Hyperspectral Image Super-resolution Using Unidirectional Total Variation with Tucker Decomposition

Ting Xu, Ting-Zhu Huang*, Liang-Jian Deng*, Xi-Le Zhao, Jie Huang

Abstract—**The hyperspectral image super-resolution (HSI-SR) problem aims to improve the spatial quality of a low spatial resolution hyperspectral image (LR-HSI) by fusing the LR-HSI combined with the corresponding high spatial resolution multispectral image (HR-MSI). The generated hyperspectral image with high spatial quality, i.e., the target HR-HSI, generally has some fundamental latent properties, e.g., the sparsity and the piecewise smoothness along with the three modes (i.e., width, height, and spectral mode). However, limited works consider both properties in the HSI-SR problem. In this work, a novel unidirectional total variation-based approach is been proposed. On the one hand, we consider the target HR-HSI exhibits both the sparsity and the piecewise smoothness on the three modes, and they can be depicted well by the ℓ_1-norm and total variation (TV), respectively. On the other hand, we utilize the classical Tucker decomposition to decompose the target HR-HSI (a 3-mode tensor) as a sparse core tensor multiplied by the dictionary matrices along with the three modes. Especially, we impose the ℓ_1-norm on core tensor to characterize the sparsity, and the unidirectional TV on three dictionaries to characterize the piecewise smoothness. The proximal alternating optimization (PAO) scheme and the alternating direction method of multipliers (ADMM) are used to iteratively solve the proposed model. Experiments on three common datasets illustrate the proposed approach has better performance than some current state-of-the-art HSI-SR methods.**

***Index Terms*—Hyperspectral image super-resolution, image fusion, piecewise smoothness, sparsity, unidirectional total variation.**

I. Introduction

HYPERSPECTRAL images (HSIs) contain abundant spectral information because of the powerful capture ability of hyperspectral imaging sensors. Therefore, HSIs have been involved in many applications [1], [2]. However, the spatial resolution is inevitably decreased because of the limited sun irradiance [3]. Thus, we need to utilize some techniques to enhance the spatial quality of the HSIs [4]–[11]. We know that MSIs are obtained with poor spectral resolution but abundant spatial resolution compared with HSIs. Therefore, fusing a LR-HSI combined with the corresponding HR-MSI to generate

*Corresponding author: Ting-Zhu Huang, Liang-Jian Deng.

The work is supported by National Natural Science Foundation of China grants 61772003, 61702083, and 61876203 and Key Projects of Applied Basic Research in Sichuan Province (Grant No. 2020YJ0216).

All authors are with the Research Center for Image and Vision Computing, School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 611731, P.R.China (Email: 17742879536@163.com; tingzhuhuang@126.com; liangjian.deng@uestc.edu.cn; xlzhao122003@163.com; happyjie07mo@163.com).

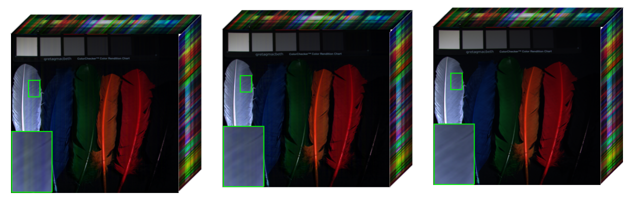


Fig. 1. The fused results reconstructed by CSTF [12] and the proposed method.

the target HR-HSI has become an increasingly promising way for the HSI-SR problem. The current HSI-SR approaches can be mainly divided into three families: non-factorization based approaches [13]–[22], matrix factorization based approaches [23]–[33] and tensor factorization based approaches [12], [34]–[40].

Non-factorization based HSI-SR approaches could generate the target HR-HSI directly under suitable priors or without specific priors. Fu et al. in work [13] fuse a LR-HSI combined with the mosaic RGB by utilizing the non-local low-rank regularization to directly get the target HR-HSI. Qu et al. in work [14] propose an efficient unsupervised deep convolutional neural networks (CNN) method for the HSI-SR problem, which could generate the target HR-HSI without specific priors. Jiang et al. in work [21] apply the advanced residual learning based single gray/RGB image super-resolution algorithms for the single HSI-SR. Chen et al. in work [22] utilize a unified framework based on rank minimization (UFRM) for super-resolution reconstruction of hyperspectral remote sensing images.

