia University
2D K-SVD (Matlab and C)	534.35s	439.24s
3D DWT-H (Matlab)	2.49s	10.52s
3D DWT-S (Matlab)	2.44s	9.91s
3D NLM (Matlab and C)	163.18s	261.09s
3D NLS (Matlab and C)	3113.77s	18462.75s
MTSNMF (Matlab)	638.56s	2526.62s
MTSNMF (CUDA)	197.19s	802.28s

is inconsistency between the theoretic computational complexity and the real running time for some denoising algorithms. The main reason is that MTSNMF, 2D K-SVD and 3D NLS are iterative algorithms which are highly time-consuming due to large iteration number, but the big $\mathcal{O}$ notation in computational complexity does not consider this factor. In order to make MTSNMF more practical, we implemented a CUDA version of MTSNMF to decrease the running time by parallel computing. In the experiment, we used a computer with Intel® Xeon® CPU E5606 @ 2.13 GHz, NVIDIA® GeForce® GTX 650 graphics card, and 24.0 GB RAM. It can be seen that the running time can be decreased by more than 3 times. How to further decrease the computational complexity of MTSNMF is the focus of our future work.

VI. CONCLUSIONS

This paper proposes a novel MTSNMF method for sparse representation based HSI denoising. MTSNMF has three distinct properties. The first is that dictionary learning and sparse coding are unified into an integrated SNMF model, which makes these two problems of sparse representation more adaptive to the observed signal. NMF is a parts-based dictionary learning method, whereas SNMF decomposes the observed signals into a dictionary containing object parts along with the sparse coefficients to recover the true signal. The second property is extending the SNMF based 2D image denoising model to 3D HSI by multi-task learning, which makes use of the joint spectral-spatial structure for sparse representation. Instead of performing SNMF band by band, MTSNMF binds SNMF tasks of different bands together via sharing a common coefficient matrix among these bands. The third property is incorporating noise information into the model by linking noise estimation and the parameter of MTSNMF, which enables the model parameter be estimated by the noise level. Moreover, VST is combined with Gaussian noise model based MTSNMF to deal with mixed Poisson-Gaussian noise, which extends the wide application of MTSNMF. Compared with several existing HSI denoising approaches, such as wavelet transform algorithms, non-local algorithms, and other sparse representation algorithms, MTSNMF can well preserve the intrinsic details of the spectral and spatial structures whilst significantly remove noise. It is especially effective for those band images with heavy noises, which is due to the capability of MTSNMF in exploiting to a great extent the correlation information in the spatial, the spectral, and the cross spectral-spatial domains.

References

[1] H. Othman and S.-E. Qian, "Noise reduction of hyperspectral imagery using hybrid spatial-spectral derivative-domain wavelet shrinkage," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 2, pp. 397–408, 2006.

[2] G. Chen and S.-E. Qian, "Denoising of hyperspectral imagery using principal component analysis and wavelet shrinkage," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 3, pp. 973–980, 2011.

[3] D. Xu, L. Sun, J. Luo, and Z. Liu, "Analysis and denoising of hyperspectral remote sensing image in the curvelet domain," *Math. Probl. Eng.*, vol. 2013, pp. 1–11, 2013.

[4] M. Lennon, G. Mercier, and L. Hubert-Moy, "Nonlinear filtering of hyperspectral images with anisotropic diffusion," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, vol. 4, 2002, pp. 2477–2479.

[5] D. Letexier and S. Bourennane, "Noise removal from hyperspectral images by multidimensional filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 7, pp. 2061–2069, 2008.

[6] A. Heo, J.-H. Lee, E.-J. Choi, W.-C. Choi, S. Kim, and D.-J. Park, "Noise reduction of hyperspectral images using a joint bilateral filter with fused images," in *Proc. SPIE*, 2011.

[7] Y. Qian, Y. Shen, M. Ye, and Q. Wang, "3-D nonlocal means filter with noise estimation for hyperspectral imagery denoising," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2012, pp. 1345–1348.

[8] Q. Yuan, L. Zhang, and H. Shen, "Hyperspectral image denoising employing a spectral-spatial adaptive total variation model," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 10, pp. 3660–3677, 2012.

[9] S. Bourguignon, D. Mary, and E. Slezak, "Sparsity-based denoising of hyperspectral astrophysical data with colored noise: Application to the MUSE instrument," in *2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 2010.

