FastHyMix: Fast and Parameter-Free Hyperspectral Image Mixed Noise Removal

Lina Zhuang, Member, IEEE, and Michael K. Ng, Senior Member, IEEE

Abstract—The decrease in the widths of spectral bands in hyperspectral imaging leads to a decrease in signal-to-noise ratio (SNR) of measurements. The decreased SNR reduces the reliability of measured features or information extracted from hyperspectral images (HSIs). Furthermore, the image degradations linked with various mechanisms also result in different types of noise, such as Gaussian noise, impulse noise, deadlines, and stripes. This article introduces a fast and parameter-free hyperspectral image mixed noise removal method (termed FastHyMix), which characterizes the complex distribution of mixed noise by using a Gaussian mixture model and exploits two main characteristics of hyperspectral data, namely, low rankness in the spectral domain and high correlation in the spatial domain. The Gaussian mixture model enables us to make a good estimation of Gaussian noise intensity and the locations of sparse noise. The proposed method takes advantage of the low rankness using subspace representation and the spatial correlation of HSIs by adding a powerful deep image prior, which is extracted from a neural denoising network. An exhaustive array of experiments and comparisons with state-of-the-art denoisers was carried out. The experimental results show significant improvement in both synthetic and real datasets. A MATLAB demo of this work is available at https://github.com/LinaZhuang for the sake of reproducibility.

Index Terms—Gaussian mixture model, hyperspectral image (HSI) denoising, hyperspectral image (HSI) restoration, low-rank representation, plug-and-play, sparse representation.

NOMENCLATURE

\[ '\mathcal{X}' \in \mathbb{R}^{I_1 \times I_2 \times I_3} \]
3-D tensor (calligraphic letter).

\[ \mathbf{X} \]
Matrix (boldface capital letter).

\[ \mathbf{x} \]
Vector (boldface lowercase letter).

\[ \mathbf{x}_i \]
Scalar (italic lowercase letter).

\[ \text{mode-3} \]
Dimension \( n \) of a tensor.

\[ \text{mode-3 vectors of } \mathcal{X} \]
\( I_3 \)-dimensional vectors obtained from \( \mathcal{X} \) by varying the third index while keeping the first and second indices fixed.

\[ \mathcal{X}(\cdot, \cdot, i) \]
\( i \)th mode-3 slice of \( \mathcal{X} \), a matrix obtained by fixing the mode-3 index of \( \mathcal{X} \) to be \( i \).

\[ \mathbf{X}(3) \in \mathbb{R}^{I_1 \times (I_2 + I_3)} \]
Mode-3 unfolding of \( \mathcal{X} \). A tensor can be unfolded into a matrix by rearranging its mode-3 vectors, which are the column vectors of \( \mathbf{X}(3) \).

\[ \mathcal{X} \times_3 \mathbf{E} \]
Tensor matrix multiplication. The mode-3 product of a tensor \( \mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3} \) by a matrix \( \mathbf{E} \in \mathbb{R}^{I_3 \times I_2} \) is a tensor \( \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times I_3} \), denoted as \( \mathcal{Y} = \mathcal{X} \times_3 \mathbf{E} \), which is corresponding to a matrix multiplication, \( \mathbf{Y}(3) = \mathbf{E} \mathbf{X}(3) \).

\[ \| \mathcal{X} \|_F \]
Definition of Frobenius norm of a matrix is extended to a tensor as follows: \( \| \mathcal{X} \|_F = \sqrt{\sum_{i_1, i_2, i_3} |x(i_1, i_2, i_3)|^2} \).

I. INTRODUCTION

HYPERSPECTRAL cameras measure the radiation arriving at the sensor with high spectral resolution over a sufficiently broad spectral band such that the acquired spectrum can be used to uniquely characterize and identify any given material [1]. Hyperspectral imaging has been used in earth remote sensing tasks, such as object classification [2]–[5], detection of landcover change, and anomaly detection [6], [7], and plays an important role in a wide array of applications, such as astronomy, agriculture, and surveillance. However, due to the decrease in the widths of spectral bands, hyperspectral cameras receive fewer photons and tend to acquire images with a lower signal-to-noise ratio (SNR). The decreased SNR reduces the reliability of measured features or information extracted from hyperspectral images (HSIs) [1]. Therefore, HSI denoising is a fundamental preprocessing before further applications.

The image degradations linked with various mechanisms also result in different types of noise, such as Gaussian noise, Poissonian noise, impulse noise, deadlines/stripes, and cross-track illumination variation. In this article, we focus on the discussion of additive and signal-independent noise (namely, Gaussian noise, impulse noise, and deadlines/stripes) and attack hyperspectral mixed noise composed of these additive noises.

Hyperspectral mixed noise is usually removed by exploiting the distinct characteristics of HSIs and noise. Due to the very high spectral–spatial correlation, hyperspectral data are low-rank and sparse on transform domain representation (such as data-adaptive subspace [8], gradient domain [9], [10],...
Fourier domain [11], wavelet domain [12], and Cosine domain). Gaussian noise in hyperspectral data is independent and densely distributed in the original image space and the abovementioned transform domains. Consequently, a large amount of Gaussian noise can be removed from observations effectively either by using low-rank matrix/tensor factorization (in NMoG [13], double-factor regularized low-rank tensor factorization (LRTF-DFR) method [14], nonlocal meets global (NG-meet) method [15], and L1HyMixDe [16]), by minimizing the rank of the underlying clean HSI (in low-rank matrix recovery (LRMR) method [17], non-convex regularized low-rank and sparse matrix decomposition (NonRLRS) method [18], and DLR [19]), or by using sparse representation of the underlying HSI (in Kronecker-basis-representation (KBR) method [20], a spectral–spatial adaptive hyperspectral total variation (SSAHTV) method [21], the first-order roughness penalty (FORP) method [12], and the sparse representation and low-rank constraint (“Spa + Lr”) method [22]).

The tensor approximation technique has been drawing increasing attention from the HSI processing community, for example, a coupled sparse tensor factorization (CSTF)-based approach [23] is introduced for fusing hyperspectral and multispectral images, which has achieved outstanding performance. Compared with using low-rank matrix approximation, low-rank tensor representation of an HSI can finely preserve its intrinsic structure, exploit data correlation in three dimensions simultaneously instead of two dimensions, and lead to better performance of mixed noise removal. For example, Zheng et al. [24] introduce an HSI denoising method 3DLogTNN by modeling the underlying HSI as a tensor with a low-fibered rank. The nonconvex low-rank tensor approximation (NonLRTA) method [25] characterizes the clean HSI component by using the $\varepsilon$-norm, which is a nonconvex surrogate to Tucker rank. A tensor subspace representation (TenSR)-based HSI denoising method [26] takes advantage of the low-tubal rankness of the HSI tensor.

Impulse noise, deadlines, and stripe noise are usually sparse distributed in spatial and spectral domains; thus, they are modeled as an additive and sparse component in observation models. The sparsity of impulse noise, deadlines, and stripe noise in cost functions can be promoted by minimizing its $\ell_1$ norm in LRTF-DFR [14], by constraining the upper bound of the cardinality of the noise component in LRMR [17], and by introducing a nonconvex regularizer named as normalized $\varepsilon$-penalty in NonRLRS [18].

The structures of HSIs and noise have been studied to develop a number of hyperspectral denoising methods, which addresses Gaussian noise well, but are not robust to mixed noise. There are several reasons.

1. Both clean HSIs and stripes are low-rank [27]; thus, only a low-rank regularization imposed on HSIs is not sufficient to separate them (see more experimental results of LRMR [17] and NonRLRS [18] in Sections V and VI).

2. Noise in real HSIs exhibits very complex statistical distributions, calling for a powerful tool to characterize its structure.

3. Noise type in each HSI varies according to the imaging conditions, such as atmospheric environment or illumination.

Therefore, the type, intensity, and distribution of noise vary in each HSI, calling for robust and data-adaptive denoising methods. To address these hurdles, we propose an adaptive Gaussian mixture model to fit the distribution of mixed noise so that we can find the noise structure and estimate the noise intensity.

A. Related Work

HSIs can be well approximated by low-dimensional subspace representations and are characterized by a high level of self-similarity [1], [28]–[35]. The idea of regularizing the subspace representation coefficients of HSIs underlies state-of-the-art Gaussian-denoisers. We refer to representative work: GLF [36], FastHyDe [8], and NG-meet [15]. This article extends this strategy to address mixed noise. The challenge of this extension lies in the estimation of the spectral subspace.