Matrix factorization based HR-HSI approaches mainly utilizing decomposing the target HR-HSI as the spectral basis and the corresponding coefficients, therefore, the HSI-SR problem could be equivalent to the estimation of the basis and the corresponding coefficients. Simoẽs et al. in work [23] exploit the low-dimensional subspace representation and the vector total variation to effectively solve the HSI-SR problem. Lanaras et al. in work [28] utilize the coupled non-negative matrix decomposition to alternately update spectral basis and coefficients to get the target HSIs. Wei et al. in work [29] exploit the circulant and downsampling matrices related to the HSI-SR problem to give a classical Sylvester equation, which has a closed-form solution, and it can easily extend to the HSI-SR problem with prior information. Note that the matrix is generally got from the matrix unfolding operation, therefore, matrix factorization based approaches may not completely exploit the spatial-spectral structures information of HSIs.

Tensors can fully exploit the inherent data structures information and there are many tools that can be used to deal with tensors. Besides, the target HR-HSI *can be treated as a 3-mode tensor*. Therefore, it is effective to deal with the HSI-SR problem from the tensor's point of view. Dian et al. in work [34] first propose a non-local sparse tensor factorization approach used for the HSI-SR problem, where they first divide the target HR-HSI as some cubes and then utilize the classical Tucker decomposition to factorize each cube as a sparse core tensor multiplied by dictionary matrices with three modes. Besides, they also assume that similar cubes have the same dictionary matrices. Then, Li et al. in work [12] present the coupled sparse tensor factorization (CSTF) method, where they use the Tucker decomposition to directly decompose the target HR-HSI and then utilize the high spatial-spectral correlation in the target HR-HSI to promote the sparsity of the core tensor. Dian et al. in work [38] give a subspace-based low tensor multi-rank regularization (LTMR) approach, where they approximate the HR-HSI by spectral subspace and the coefficients. Then, they obtain the spectral subspace by singular value decomposition, meanwhile, the corresponding coefficients are generated by utilizing the LTMR prior. The above methods have achieved good results in the HSI-SR problem, but they do not take into account the sparsity and the piecewise smoothness of the target HR-HSIs, simultaneously.

Due to the mentioned limitations in the existing tensor factorization based approaches. There is still much room to improve them. We report a new tensor factorization based approach in this work. It considers the sparsity and the piecewise smoothness of the target HR-HSIs and is a more comprehensive characterization of the properties of the target HR-HSIs. Fig. 1 shows the visual effect comparison between the CSTF [12] and the proposed method.

We organize the following parts as follows. Section II reports the related works and motivations. Section III introduces the proposed model. Section IV reports the proposed algorithm. Experiments and discussions are presented in Section V. Finally, we give a conclusion in Section V.

II. Related Works and Motivations

A. Notations and Preliminaries

In this work, we utilize boldface lowercase letters $\mathbf{x}$, boldface capital letters $\mathbf{X}$, and calligraphic letters $\mathcal{X}$ to represent vector, matrix and tensor, respectively. $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \ldots \times I_N}$ denotes an N-mode tensor and $\mathbf{x}_{i_1 i_2 \ldots i_N} (1 \le i_n \le I_n)$ denote it's elements. $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 I_2 \ldots I_{n-1} I_{n+1}, \ldots, I_N}$ denotes the n-mode unfolding matrix of $\mathcal{X}$. $||\mathcal{X}||_1 = \sum_{i_1,\ldots,i_N} |\mathbf{x}_{i1\ldots i_N}|$ and $||\mathcal{X}||_F = \sqrt{\sum_{i_1,\ldots,i_N} |\mathbf{x}_{i1\ldots i_N}|^2}$ denote the ℓ_1-norm and the Frobenius norm, respectively.