[10] Y. Qian and M. Ye, "Hyperspectral imagery restoration using nonlocal spectral-spatial structured sparse representation with noise estimation," *IEEE J. Select. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 2, pp. 499–515, 2013.

[11] B. Rasti, J. R. Sveinsson, M. O. Ulfarsson, and J. A. Benediktsson, "Hyperspectral image denoising using a new linear model and sparse regularization," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2013, pp. 457–460.

[12] J. Martin-Herrero and M. Ferreiro-Arman, "Tensor-driven hyperspectral denoising: A strong link for classification chains?" in *Proc. IEEE 20th Int. Conf. Patt. Recogn.*, 2010, pp. 2820–2823.

[13] A. Karami, M. Yazdi, and A. Zolghadre Asli, "Noise reduction of hyperspectral images using kernel non-negative Tucker decomposition," *IEEE J. Select. Topics Signal Process.*, vol. 5, no. 3, pp. 487–493, 2011.

[14] D. Liao, M. Ye, S. Jia, and Y. Qian, "Noise reduction of hyperspectral imagery based on nonlocal tensor factorization," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2013, pp. 1083–1086.

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

[16] S. Valiollahzadeh, H. Firouzi, M. Babaie-Zadeh, and C. Jutten, "Image denoising using sparse representations," in *Independent Component Analysis and Signal Separation*. Springer, 2009, pp. 557–564.

[17] M. Elad, M. A. T. Figueiredo, and Y. Ma, "On the role of sparse and redundant representations in image processing," *Proc. IEEE*, vol. 98, no. 6, pp. 972–982, 2010.

[18] M. Ye and Y. Qian, "Mixed Poisson-Gaussian noise model based sparse denoising for hyperspectral imagery," in *4th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 2012.

[19] B. Rasti, J. Sveinsson, M. Ulfarsson, and J. Benediktsson, "Hyperspectral image denoising using 3D wavelets," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2012, pp. 1349–1352.

[20] Y. Zhao and J. Yang, "Hyperspectral image denoising via sparsity and low rank," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2013, pp. 1091–1094.

[21] E. Simoncelli, W. Freeman, E. Adelson, and D. Heeger, "Shiftable multiscale transforms," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 587–607, 1992.

[22] I. Tosic and P. Frossard, "Dictionary learning," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 27–38, 2011.

[23] K. Engan, S. Aase, and J. Hakon Husoy, "Method of optimal directions for frame design," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 5, 1999, pp. 2443–2446.

[24] R. Vidal, Y. Ma, and S. Sastry, “Generalized principal component analysis (GPCA),” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1945–1959, 2005.

[25] M. Aharon, M. Elad, and A. Bruckstein, “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation,” *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, 2006.

[26] R. Rubinstein, A. M. Bruckstein, and M. Elad, “Dictionaries for sparse representation modeling,” *Proc. IEEE*, vol. 98, no. 6, pp. 1045–1057, 2010.

[27] S. Yang, Z. Liu, M. Wang, F. Sun, and L. Jiao, “Multitask dictionary learning and sparse representation based single-image super-resolution reconstruction,” *Neurocomputing*, vol. 74, no. 17, pp. 3193–3203, 2011.

[28] M. Ye, Y. Qian, and Q. Wang, “Panchromatic image based dictionary learning for hyperspectral imagery denoising,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2013, pp. 4130–4133.

[29] Z. Xing, M. Zhou, A. Castrodad, G. Sapiro, and L. Carin, “Dictionary learning for noisy and incomplete hyperspectral images,” *SIAM J. Imag. Sci.*, vol. 5, no. 1, pp. 33–56, 2012.

[30] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[31] M. Heiler and C. Schnörr, “Learning sparse representations by non-negative matrix factorization and sequential cone programming,” *J. Mach. Learn. Res.*, vol. 7, pp. 1385–1407, 2006.

[32] Z. Tang, S. Ding, Z. Li, and L. Jiang, “Dictionary learning based on nonnegative matrix factorization using parallel coordinate descent,” *Abstr. Appl. Anal.*, vol. 2013, pp. 1–11, 2013.

[33] R. Farouk and H. Khalil, “Image denoising based on sparse representation and non-negative matrix factorization,” *Life Sci. J.*, vol. 9, no. 1, pp. 337–341, 2012.