In the scenario of mixed noise, a spectral subspace is estimated iteratively and jointly with subspace coefficients of the HSI, for example, in LRTF-DFR [14] and SNLRSF [37]. Joint estimation of the subspace and the corresponding coefficients of the HSI usually produce poor estimates of the subspace when HSI is affected by sever mixed noise. This article introduces a strategy to estimate the subspace and the corresponding coefficients of the HSI separately, leading to a noniterative and more effective method.

Coefficients of subspace representation of an HSI are termed Eigen-images, which have a very high spatial correlation. Therefore, remaining noise in the eigen-images can be further alleviated by image filtering. For example, single-band image denoisers, BM3D [38], WNNM [39], weighted TV are used in HSI denoising methods, FastHyDe [8], NG-meet [15], and LRTF-DFR [14], respectively. In this article, eigen-images are regularized with a more powerful deep image prior, as neural networks have shown impressive performance in recovering clean natural images from noisy observations [40]–[42]. Some neural networks have been conceived specially for addressing HSI noise. We refer to some representative networks, such as a spatial-spectral gradient network (SSGN) [43], a CNN-based HSI denoising method HSI-DeNet [44], a novel deep spatiotemporal Bayesian posterior (DSSBP) framework [45], and a 3-D dual-attention denoising network (3D-ADNet) [46].

As the performance of deep learning-based denoisers highly depends on the quality and quantity of training data, a challenge of deep-denoisers is the lack of real HSIs that can be used as training data or how to simulate pairs of clean/noisy images close to real HSIs. To sidestep the lack of training HSIs, this work takes advantage of the similarity between HSIs and RGB/gray-scale images. Both kinds of images are natural images, sharing common image structures; thus, it is reasonable that a denoising network, well-trained using vast amounts of RGB images, also works well for HSIs. This article incorporates a well-known single-band deep denoiser, FFDNet [41], into a mixed noise removal framework derived using traditional ML technique. This fall in the line of research, called plug-and-play technique [47]–[49] or regularization by denoising (RED) framework [50].

B. Contributions

The work aims to recover an underlying clean HSI from observations corrupted by additive mixed noise (containing Gaussian noise, impulse noise, and deadlines/stripes) by characterizing the statistical distribution of mixed noise using a Gaussian mixture model and exploiting the low rankness in the spectral domain and high correlation in the spatial domain of HSIs. Contributions of this work are summarized as follows.
1) A noise estimation method is proposed for mixed noise by exploiting the high spectral correlation of HSIs. The mixed noise is partitioned into Gaussian noise and sparse noise using a Gaussian mixture model fit to the mixed noise. The partition enables us to make a good estimation of Gaussian noise intensity per band and the locations of sparse noise.

2) The proposed mixed noise removal method is user-friendly and parameter-free by allowing parameters adaptive to specific images. That is the parameters can be set adaptively to noise statistics.

3) An image prior extracted from a state-of-the-art neural denoising network, FFDNet, is seamlessly embedded within our HSI mixed noise removal framework, which is a successful combination of traditional machine learning techniques and deep learning techniques. Experimental results demonstrate that the embedded deep image prior significantly improves the estimation accuracy of clean HSIs.

This article is organized as follows. Section II formulates the hyperspectral mixed noise removal problem. Section III describes a new noise estimation approach elaborated for mixed noise. Section IV formally introduces the proposed mixed noise removal method. Sections V and VI show and analyze the experimental results of the proposed method and the comparison methods. Finally, we make a conclusion of this article in Section VII.

II. PROBLEM FORMULATION

Some notations and tensor operations used in this article and their definitions are provided in Nomenclature. Let \( \mathcal{X} \in \mathbb{R}^{r \times c \times B} \) denote an underlying clean HSI with \( r \times c \) pixels and \( B \) bands. Assuming that noise is additive, we can write an observation model as

\[
\mathcal{Y} = \mathcal{X} + \mathcal{G}
\]

where \( \mathcal{Y}, \mathcal{G} \in \mathbb{R}^{r \times c \times B} \) denote an observed HSI data and mixed noise, respectively. Elements in \( \mathcal{G} \) are assumed to be a mixture of Gaussian noise, stripes, deadlines, and impulse noise. Gaussian noise in real HSI tends to be nonindependent and identically distributed \( \text{(i.i.d.)} \) that is pixelwise-independent but bandwise-dependent.

Due to the extreme high spectral correlation, hyperspectral vectors can be represented well in a low-dimensional subspace \([1]\), that is

\[
\mathcal{X} = Z \times_3 E
\]

with \( E \in \mathbb{R}^{B \times P} \) \((P \ll B)\) and \( Z \in \mathbb{R}^{r \times c \times P} \). \( E \) holds an orthogonal basis for the signal subspace, and the entries of \( Z \) are representation coefficients of \( \mathcal{X} \) with respect to \( E \). Hereafter, mode-3 slices of \( Z \) are termed the eigen-images.

HSI mixed noise removal aims to estimate an underlying clean image \( \mathcal{X} \), given an observed noisy HSI \( \mathcal{Y} \). This article introduces a Fast and parameter-free Hyperspectral image Mixed noise removal method (termed FastHyMix), which characterizes the complex distribution of mixed noise by using a Gaussian mixture model and exploits two main characteristics of hyperspectral data, namely, low rankness in the spectral domain and high correlation in the spatial domain. The main steps of the proposed method are described in the flowchart in Fig. 1. In the following, we first start by modeling the complex noise as a mixture of Gaussian densities, which is a universal approximation to any continuous distribution and, hence, capable of modeling a wide range of noise distributions \([51]\). Then, we take advantage of the noise statistics learned from the fit Gaussian mixture model and derive a mixed noise removal algorithm adaptive to the noise statistics of the image using subspace representation and deep image prior.

III. HYPERSPECTRAL MIXED NOISE ESTIMATION

In this section, we study the data structure of mixed noise by modeling the complex noise as a mixture of Gaussian densities, which enables us to make a good estimation of Gaussian noise intensity per band and the locations of
sparse noise. Fig. 2 depicts the procedure to estimate noise statistics in HSIs. It performs band by band and mainly contains the following steps.

1) It starts by estimating a coarse noise per band by using a linear regression method exploiting spectral correlation of HSIs.

2) A normality test is used to determine if a histogram of coarse noise per band is well-modeled by a Gaussian distribution.

3) If a histogram of coarse noise is approximately Gaussian, then we consider the noise in this band to be Gaussian noise.

4) If a histogram of coarse noise cannot be fit well by a Gaussian distribution, then we consider the noise in this band to be a mixture of Gaussian noise and sparse noise. The mixed noise is then fit by a Gaussian mixture model with two components. The fitting results allow us to partition the coarse noise into two sets (namely, Gaussian noise and sparse noise).

5) Estimate the standard deviation of Gaussian noise per band and the locations of sparse noise.

Details of each step are given in the following.

A. Estimation of Noise Statistics

Since spectral bands are highly correlated in a HSI, we assume that one band can be approximately represented as a linear combination of the remaining \((B - 1)\) bands [52], [53], that is,

\[
Y_{(3)}^T_{(i,b)} = Y_{(3)}^T_{(i)} \beta_b + \xi_b
\]  

(3)

where \(Y_{(3)}^T\) denotes the transpose of the mode-3 unfolding matrix \(Y_{(3)}\), the subscript \([\cdot]_{(i,b)}\) means extracting \(b\)th column from a matrix, a matrix with the subscript \([\cdot]_{(i)}\) means the matrix including all columns except \(b\)th column, \(\beta_b \in \mathbb{R}^{B-1}\) denotes regression coefficients, and \(\xi_b \in \mathbb{R}^I\) (with \(I = r \times c\)) denotes regression error.

The regression coefficients \(\beta_b\) can be estimated by the least-squares method, that is,

\[
\hat{\beta}_b = \arg \min_{\beta_b} \|Y_{(3)}^T_{(i,b)} - Y_{(3)}^T_{(i)} \beta_b\|_F^2
\]

\[
= \left(\left[\left(Y_{(3)}^T_{(i)} \right]_{(i,b)}\right)^{-1} \left[Y_{(3)}^T_{(i,b)} \right]_{(i,b)}\right]_{(i,b)}
\]

(4)

Given \(\hat{\beta}_b\), the estimate of regression error, \(\hat{\xi}_b\), is computed by

\[
\hat{\xi}_b = \left[Y_{(3)}^T_{(i,b)} \right]_{(i,b)} - Y_{(3)}^T_{(i,b)} \hat{\beta}_b
\]

(5)

We take the regression error, \(\hat{\xi}_b (b = 1, \ldots, B)\), as a coarse estimate of uncorrelated noise in the \(b\)th band. Statistical distribution of \(\hat{\xi}_b\) is studied band by band in the following.
Fig. 4. First row shows the 143rd band of the Hyperion Cuprite image and its noise statistics, and the second row shows the 36th band of the Tiangong-1 image and its noise statistics. (a) Observed 143th band Hyperion Cuprite data. (b) Regression errors, $\xi_{143}$. (c) Histogram of $\xi_{143}$ fit by a Gaussian distribution. (d) Gaussian mixture model fit to $\xi_{143}$. (e) Clustering labels-based GMM. (f) $\hat{M}(\cdot, \cdot, 143)$. (g) Observed 36th band Tiangong-1 data. (h) Regression errors, $\xi_{36}$. (i) Histogram of $\xi_{36}$ fit by a Gaussian distribution. (j) Gaussian mixture model fit to $\xi_{36}$. (k) Clustering labels-based GMM. (l) $\hat{M}(\cdot, \cdot, 36)$.