$\mathcal{X} \times_n \mathbf{C}$ represents the n-mode product of the tensor $\mathcal{X}$ and the matrix $\mathbf{C} \in \mathbb{R}^{J_n \times I_n}$, and it denotes an N-dimensional tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \cdots \times I_{n-1} \times J_n \times I_{n+1} \cdots \times I_N}$, its elements are calculated

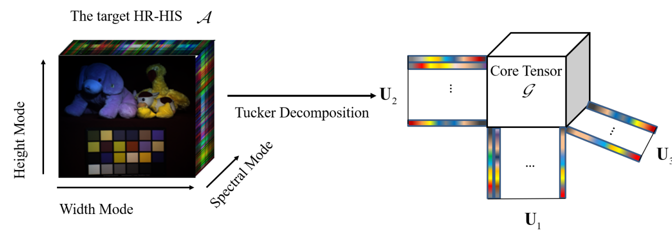


Fig. 2. Illustration the process of Tucker decomposition.

by

$$y_{i_1 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} = \sum_{i_n} \mathbf{x}_{i_1 \ldots i_{n-1} i_n i_{n+1} \ldots i_N} \mathbf{c}_{j_n i_n}. \tag{1}$$

The n-mode product, i.e., $\mathcal{X} \times_n \mathbf{C}$, can be equivalent to matrix multiplication, namely $\mathbf{Y}_{(n)} = \mathbf{C}\mathbf{X}_{(n)}$. The order of the multiplications is independent for different modes, i.e.,

$$\mathcal{X} \times_m \mathbf{E} \times_n \mathbf{F} = \mathcal{X} \times_n \mathbf{F} \times_m \mathbf{E} \ (m \neq n), \tag{2}$$

especially,

$$\mathcal{X} \times_m \mathbf{E} \times_n \mathbf{F} = \mathcal{X} \times_n \mathbf{F}\mathbf{E} \ (m = n). \tag{3}$$

Given the set of matrices $\mathbf{B}_n \in \mathbb{R}^{J_n \times I_n} (n = 1, 2, \ldots, N)$, we define the tensor $\mathcal{C} \in \mathbb{R}^{J_1 \times J_2 \ldots J_N}$ as

$$\mathcal{C} = \mathcal{X} \times_1 \mathbf{B}_1 \times_2 \mathbf{B}_2 \ldots \times_N \mathbf{B}_N, \tag{4}$$

then, we have

$$\mathbf{c} = (\mathbf{B}_N \otimes \mathbf{B}_{N-1} \otimes \ldots \otimes \mathbf{B}_1) x, \tag{5}$$

where $\otimes$ denotes Kronecker product, $\mathbf{c} = \text{vec}(\mathcal{C}) \in \mathbb{R}^J (J = \prod_{n=1}^N J_n)$, and $\mathbf{x} = \text{vec}(\mathcal{X}) \in \mathbb{R}^I (I = \prod_{n=1}^N I_n)$ are generated by stacking all the 1-mode vectors of $\mathcal{C}$ and $\mathcal{X}$, respectively.

B. Related Works

As mentioned in Li et al.'s work [12], the target HR-HSI $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ as a 3-mode tensor could be factorized as a core tensor multiplied by the dictionary matrices along with the three modes (i.e., width mode (1-mode), height mode (2-mode), and spectral mode (3-mode)) via the classical Tucker decomposition, as visualized in Fig. 2. Therefore, the target HR-HSI $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ could be decomposed as

$$\mathcal{A} = \mathcal{G} \times_1 \mathbf{U_1} \times_2 \mathbf{U_2} \times_3 \mathbf{U_3}, \tag{6}$$

where $\mathcal{G} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ represents the core tensor, $\mathbf{U_1} \in \mathbb{R}^{I_1 \times n_1}$, $\mathbf{U_2} \in \mathbb{R}^{I_2 \times n_2}$, and $\mathbf{U_3} \in \mathbb{R}^{I_3 \times n_3}$ denote the dictionary matrices of the 1-mode, 2-mode, and 3-mode, respectively.