[34] L. Shang, Y. Zhou, J. Chen, and Z. Sun, “Image denoising using a modified LNMF algorithm,” in *Proc. Int Conf. Comput. Sci. Ser. Sys.*, 2012, pp. 1840–1843.

[35] P. O. Hoyer, “Non-negative sparse coding,” in *Proc. 12th IEEE Workshop Neural Net. Signal Process.*, 2002, pp. 557–565.

[36] Y. Qian, M. Ye, and Q. Wang, “Noise reduction of hyperspectral imagery using nonlocal sparse representation with spectral-spatial structure,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2012, pp. 3467–3470.

[37] S. Chen, X. Hu, and S. Peng, “Denoising of hyperspectral imagery using a spatial-spectral domain mixing prior,” in *Proc. Int. Conf. Geoinformatics*, 2012, pp. 1–7.

[38] T. Evgeniou and M. Pontil, “Regularized multi-task learning,” in *Proc. ACM SIGKDD*. ACM, 2004, pp. 109–117.

[39] N. Acito, M. Diani, and G. Corsini, “Signal-dependent noise modeling and model parameter estimation in hyperspectral images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 8, pp. 2957–2971, 2011.

[40] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, “Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data,” *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1737–1754, 2008.

[41] S. Chen, D. Donoho, and M. Saunders, “Atomic decomposition by basis pursuit,” *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.

[42] L. Badea, “Extracting gene expression profiles common to colon and pancreatic adenocarcinoma using simultaneous nonnegative matrix factorization.” in *Proc. Pac. Symp. Biocomput.*, 2008, pp. 279–290.

[43] A. Cichocki, R. Zdunek, A. H. Phan, and S.-I. Amari, *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. Wiley, 2009.

[44] J. Bioucas-Dias and J. Nascimento, “Hyperspectral subspace identification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 8, pp. 2435–2445, 2008.

[45] F. Anscombe, “The transformation of Poisson, binomial and negative-binomial data,” *Biometrika*, vol. 35, no. 3/4, pp. 246–254, 1948.

[46] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Adv. Neural Inform. Process. Syst.* MIT Press, 2001, vol. 13, pp. 556–562.

[47] M. Chu, F. Diele, R. Plemmons, and S. Ragni, “Optimality, computation, and interpretation of nonnegative matrix factorizations,” *SIAM J. Matrix Anal.*, pp. 4–8030, 2004.

[48] C.-J. Lin, “Projected gradient methods for nonnegative matrix factorization,” *Neural Comput.*, vol. 19, no. 10, pp. 2756–2779, 2007.

[49] N. Gillis and F. Glineur, “Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization,” *Neural Comput.*, vol. 24, no. 4, pp. 1085–1105, 2012.

[50] A. Cichocki, R. Zdunek, and S.-i. Amari, "Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization," in *Independent Component Analysis and Signal Separation*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, vol. 4666, pp. 169–176.

[51] "Urban and Mixed Environment Sample: Reno, NV, USA (Reflectance+IGM)," http://www.spectir.com/free-data-samples/.

[52] D. Donoho and J. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[53] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proc. 12th IEEE Int. Conf. Comp. Vis.*, 2009, pp. 2272–2279.

[54] "AVIRIS image of Indian Pine Test Site," https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html.

[55] "Pavia University scene," http://www.ehu.es/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes.

[56] B. Rasti, J. Sveinsson, M. Ulfarsson, and J. Benediktsson, "Hyperspectral image denoising using first order spectral roughness penalty in wavelet domain," *IEEE J. Select. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2458–2467, 2014.

[57] P. Zhong and R. Wang, "Multiple-spectral-band CRFs for denoising junk bands of hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 4, pp. 2260–2275, 2013.

[58] ——, "Jointly learning the hybrid CRF and MLR model for simultaneous denoising and classification of hyperspectral imagery," *IEEE Trans. Neural Net. Learn. Syst.*, vol. 25, no. 7, pp. 1319–1334, 2014.

[59] Y. Tarabalka, M. Fauvel, J. Chanussot, and J. Benediktsson, "SVM- and MRF-based method for accurate classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 4, pp. 736–740, 2010.

[60] Y. Qian, M. Ye, and J. Zhou, "Hyperspectral image classification based on structured sparse logistic regression and three-dimensional wavelet texture features," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 4, pp. 2276–2291, 2013.