B. Standard Deviation Estimation of Gaussian Noise

If the histogram of the coarse noise in the $b$th band, $\hat{\xi}_b$, can be approximated by a Gaussian distribution, then we consider the coarse noise in this band to be Gaussian noise. Consequently, the standard deviation of Gaussian noise, denoted by $\hat{\sigma}_b$, is estimated by computing

$$\hat{\sigma}_b = \text{std}(\hat{\xi}_b)$$  \hspace{1cm} (6)

where std(·) is a function computing standard deviation of the elements of the input vector. For normality test, we use skewness and kurtosis estimates. That is, if absolute value of the skewness of $\hat{\xi}_b$ is less than 3 and the absolute value of its kurtosis is less than 10 [54], then we consider $\hat{\xi}_b$ to be approximately Gaussian distributed. Otherwise, $\hat{\xi}_b$ is a mixture of different kinds of noise.

Two bands from real HSIs and their coarse noise are shown in Fig. 3, where we can see that the histograms of coarse noise can be fit well by Gaussian distributions. Therefore, we consider these two bands to have only Gaussian noise.

C. Detection of Pixels Corrupted by Mixed Noise

If the histogram of the coarse noise $\hat{\xi}_b$ is not well-modeled by a Gaussian distribution, then it implies that the noise in the $b$th band is a mixture of Gaussian noise and sparse noise. For example, Fig. 4 shows two bands from two real HSIs and their corresponding coarse noise. The red curves in Fig. 4(c) and (i) are Gaussian distributions fit to the histograms of coarse noise. It is obvious that simple Gaussian distributions do not fit the coarse noise well and underfitting occurs. Then, we fit this mixed noise using a Gaussian mixture model, which is a universal approximation to any continuous distribution and, hence, capable of modeling a complex noise distribution.

To create a useful GMM, the choice of the number of components (i.e., the number of Gaussian distributions) should be careful. We can choose the best number of components using Akaike information criterion (AIC) [55] and Bayesian information criterion (BIC) [56]. The AIC or BIC for a model is usually written in the form $\left(-2 \log L + cK\right)$, where $L$ is the likelihood function, $K$ is the number of parameters in the model (i.e., the number of components in GMMs), and $c$ is 2 for AIC and $\log(I)$ for BIC. Both AIC and BIC are a way to find the balance between a good fit and over complexity in a model. Fig. 5 shows the AIC and BIC scores of GMMs with a various number of components fit to the coarse noise shown in Fig. 4(b). The model with two components is the best one that balances low AIC and BIC with simplicity. Also, AIC and BIC have been tested in other images in the experimental part, and it comes to a conclusion that the GMMs with two components are good enough to fit the data. Therefore, the number of components in GMM is set to 2 in our work. Meanwhile, for HSI mixed noise removal problem,
we mainly interest in the statistical parameters of Gaussian noise and the spatial locations of sparse noise. A GMM with two components can exactly group elements in $\xi_b$ into two sets: Gaussian noise and sparse noise.

Let $a_i = \{a_{1b}, \ldots, a_{ib}, \ldots, a_{t_b}\}^T := \xi_b \in \mathbb{R}^l$ denote the coarse noise in the $b$th band. To simplify the notation in the following derivation, we omit its band index in Section III-C. Then, we have $a = [a_1, \ldots, a_i, \ldots, a_T]^T := \xi_b$.

1) Gaussian Mixture Model Fit to Mixed Noise: A Gaussian mixture distribution model with two components fit to the mixed noise takes the form

$$f(a_i; \psi) = \sum_{k=1}^{2} \pi_k \mathcal{N}(a_i; \mu_k, \sigma_k^2)$$  \hfill (7)$$

where $\pi_k$ denotes nonnegative and sum-to-one mixing proportions, $\mathcal{N}(a_i; \mu_k, \sigma_k^2)$ are the component Gaussian densities with mean value $\mu_k$ and variance $\sigma_k^2$ computed at $a_i$, and $\psi = (\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \pi_1, \pi_2)^T$ denotes a vector of unknown parameters.

The maximum likelihood estimate of $\psi$ is written as

$$\hat{\psi} = \arg \max_{\psi} \log L(\psi) = \arg \max_{\psi} \log \left( \prod_{i=1}^{l} \sum_{k=1}^{2} \pi_k \mathcal{N}(a_i; \mu_k, \sigma_k^2) \right).$$  \hfill (8)$$

The above maximum likelihood problem can be solved by implementing expectation-maximization (EM) algorithm [57], which starts by conceptualizing $a_i$ to have arisen from one of the component distributions of the mixture model. We define an indicator, $u_{ik} = 1$ if $a_i$ has arisen from the $k$th component distribution, and $u_{ik} = 0$ otherwise. Let $\psi^{(t)} := \{\rho_1^{(t)}, \rho_2^{(t)}, \sigma_1^{(t)}, \sigma_2^{(t)}, \pi_1^{(t)}, \pi_2^{(t)}\}$ denote the parameters of mixture distributions on the $t$th iteration of the EM algorithm. The EM algorithm seeks to find the maximum likelihood estimate of mixture distributions by iteratively applying the following two steps (see [57] for details).

a) E-step: Given $\psi^{(t)}$, E-step on the $(t+1)$th iteration calculates the current conditional expectation of $U_{ik}$ given data $a_i$, where $U_{ik}$ is the random variable corresponding to $u_{ik}$. We have

$$E(U_{ik} | a_i; \psi^{(t)}) = \tau_k(a_i; \psi^{(t)})$$ \hfill (9)$$

where

$$\tau_k(a_i; \psi^{(t)}) = \frac{\pi_k^{(t)} \mathcal{N}(a_i; \mu_k^{(t)}, \sigma_k^{(2,t)})}{f(a_i; \psi^{(t)})}. \hfill (10)$$

After taking the conditional expectation with $\psi = \psi^{(t)}$, we have

$$Q(\psi; \psi^{(t)}) = \sum_{k=1}^{2} \sum_{i=1}^{l} \tau_k(a_i; \psi^{(t)}) \{ \log \pi_k + \log \mathcal{N}(a_i; \mu_k, \sigma_k^2) \}. \hfill (11)$$

b) M step: The M-step on the $(t+1)$th iteration updates $\psi$ by calculating the global maximization of $Q(\psi; \psi^{(t)})$ with respect to $\pi_k$, $\mu_k$, $\sigma_k$ $(k = 1, 2)$, yielding

$$\pi_k^{(t+1)} = \frac{\sum_{i=1}^{l} \tau_k(a_i; \psi^{(t)})}{n}, \quad (k = 1, 2) \hfill (12)$$

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^{l} \tau_k(a_i; \psi^{(t)}) a_i}{\sum_{i=1}^{l} \tau_k(a_i; \psi^{(t)})}, \quad (k = 1, 2) \hfill (13)$$

and

$$\sigma_k^{2(t+1)} = \frac{\sum_{i=1}^{l} \tau_k(a_i; \psi^{(t)}) (a_i - \mu_k^{(t+1)})^2}{\sum_{i=1}^{l} \tau_k(a_i; \psi^{(t)})}, \quad (k = 1, 2). \hfill (14)$$

2) Cluster Using Gaussian Mixture Model: Given $\hat{\psi}$, the posterior probability that the element $a_i$ arose from group $k$ is given by the fit posterior probability

$$\tau_k(a_i; \hat{\psi}) = \frac{\hat{\pi}_k \mathcal{N}(a_i; \hat{\mu}_k, \hat{\sigma}_k^2)}{f(a_i; \hat{\psi})} \quad (k = 1, 2; i = 1, \ldots, I). \hfill (15)$$

The coarse noise elements $(a_1, \ldots, a_T)$ can be partitioned into two sets $S = \{S_G, S_S\}$ (where $S_G$ and $S_S$ represent a set of Gaussian noise and a set of sparse noise, respectively) by assigning each $a_i$ to the group to which it has the highest estimated posterior probability of belonging. Let $L_i \in \{1, 2\}$ denote the cluster label of $a_i$. We have

$$L_i = \left\{ \begin{array}{ll} 1 & \text{if } \tau_k(a_i; \hat{\psi}) \geq \tau_{1-k}(a_i; \hat{\psi}) \\
2 & \text{otherwise} \end{array} \right\}$$ \hfill (16)$$

As the Gaussian noise is densely distributed and other kinds of noise (namely, stripes, deadlines, and impulse noise) are usually sparsely distributed in the spatial domain, the group with a larger component proportion is considered to be Gaussian noise, and the other group is sparse noise [see Fig. 4(e) and (k)].

D. Estimation of $\mathbf{C}$ and $\mathbf{M}$

After clustering the coarse noise using the Gaussian mixture model, we can derive the following noise statistics that are of importance for conceiving a parameter-free denoising algorithm.

1) The standard deviation of Gaussian noise in the $b$th band can be computed using the set of Gaussian noise via $\hat{\sigma}_b = \text{std}(S_G)$, where $S_G$ represents a set of coarse noise elements in the $b$th band belonging to sparse noise. Then, an estimate of the covariance matrix of Gaussian noise is obtained as $\hat{C} = \text{diag}(\hat{\sigma}_1^2, \ldots, \hat{\sigma}_b^2, \ldots, \hat{\sigma}_B^2) \in \mathbb{R}^{B \times B}$.

2) Let $\mathcal{M} \in \{0, 1\}^{I \times C \times B}$ denote a mask tensor indicating noise types as follows:

$$m_{i,j,b} = \begin{cases} 1, & \text{if observation } y_{i,j,b} \text{ is corrupted only by Gaussian noise} \\ 0, & \text{if } y_{i,j,b} \text{ is corrupted by mixed noise} \end{cases}$$

We can derive an estimate of the mask, $\hat{\mathcal{M}}$, from the clustering result of each element in the coarse noise [see Fig. 4(f) and (l)].

IV. FASTHyMix: Fast HSI Mixed Noise Removal

Given the estimates of the correlation matrix of Gaussian noise and the locations of sparse noise, we introduce a fast HSI mixed noise removal method, preceded by a noise-whitening step, which is based on the estimated correlation matrix of Gaussian noise, $\hat{C}$. 

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
A. Gaussian Noise-Whitening Transformation

Gaussian noise in real HSIs usually has different intensity per band. To remove Gaussian noise effects on spectral subspace learning, we include a noise-whitening process. Let \( \bar{Y} \in \mathbb{R}^{T \times C \times B} \) denote the noise-whitened image, and we have

\[
\bar{Y} = Y \times 3 \sqrt{C^{-1}} = \bar{X} + \bar{\Omega}
\]

where \( \bar{X} = X \times 3 \sqrt{C^{-1}} \) and \( \bar{\Omega} = \mathcal{O} \times 3 \sqrt{C^{-1}} \). After whitening, mode-3 vectors of \( \bar{X} \) still live in a low-dimensional subspace; thus,

\[
\hat{Y} = \bar{Z} \times 3 \bar{E} + \bar{\Omega}
\]

where columns of \( \bar{E} \in \mathbb{R}^{B \times P} \) span an orthogonal subspace, which can be estimated from the observations corrupted only by Gaussian noise. We take observed pixels corrupted only by Gaussian noise from \( \bar{Y} \) and stack them as columns to make a single matrix, denoted as \( \hat{Y} \in \mathbb{R}^{B \times Q} \) with \( Q \) pixels. An estimate of the spectral subspace can be obtained as

\[
\hat{E} = \hat{U}(1:1:P)
\]

where \( P \) denotes the dimension of spectral subspace and \( \hat{U} \in \mathbb{R}^{B \times B} \) is an orthogonal matrix and \( [U, \Sigma, V] = \text{SVD}(\hat{Y}) \) with singular values in \( \Sigma \) ordered by nonincreasing magnitude. In the scenario of mixed noise, the estimation of the dimension of the signal subspace is challenging [1], [52]. Fortunately, our proposed method is extremely robust to errors in the estimation of the subspace dimension, as far as the subspace dimension is not underestimated. Relative evidence and analysis are provided in Section V-E.

B. Estimation of Eigen-Images

Given the estimate of the spectral subspace, the subspace coefficients of the HSI can be estimated by solving the following optimization problem:

\[
\hat{Z} = \arg \min_{\mathcal{Z}} \frac{1}{2} \| \hat{M} \odot (\bar{Y} - \bar{Z} \times 3 \bar{E}) \|_F^2 + \lambda \phi_{\text{deep}}(\hat{Z})
\]

where \( \odot \) denotes elementwise multiplication and \( \lambda > 0 \) is a parameter of the regularization. The first term on the right-hand side represents the data fidelity and accounts only for the Gaussian noise, and \( \phi_{\text{deep}}(\cdot) \) is a regularizer expressing prior information tailored to spatially correlated eigen-images.

Solvers for optimization problem (20) will be iterative due to a nondiagonal operator involved in (20). To sidestep iterations and speed up the algorithm, we propose a suboptimal solution that is very fast and effective. Problem (20) is solved approximately by the following two steps.

In the first step, we estimate the components corrupted by sparse noise at each pixel, \( \bar{Y}_i \in \mathbb{R}^B \) \((i = 1, \ldots, I)\), by solving a simple least-squares problem

\[
\hat{z}_i = \arg \min_{\hat{z}_i} \| \hat{m}_i \odot (\bar{y}_i - \hat{E}\hat{z}_i) \|_2
\]

\[
= (\hat{E}^T \text{diag}(\hat{m}_i) \hat{E})^{-1} \hat{E}^T (\hat{m}_i \odot \bar{y}_i), \quad (i = 1, \ldots, I)
\]

where \( \bar{y}_i \in \mathbb{R}^B, \hat{m}_i \in \mathbb{R}^B, \) and \( \hat{z}_i \in \mathbb{R}^P \) are \( i \)th pixel in the noise-whiten image, corresponding mask, and subspace coefficients, respectively. The matrix \( (\hat{E}^T \text{diag}(\hat{m}_i) \hat{E}) \) is nonsingular when \( \| m_i \|_0 \geq P \), meaning that the number of components corrupted by only Gaussian noise is larger or equal than the dimension of the signal subspace. Given \( \hat{z}_i \), we can recover the components corrupted by sparse noise at each pixel by computing

\[
\hat{y}_i = \hat{E} \hat{z}_i, \quad (i = 1, \ldots, I).
\]

Now, remaining noise in \( \hat{y}_i \) \((i = 1, \ldots, I)\) is mainly Gaussian; thus, in the second step, we perform a Gaussian noise removal step on \( \hat{Z} := \text{folding}((\hat{y}_1, \ldots, \hat{y}_B)) \in \mathbb{R}^{T \times C \times B} \)

\[
\hat{Z} = \arg \min_{\hat{Z}} \frac{1}{2} \| \bar{Y} - \bar{Z} \times 3 \bar{E} \|_F^2 + \lambda \phi_{\text{deep}}(\hat{Z})
\]

\[
= \arg \min_{\hat{Z}} \frac{1}{2} \| \bar{Y} - \bar{Z} \times 3 \bar{E} \|_F^2 + \lambda \phi_{\text{deep}}(\hat{Z})
\]

which is a proximity operator of \( \phi_{\text{deep}} \) applied to \( \hat{Z} \). A proof of the equivalence of (23) and (24) can be found in Appendix A of [36]. Considering that the orthogonal projection is a decorrelation transformation and mode-3 slices of \( \bar{Z} \) tend to be decorrelated, we decouple \( \phi_{\text{deep}}(\cdot) \) with respect to the mode-3 slices, that is,

\[
\phi_{\text{deep}}(\hat{Z}) = \sum_{p=1}^P \phi_{\text{deep},p}(\hat{Z}(\cdot; p))
\]

where \( \hat{Z}(\cdot; p) \) denotes the \( p \)th mode-3 slice. The solution of (24) is decoupled with respect to \( \hat{Z}(\cdot; p) \) and may be written as

\[
\hat{Z}(\cdot; p) = \arg \min_{\hat{Z}(\cdot; p)} \frac{1}{2} \| \hat{Z}(\cdot; p) - \bar{Z}(\cdot; p) \|_F^2 + \lambda \phi_{\text{deep},p}(\hat{Z}(\cdot; p)), \quad p = 1, \ldots, P.
\]

To solve subproblem (26), we resort to the plug-and-play trick [47], [58]–[60], whose main idea is to directly use an existing regularizer from a state-of-the-art denoiser, instead of investing effort in designing a new regularizer exploiting the high spatial correlation of eigen-images. In this article, the prior of a denoising network, FFDNet\(^1\) [41], is plugged in (26), leading to

\[
\hat{Z}(\cdot; p) = \text{FFDNet}(\hat{Z}(\cdot; p), \lambda)
\]

where the function FFDNet(\cdot) outputs a denoised eigen-image. In the FFDNet network, parameter \( \lambda \) is related to the standard deviation of Gaussian noise in the input image. As Gaussian noise in \( \hat{Y} \) has been whitened, we set \( \lambda = 1 \) in (27). We remark that other state-of-the-art single-band Gaussian-denoisers (such as BM3D [38] and WNNM [39]) also can be adapted to estimate the eigen-images. The FFDNet is selected in this work due to its following advantages over others. Compared with other machine learning-based denoisers [38], [39], deep-learning-based FFDNet is much faster as long as it has been well trained. Compared with other deep-learning-based denoisers [44], [61], FFDNet is able to address images with various noise levels, meaning that we can input a new image without retraining.

The clean data of \( \hat{Y} \) are recovered as

\[
\hat{X} = \hat{Z} \times 3 \hat{E}.
\]

\(^1\)https://github.com/cszn/FFDNet
C. Inverse Noise-Whitening Transformation

Finally, we perform inverse noise-whitening transformation to obtain an estimate of the clean HSI

\[ \hat{X} = \sqrt{\hat{\mathbf{C}}} \hat{Z}. \]  

(29)

The pseudocode in Algorithm 1 shows how FastHyMix is implemented to reduce mixed noise for an HSI. Given an HSI of size \(r \times c \times B\) (bands) with subspace dimension \(P \ll B\), the computational complexity of obtaining \(\hat{\mathbf{C}}\) and \(\hat{\mathbf{M}}\) in line 1 is \(O(r*c*B^3)\) and \(O(r*c*B)\), respectively. The Gaussian noise-whitening in line 2, its inverse transformation in line 6, and the image reconstruction step in line 5 have same computational complexity, that is, \(O(r*c*B^2)\). The estimation of the spectral subspace in line 3 and eigen-images in line 4, respectively, costs \(O(r^2*c^2*B)\) and \(O(r*c*B^2*P + P*d)\), where \(d\) represents the computational complexity of denoising an eigenimage. Consequently, the overall computational complexity of FastHyMix is \(O(r*c*B^3 + r^2*c^2*B + P*d)\).

**Algorithm 1 FastHyMix: Fast Hyperspectral Image Mixed Noise Removal**

**Input:** A noisy HSIb \(Y\)
1: Estimation of noise statistics: \(\hat{\mathbf{C}}\) and \(\hat{\mathbf{M}}\)
2: Gaussian noise-whitening: \(\tilde{Y} = \sqrt{\hat{\mathbf{C}}} \hat{Z}\)
3: Estimation of the spectral subspace, \(\hat{\mathbf{E}}\), via \(19\)
4: Estimation of eigen-images, \(\tilde{Z}\), via \(21, 22,\) and \(27\)
5: \(\hat{X} = \sqrt{\hat{\mathbf{C}}} \hat{Z}\)
6: Inverse noise-whitening: \(\tilde{X} = \sqrt{\hat{\mathbf{C}}} \tilde{\mathbf{Z}}\)

**Output:** The denoised HSI \(\tilde{X}\)

V. EXPERIMENTS WITH SIMULATED IMAGES

The experiments of hyperspectral mixed noise removal were conducted on two simulated hyperspectral datasets and two real hyperspectral datasets (see Fig. 6).

A. Simulation of Noisy Datasets and Comparisons

Two noisy HSIs were generated based on two public hyperspectral datasets [as shown in Fig. 6(a) and (b)], namely, a subregion of Washington DC Mall data \(^2\) [of size 150(rows) \times 200(columns) \times 191(bands)] and a subregion of Pavia University data \(^3\) [of size 310(rows) \times 250(columns) \times 87(bands)]. First, we removed bands severely corrupted by water vapor in the atmosphere. To obtain relatively clean images, we projected spectral vectors of each image onto a subspace spanned by principal eigenvectors of each image. The projection of each image is considered to be a clean image.

To simulate noisy HSIs, we added four kinds of additive noise into images as follows.

**Case 1 (Gaussian Non-i.i.d. Noise):** \(\mathbf{n} \sim \mathcal{N}(0, \mathbf{D}^2)\) where \(\mathbf{D}\) is a diagonal matrix with diagonal elements sampled from a uniform distribution \(U(0.01, 0.02)\) in Washington DC Mall data and a uniform distribution \(U(0.05, 0.10)\) in Pavia University data.

**Case 2 (Gaussian Noise + Stripes):** Synthetic data with Gaussian noise (described in case 1) and oblique stripe noise randomly affecting 30% of the bands and, for each band, about random 10% of the pixels.

**Case 3 (Gaussian Noise + “Salt & Pepper” Noise):** Synthetic data with Gaussian noise, described in case 1 and “Salt & Pepper” noise with noise density 0.5%, meaning affecting approximately 0.5% of elements in \(X\).

**Case 4 (Gaussian Noise + Stripes + “Salt & Pepper” Noise):** Synthetic data with Gaussian noise, described in case 1, random oblique stripes, described in case 2, and “Salt & Pepper” noise, described in case 3.

Six recent state-of-the-art HSI denoising methods are taken for experimental comparison. They are NG-meet \(^4\), LIRM [17], \(^5\) NonRLRS [18], KBR [20], \(^6\) NMoG [13], and LRTF-DFR method [14]. \(^7\) Among them, the NG-meet is conceived to address hyperspectral Gaussian noise, but we included it to see whether mixed noise could be removed well by a Gaussian-denoiser. All experiments were implemented in MATLAB (R2020a) on Windows 10 with an AMD Ryzen 9 4900HS 3.00-GHz processor and 16-GB RAM.

Regarding the parameter setting of compared methods, we basically fine-tuned the parameters of regularizations for all simulated and real images. Also, for a fair comparison, we set the same values for common parameters, such as the dimension of spectral subspace used in NG-meet, LRTF-DFR, and FastHyMix.

For quantitative assessment, the peak signal-to-noise ratio (PSNR) index, the structural similarity (SSIM) index, the feature similarity (FSIM) index of each band, and the spectral angle distance (SAD) were calculated. The SSIM metric [62], measuring the similarity between two images, is considered to be correlated with the quality perception of the human visual system (HVS). The SSIM is designed by modeling any image distortion as a combination of three factors that are loss of correlation, luminance distortion, and contrast distortion. The FSIM metric [63] measures the similarity of images using gradient magnitude and Fourier phase congruency. The mean PSNR (MPSNR), mean SSIM (MSSIM), mean FSIM (MFSIM), and mean SAD (MSAD) over bands of denoised images are presented in Table I, where we highlighted the best results in bold.

B. Mixed Noise Removal

The mixed noise removal performance of each method on two simulated datasets in terms of MPSNR, MSSIM, MFSIM, and MSAD is presented in Table I. The bandwise PSNR is

\(^2\)https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html
\(^3\)http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes
\(^4\)https://github.com/quanmingyao/NGmeet
\(^5\)https://sites.google.com/site/teshewei/home
\(^6\)http://gr.xju.edu.cn/web/dymeng/
\(^7\)https://yubangzheng.github.io/homepage/#publications


TABLE I

PERFORMANCE OF THE PROPOSED AND COMPARISON METHODS ON WASHINGTON DC MALL DATA AND PAVIA UNIVERSITY DATA

| Indexes | Noisy | NG-meet [15] | LMRMR [17] | NonRLRSP [18] | KBR [20] | NMoG [13] | LRTF-DFR (Proposed) |
|---------|-------|--------------|------------|---------------|----------|-----------|---------------------|
|         | MPSNR (dB) | MSSIM | MSIM | MFSIM | MSAD | Time (s) | MPSNR (dB) | MSSIM | MSIM | MFSIM | MSAD | Time (s) | MPSNR (dB) | MSSIM | MSIM | MFSIM | MSAD | Time (s) | MPSNR (dB) | MSSIM | MSIM | MFSIM | MSAD | Time (s) |
| Case 1  | 19.33 | 37.48 | 34.13 | 31.74 | 33.84 | 32.36 | 39.73 | 40.20 | 0.8586 | 0.9973 | 0.9941 | 0.9848 | 0.9995 | 0.9907 | 0.9981 | 0.9982 |
|         | 0.9017 | 0.9937 | 0.989 | 0.9833 | 0.9905 | 0.9892 | 0.998 | 0.998 |
|         | 0.124 | 0.016 | 0.022 | 0.027 | 0.024 | 0.036 | 0.014 | 0.014 | 0 | 28 | 21 | 19 | 115 | 53 | 65 | 2 |
| Case 2  | 12.62 | 12.79 | 26.42 | 24.63 | 31.24 | 32.34 | 35.58 | 37.9 | 0.7589 | 0.6356 | 0.974 | 0.9643 | 0.9894 | 0.9907 | 0.9922 | 0.997 |
|         | 0.8351 | 0.8462 | 0.9733 | 0.9592 | 0.9847 | 0.9892 | 0.9939 | 0.9959 |
|         | 0.584 | 0.348 | 0.071 | 0.132 | 0.030 | 0.036 | 0.033 | 0.018 | 0 | 30 | 19 | 39 | 117 | 53 | 61 | 6 |
| Case 3  | 13.61 | 31.08 | 34.12 | 32.07 | 31.25 | 32.19 | 35.64 | 38.97 | 0.8248 | 0.9885 | 0.9941 | 0.989 | 0.9899 | 0.9906 | 0.9936 | 0.9978 |
|         | 0.8877 | 0.9833 | 0.9891 | 0.9838 | 0.9847 | 0.9892 | 0.9939 | 0.9975 |
|         | 0.177 | 0.028 | 0.022 | 0.026 | 0.030 | 0.036 | 0.024 | 0.015 | 0 | 31 | 20 | 52 | 114 | 68 | 68 | 68 |
| Case 4  | 8.46 | 13.59 | 26.51 | 24.58 | 33.02 | 32.1 | 35.53 | 37.22 | 0.7291 | 0.682 | 0.976 | 0.9642 | 0.9936 | 0.9906 | 0.9922 | 0.9966 |
|         | 0.8236 | 0.8501 | 0.9729 | 0.9593 | 0.9897 | 0.9891 | 0.9938 | 0.9949 |
|         | 0.602 | 0.326 | 0.070 | 0.130 | 0.025 | 0.036 | 0.033 | 0.018 | 0 | 30 | 20 | 40 | 118 | 52 | 78 | 7 |
|         | 0 | 70 | 30 | 22 | 124 | 76 | 104 | 3 |
|         | 18.28 | 28.4 | 32.73 | 30.15 | 33.36 | 35.41 | 32.71 | 38.54 | 0.3259 | 0.7799 | 0.9169 | 0.8964 | 0.9164 | 0.9317 | 0.8443 | 0.9742 |
|         | 0.6386 | 0.8926 | 0.9571 | 0.9412 | 0.9599 | 0.9732 | 0.9324 | 0.9840 |
|         | 0.668 | 0.329 | 0.116 | 0.146 | 0.11 | 0.087 | 0.189 | 0.055 | 0 | 72 | 27 | 32 | 129 | 73 | 129 | 6 |
|         | 0 | 73 | 30 | 22 | 128 | 78 | 126 | 12 |
|         | 18.61 | 33.3 | 34.23 | 30.67 | 33.48 | 34.24 | 32.86 | 38.63 | 0.3124 | 0.866 | 0.9218 | 0.9017 | 0.9157 | 0.9275 | 0.8461 | 0.9746 |
|         | 0.6432 | 0.9399 | 0.9644 | 0.9468 | 0.9602 | 0.9681 | 0.9323 | 0.9843 |
|         | 0.602 | 0.187 | 0.099 | 0.120 | 0.11 | 0.099 | 0.192 | 0.055 | 0 | 73 | 30 | 22 | 128 | 78 | 126 | 12 |
|         | 16.79 | 28.21 | 32.78 | 29.9 | 33.33 | 35.33 | 32.75 | 38.28 | 0.2914 | 0.7889 | 0.9165 | 0.8921 | 0.9152 | 0.9313 | 0.8456 | 0.9756 |
|         | 0.6175 | 0.8875 | 0.9571 | 0.9388 | 0.9597 | 0.973 | 0.9328 | 0.9836 |
|         | 0.709 | 0.336 | 0.115 | 0.145 | 0.112 | 0.088 | 0.192 | 0.057 | 0 | 72 | 30 | 34 | 129 | 78 | 108 | 12 |

For visual comparison, we display the 126th band of Washington DC Mall data and 37th band of Pavia University data in Figs. 8 and 9, respectively. For case 1 (Gaussian noise), all methods can reduce noise significantly. As shown in Figs. 8 and 9, LMRMR, NonRLRSP, and LRTF-DFR are able to remove light stripes but still leaving some wide stripes. Heavy stripe noise still remains in the results of NG-meet. However, we emphasize that it is unfair to compare with NG-meet in the cases, including mixed noise, as NG-meet is designed especially for Gaussian noise. KBR, NMoG, and FastHyMix methods visually yield comparable results in Figs. 8 and 9.

C. Estimation of Mixed Noise

One of the contributions in this article is to introduce a noise estimation method elaborated for mixed noise by exploiting the high spectral correlation of HSIs. The mixed noise estimation is based on the high correlation between spectral channels of HSIs, which leads to the low-rank structure of the HSIs in the spectral domain. This property is exploited by constraining the upper bound of the rank in LRMR, by introducing a nonconvex normalized ε-penalty to the matrix rank in NonRLRSP, by relaxing the tensor rank term with a log-sum form in KBR, and by low-rank matrix/tensor factorization in NMoG, LRTF-DFR, and FastHyMix. Results in Table I show that low-rank matrix/tensor factorization-based denoisers achieve better denoising performance.

depicted in Fig. 7 for quantitative assessment. It can be seen that FastHyMix uniformly yields the best performance in the shortest time in HSIs with different kinds of noise. Among the competitors, NG-meet is conceived specially for addressing Gaussian noise; thus, it works well in case 1 (including only Gaussian noise) but not in cases 2–4 (including mixed noise). The results of NG-meet in four cases imply that a mixture of noise cannot be addressed simply using a Gaussian-denoiser and, thus, call for efficient mixed noise removal methods. A high correlation between spectral channels of HSIs leads to the low-rank structure of the HSIs in the spectral domain, which is exploited by constraining the upper bound of the rank in LRMR, by introducing a nonconvex normalized ε-penalty to the matrix rank in NonRLRSP, by relaxing the tensor rank term with a log-sum form in KBR, and by low-rank matrix/tensor factorization in NMoG, LRTF-DFR, and FastHyMix. Results in Table I show that low-rank matrix/tensor factorization-based denoisers achieve better denoising performance.
Fig. 7. Bandwise PSNR values of denoised Washington DC Mall data in the first row and of denoised Pavia University data in the second row. Subfigures in (a) and (e), (b) and (f), (c) and (g), and (d) and (h) correspond to case 1, case 2, case 3, and case 4, respectively.

Fig. 8. Band 126 of the simulated Washington DC Mall data before and after denoising in four cases.

Fig. 9. Band 37 of the simulated Pavia University data before and after denoising in four cases.

noise is fit by a Gaussian mixture model with two components. The fitting enables us to make a good estimation of Gaussian noise intensity per band. We conducted experiments using a simulated Pavia University dataset to compare the proposed noise estimation method with a typical noise estimation method,
Fig. 10. Estimates of standard deviations of Gaussian noise per band in simulated Pavia University dataset. (a) Case 1. (b) Case 2. (c) Case 3. (d) Case 4.

Fig. 11. True masks (in the first row) and its estimates (in the second row) yielded by FastHyMix in the band 37 of simulated Pavia University dataset.

HySime [52], which has been used widely in state-of-the-art denoisers [18], [64]. A comparison of estimated standard deviations, \( \hat{\sigma}_i \), of Gaussian noise per band is presented in Fig. 10, where we can see both methods yield a good estimate of \( \hat{\sigma}_i \) in case 1, where the image is corrupted only by Gaussian noise. However, in cases 2–4 with mixed noise, compared with the HySime, the estimates obtained by the proposed method are much closer to the true ones. Our method can provide a better estimation of Gaussian noise intensity under the circumstance of mixed noise.

The Gaussian mixture model fit to the mixed noise also helps to identify the locations of sparse noise, represented by the mask, \( \mathcal{M} \). Fig. 11 displays the mask estimated by FastHyMix in band 37 of the Pavia University dataset. We can see FastHyMix can accurately identify pixels corrupted by sparse noise in all cases.

D. Deep Prior for Eigen-Images

One may question whether the deep prior from the deep network, FFDNet, which has been trained using gray-scale images acquired from commercial cameras, is suitable for remote sensing images. To see the impact of deep prior embedded in FastHyMix, a comparison experiment using simulated datasets was conducted. Fig. 12 gives MPSNR values of images denoised by original FastHyMix (with deep image prior) and modified FastHyMix (without deep image prior). MPSNR value per case is increasing considerably when eigen-images are filtered by a powerful network, FFDNet. It demonstrates that, given a good estimate of Gaussian noise level, a deep image prior makes a positive contribution to HSI denoising. Although the network was trained using gray-scale images, but not remote sensing images, the network still achieves impressive performance for HSI denoising. The reason is that both kinds of images are natural images, sharing the same properties, such as local and nonlocal similarity and piecewise smoothness. Therefore, the image prior learned by the network from gray-scale images is also applicable to HSIs.

E. Robustness of FastHyMix to Subspace Overestimation

The subspace dimension of two datasets input to compared methods, NG-meet, LRTF-DFR, and FastHyMix, was set to 8, the true value, for a fair comparison. In fact, it is challenging to estimate the subspace dimension with high accuracy from observed HSIs corrupted by mixed noise. Fortunately, FastHyMix is robust to subspace overestimation. We take two datasets, namely, the Washington DC Mall data and Pavia University data, to show their robustness. Fig. 13 shows the MPSNR yielded by FastHyMix as a function of the dimension of the subspace estimation. It is clear that the MPSNR is nearly constant, provided that the subspace dimension is not underestimated. To give an insight into the robustness of FastHyMix with respect to subspace overestimation, we analyze the impact of subspace overestimation on the processing of image signal, Gaussian noise, and sparse noise. In the objective function of FastHyMix, (20), as the dimension of subspace increases, the new subspace still can represent the image well but includes more amount of Gaussian noise.
TABLE II
SERIES OF MIXED NOISE WITH DIFFERENT INTENSITIES ADDED IN PAVIA UNIVERSITY DATA

| Case   | Gaussian noise distribution of standard deviation of Gaussian noise over bands | Stripe noise proportion of bands affected by stripe noise | Stripe value | ‘Salt & Pepper’ noise proportion of elements in $\mathcal{X}$ affected by ‘Salt & Pepper’ |
|--------|--------------------------------------------------------------------------------|--------------------------------------------------------|--------------|--------------------------------------------------------------------------|
| Case 5 | $U(0, 0.01)$                                                                  | 30%                                                    | maximum-valued stripes | ×                                                                            |
| Case 6 | $U(0, 0.02)$                                                                  | 30%                                                    | maximum-valued stripes | ×                                                                            |
| Case 7 | $U(0.01, 0.06)$                                                               | 50%                                                    | maximum-valued stripes | ×                                                                            |
| Case 8 | $U(0.05, 0.10)$                                                               | 70%                                                    | maximum-valued stripes | ×                                                                            |
| Case 9 | $U(0.01, 0.06)$                                                               | 5%                                                     | maximum-valued stripes | 0.01%                                                                      |
| Case 10| $U(0.01, 0.06)$                                                               | 30%                                                    | maximum-valued stripes | 0.05%                                                                      |
| Case 11| $U(0.01, 0.06)$                                                               | 50%                                                    | maximum-valued stripes | 0.10%                                                                      |
| Case 12| $U(0.01, 0.06)$                                                               | 70%                                                    | maximum-valued stripes | 0.50%                                                                      |
| Case 13| $U(0.05, 0.10)$                                                               | 30%                                                    | minimum-valued stripes | ×                                                                            |
| Case 14| $U(0.05, 0.10)$                                                               | 50%                                                    | minimum-valued stripes | ×                                                                            |
| Case 15| $U(0.05, 0.10)$                                                               | 70%                                                    | minimum-valued stripes | ×                                                                            |
| Case 16| $U(0.05, 0.10)$                                                               | 90%                                                    | minimum-valued stripes | ×                                                                            |

Fig. 14. MPSNR values of comparison methods on Pavia University data corrupted by noise with different intensities. Noise intensity per case can be found in Table II. (a) Cases with different Gaussian noise intensities. (b) Cases with different stripe intensities. (c) Cases with different “salt & pepper” intensities. (d) Cases with different stripe types.

F. Robustness of FastHyMix to Different Noise Intensities and Noise Types

To evaluate whether the performance of FastHyMix is robust under high-intensity or low-intensity noise conditions, we simulated 12 kinds of mixed noise (see cases 5–16 in Table II) and added them into Pavia University data. For example, as described in Table II, the image in case 5 is corrupted by Gaussian noise and stripes. The standard deviation of the Gaussian noise per band is sampled from a uniform distribution $U(0, 0.01)$ and oblique stripe noise randomly affected 30% of the bands. Images in cases 5–8, 9–12, and 13–16 were designed to evaluate the effect of Gaussian noise intensity, stripe noise intensity, and “salt and pepper” intensity, respectively. Denoising performance of the proposed FastHyMix and comparison methods in terms of MPSNR is shown in Fig. 14(a)–(c), where we can see FastHyMix uniformly yields best results in all cases, implying that FastHyMix is robust when addressing high-intensity and low-intensity noise.

The effect of stripe value is studied as well. The pixel value of stripes can be the possible maximum value, the possible minimum value, or the random value. We simulated the above three kinds of stripes. The image in case 8 contains maximum-valued stripes. We generated two new noisy images similar to the image in case 8 but with minimum-valued (case 17) and random-valued stripes (case 18), respectively. Results in Fig. 14(d) show the superiority of FastHyMix over other denoising methods in cases with different stripes.

VI. EXPERIMENTS WITH REAL IMAGES

The performance of hyperspectral mixed noise removal methods is also tested on two real HSI datasets, namely, Tiangong-1 image and Hyperion Cuprite image, as shown in Fig. 6(c) and (d).

A. Tiangong-1 Dataset

The Tiangong-1 dataset was acquired over an area of Qinghai Province, China, in May 2013, by a sensor placed in Tiangong-1 imager, which has a 75-band push broom scanner with a nominal bandwidth of 23-nm short wave infrared (SWIR), covering from 800 to 2500 nm. A subregion image of size 351 × 253 pixels was tested. Five bands displaying strong noise are shown in Fig. 15, where bands 29, 30, 31, and 59 contain obvious stripes. Comparing the images before denoising and after denoising, we can see that NMoG, LRTF-DFR, and FastHyMix can alleviate stripe noise in these bands. If we focus on bands 29 and 59, LRTF-DFR obtains results with incorrect illumination (see the area marked by red circles). The computational time of each method is reported in the figure caption. Qualitatively, FastHyMix yields the best result in the shortest time.
Fig. 15. Recovered bands obtained by NG-meet (63 s), LRMR (30 s), NonRLRS (29 s), KBR (203 s), NMoG (90 s), LRTF-DFR (85 s), and FastHyMix (7 s) in the real HSI Tiangong-1 data. Red arrows and dash circles are added to mark the artifacts.

Fig. 16. Recovered bands obtained by NG-meet (45 s), LRMR (28 s), NonRLRS (38 s), KBR (248 s), NMoG (83 s), LRTF-DFR (58 s), and FastHyMix (7 s) in the real HSI Hyperion Cuprite data. Red arrows and dash circles are added to mark the artifacts.

B. Hyperion Cuprite Dataset

The Cuprite HSI was captured at Cuprite, NV, USA, by Hyperion sensor, which divides the spectrum from 355 to 2577 nm into 242 channels with a spectral resolution of 10 nm. The spatial resolution of the image is 30 m. A subregion of size 240 \times 178 pixels with 177 spectral
channels (after removing water vapor absorption bands) is cropped for the test. Four bands are shown in Fig. 16, where bands 132, 134, and 138 are corrupted by severe stripes, and band 143 is mainly affected by deadlines. All methods (except NonRLRS) are able to remove the deadlines in bands 130 and 143. For severe stripes in the other three bands, we can see that NMoG and FastHyMix achieved good restoration results, while obvious stripes remained within the results of other methods. As shown in the figure caption, the running time of FastHyMix is much shorter than the comparison methods.

VII. CONCLUSION

This article introduces a fast and user-friendly hyperspectral mixed noise removal method, FastHyMix. It fits the mixed noise using a Gaussian mixture model, which is a universal approximation to any continuous distribution and hence capable of modeling a complex noise distribution. The fitting model enables us to make a good estimation of Gaussian noise intensity per band and the locations of sparse noise. The characteristics of HSIs, namely, spectral low rankness and high spatial correlation, are exploited by using a subspace representation and a deep image prior. The proposed method has some merits: 1) FastHyMix method is user-friendly in the sense that its regularization parameters are set adaptively to the noise statistics and 2) a comparison of FastHyMix with the state-of-the-art algorithms was conducted, leading to the conclusion that FastHyMix yields similar or better performance for complex mixed noise, with a much shorter running time. These characteristics put FastHyMix in a superior position to be used as an HSI denoiser.

REFERENCES

[1] J. M. Bioucas-Dias et al., “Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 5, no. 2, pp. 354–379, Apr. 2012.
[2] Y. Chang, L. Yan, and S. Zhong, “Hyper-Laplacian regularized unidirectional low-rank tensor recovery for multispectral image denoising,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 4260–4268.
[3] W. Xie, J. Lei, Y. Cui, Y. Li, and Q. Du, “Hyperspectral pansharpening by a low-rank and sparse matrix decomposition,” IEEE Trans. Geosci. Remote Sens., vol. 52, no. 8, pp. 4729–4743, Aug. 2014.
[4] T. Xie, S. Li, and B. Sun, “Hyperspectral images denoising via non-convex regularized low-rank and sparse matrix decomposition,” IEEE Trans. Image Process., vol. 29, pp. 44–56, 2020.
[5] Y. Zhang, H. Cai, W. He, H. Shen, and L. Zhang, “Double low-rank matrix decomposition for hyperspectral image denoising and destriping,” IEEE Trans. Geosci. Remote Sens., early access, Mar. 10, 2021, doi: 10.1109/TGRS.2021.3061148.
[6] Q. Xie, Q. Zhao, D. Meng, and Z. Xu, “Kronecker-basis-representation based tensor sparsity and its applications to tensor recovery,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 8, pp. 1888–1902, Aug. 2018.
[7] 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, Oct. 2012.
[8] Y.-Q. Zhao and J. Yang, “Hyperspectral image denoising via sparse representation and low-rank constraint,” IEEE Trans. Geosci. Remote Sens., vol. 53, no. 1, pp. 296–308, Jan. 2015.
[9] L. Zhang, Y. Cao, D. Meng, and Z. Xu, “Double-factorization for ill-conditioned mixtures via John ellipsoid,” IEEE Trans. Neural Netw. Learn. Syst., vol. 27, no. 8, pp. 4118–4130, Aug. 2018.
[10] Y.-B. Zheng, T.-Z. Huang, X.-L. Zhao, T.-X. Jiang, T.-H. Ma, and T.-Y. Ji, “Mixed noise removal in hyperspectral image via low-fibered-rank regularization,” IEEE Trans. Geosci. Remote Sens., vol. 58, no. 1, pp. 734–749, Jan. 2020.
[11] J. Lin, T.-Z. Huang, X.-L. Zhao, T.-H. Ma, X.-J. Jiang, and Y.-B. Zheng, “A novel non-convex low-rank tensor approximation model for hyperspectral image denoising,” Appl. Math. Comput., vol. 408, Nov. 2021, art. no. 126342.
[12] J. Lin, T.-Z. Huang, X.-L. Zhao, T.-X. Jiang, and L. Zhuang, “A tensor subspace representation-base method for hyperspectral image denoising,” IEEE Trans. Geosci. Remote Sens., vol. 59, no. 9, pp. 7739–7757, Sep. 2021.
[13] Y. Chang, L. Yan, T. Wu, and S. Zhong, “Remote sensing image stripe noise removal: From image decomposition perspective,” IEEE Trans. Geosci. Remote Sens., vol. 54, no. 12, pp. 8018–8031, Dec. 2016.
[14] L. Gao et al., “Subspace-based support vector machines for hyperspectral image classification,” IEEE Trans. Geosci. Remote Sens., vol. 50, no. 2, pp. 349–353, Feb. 2015.
[15] S. Wang, J. Li, T.-Z. Huang, T.-H. Ma, and Y.-B. Zheng, “Subspace-based support vector machine for hyperspectral image classification,” IEEE Trans. Geosci. Remote Sens., vol. 51, no. 6, pp. 3515–3526, Jun. 2013.
[16] R. Dian and J. M. Bioucas-Dias, “Adaptive hyperspectral mixed noise removal,” in Proc. IEEE Int. Geosci. Remote Sens. Symp., Jul. 2018, pp. 4035–4038.
[17] R. Dian and S. Li, “Hyperspectral image super-resolution via subspace-based low tensor multi-rank regularization,” IEEE Trans. Image Process., vol. 28, no. 10, pp. 5135–5146, Oct. 2019.
[18] R. Dian, S. Li, and L. Fang, “Learning a low-tensor rank representation for hyperspectral image super-resolution,” IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 9, pp. 2672–2683, Sep. 2019.
[19] Y. Xu, Z. Wu, J. Chanie, and Z. Wei, “Hyperspectral images super-resolution via learning high-order coupled tensor ring representation,” IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 11, pp. 4747–4763, Nov. 2020.
et al.

Q. Zhang, Q. Yuan, J. Li, F. Sun, and L. Zhang, “Deep spatio-spectral image denoising based on global and nonlocal low-rank factorizations,” IEEE Trans. Geosci. Remote Sens., early access, Jan. 8, 2021, doi: 10.1109/TGRS.2020.3046038.

K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-D transform-domain collaborative filtering,” IEEE Trans. Image Process., vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

C. Cao, J. Yu, C. Zhou, K. Hu, F. Xiao, and X. Gao, “Hyperpectral image denoising via subspace-based nonlocal low-rank and sparse representation,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 12, no. 3, pp. 973–988, Mar. 2019.

K. Babov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-D transform-domain collaborative filtering,” IEEE Trans. Image Process., vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

K. Zhang, W. Zuo, and X. Feng, “Weighted nuclear norm minimization with application to image denoising,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 2862–2869.

K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a Gaussian Denoiser: Residual learning of deep CNN for image denoising,” IEEE Trans. Image Process., vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

Y. Zhang, W. Zhao, and L. Zhang, “FFDNet: Toward a fast and flexible solution for CNN-based image denoising,” IEEE Trans. Image Process., vol. 27, no. 9, pp. 4608–4622, Sep. 2018.

S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, “Toward convolutional blind denoising of real photographs,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 1712–1722.

Q. Zhang, Q. Yuan, J. Li, X. Liu, H. Shen, and L. Zhang, “Hybrid noise removal in hyperspectral imagery with a spatial–spectral gradient network,” IEEE Trans. Geosci. Remote Sens., vol. 57, no. 10, pp. 7317–7329, Oct. 2019.

Y. Chang, L. Yan, and W. Liao, “HSI-DeNet: Hyperspectral image restoration via convolutional neural network,” IEEE Trans. Geosci. Remote Sens., vol. 57, no. 2, pp. 667–682, Feb. 2019.

Q. Zhang, Q. Yuan, J. Li, F. Sun, and L. Zhang, “Deep spatio-spectral Bayesian posterior for hyperspectral image non-i.i.d. noise removal,” ISPRS J. Photogramm. Remote Sens., vol. 164, pp. 125–137, Nov. 2020.

Q. Shi, X. Tang, T. Yang, R. Liu, and L. Zhang, “Hyperspectral image denoising using a 3-D attention denoising network,” IEEE Trans. Geosci. Remote Sens., early access, Jan. 8, 2021, doi: 10.1109/TGRS.2020.3045273.

S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, “Plug-and-play priors for model based reconstruction,” in Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP), Dec. 2013, pp. 945–948.

S. H. Chan, X. Wang, and O. A. Elgendy, “Plug-and-play admm for image restoration: Fixed-point convergence and applications,” IEEE Trans. Comput. Imag., vol. 3, no. 1, pp. 84–98, Jan. 2017.

R. Dian, S. Li, and X. Kang, “Regularizing hyperspectral and multispectral image fusion by CNN denoiser,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 3, pp. 1124–1135, Mar. 2021.

Y. Romano, M. Elad, and P. Milanfar, “The little engine that could: Regularization by denoising (RED),” SIAM J. Imag. Sci., vol. 10, no. 4, pp. 1804–1844, 2017.

X. Chen et al., “A generalized model for robust tensor factorization with noise modeling by mixture of Gaussians,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 11, pp. 5380–5393, Nov. 2018.

J. M. Bioucas-Dias and J. M. P. Nascimento, “Hyperspectral subspace identification,” IEEE Trans. Geosci. Remote Sensing, vol. 46, no. 8, pp. 2435–2445, Aug. 2008.

L. Gao, Q. Du, B. Zhang, W. Yang, and Y. Wu, “A comparative study on linear regression-based noise estimation for hyperspectral imagery,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 6, no. 2, pp. 488–498, Apr. 2013.

J. Bai and S. Ng, “Tests for skewness, kurtosis, and normality for time series data,” J. Econ. Statist., vol. 23, no. 1, pp. 49–60, Jan. 2005.

Y. Sakamoto, M. Ishiguro, and G. Kitagawa, Akaike Information Criterion Statistics, vol. 81. Dordrecht, The Netherlands: D. Reidel, 1986, p. 26853.