The observed LR-HSI $\mathcal{B} \in \mathbb{R}^{i_1 \times i_2 \times I_3} (0 < i_1 < I_1, 0 < i_2 < I_2)$ could be formulated as a spatially downsampled form of $\mathcal{A}$, i.e.,

$$\mathcal{B} = \mathcal{A} \times_1 \mathbf{D_1} \times_2 \mathbf{D_2}, \tag{7}$$

where $\mathbf{D_1} \in \mathbb{R}^{i_1 \times I_1}$ and $\mathbf{D_2} \in \mathbb{R}^{i_2 \times I_2}$ are downsampling matrices of 1-mode and 2-mode, respectively. Meanwhile, superseding $\mathcal{A}$ by (6), $\mathcal{B}$ can be rewritten by

$$\begin{aligned} \mathcal{B} &= \mathcal{G} \times_1 (\mathbf{D_1 U_1}) \times_2 (\mathbf{D_2 U_2}) \times_3 \mathbf{U_3} \\ &= \mathcal{G} \times_1 \widehat{\mathbf{U_1}} \times_2 \widehat{\mathbf{U_2}} \times_3 \mathbf{U_3}, \end{aligned} \tag{8}$$

where $\widehat{\mathbf{U_1}} = \mathbf{D_1 U_1} \in \mathbb{R}^{i_1 \times n_1}$ and $\widehat{\mathbf{U_2}} = \mathbf{D_2 U_2} \in \mathbb{R}^{i_2 \times n_2}$ denote the downsampled form of $\mathbf{U_1}$ and of $\mathbf{U_2}$, respectively.

Similarly, the HR-MSI $\mathcal{C} \in \mathbb{R}^{I_1 \times I_2 \times i_3}$ $(0 < i_3 < I_3)$ could be treated as a spectrally downsampled form of $\mathcal{A}$, i.e.,

$$\mathcal{C} = \mathcal{A} \times_3 \mathbf{D_3}, \tag{9}$$

where $\mathbf{D_3} \in \mathbb{R}^{i_3 \times I_3}$ denotes the downsampling matrix of 3-mode. Similarly, superseding $\mathcal{A}$ by (6), $\mathcal{C}$ can be rewritten by

$$\begin{aligned} \mathcal{C} &= \mathcal{G} \times_1 \mathbf{U_1} \times_2 \mathbf{U_2} \times_3 (\mathbf{D_3 U_3}) \\ &= \mathcal{G} \times_1 \mathbf{U_1} \times_2 \mathbf{U_2} \times_3 \widehat{\mathbf{U_3}}, \end{aligned} \tag{10}$$

where $\widehat{\mathbf{U_3}} = \mathbf{D_3 U_3} \in \mathbb{R}^{i_3 \times n_3}$ is the downsampled form of $\mathbf{U_3}$.

Therefore, the estimation of $\mathcal{A}$ is equivalent to the estimation of $\mathbf{U_1}$, $\mathbf{U_2}$, $\mathbf{U_3}$, and $\mathcal{G}$, as visualized in Fig. 2.

C. Motivations

Based on the Tucker decomposition and the downsampled processing along three dimensions reported in Section II-B, the HSI-SR problem could be represented by

$$\begin{aligned} \underset{\mathbf{U_1},\mathbf{U_2},\mathbf{U_3},\mathcal{G}}{\operatorname{argmin}} \quad & ||\mathcal{B} - \mathcal{G} \times_1 \widehat{\mathbf{U_1}} \times_2 \widehat{\mathbf{U_2}} \times_3 \mathbf{U_3}||_F^2 \\ & + ||\mathcal{C} - \mathcal{G} \times_1 \mathbf{U_1} \times_2 \mathbf{U_2} \times_3 \widehat{\mathbf{U_3}}||_F^2. \end{aligned} \tag{11}$$

Obviously, problem (11) is a typical ill-posed inverse problem. Therefore, in order to solve the problem (11), some prior information on the target HR-HSI $\mathcal{A}$ is necessary. In this paper, we consider the sparsity and the piecewise smoothness on both spatial and spectral modes.

The sparsity of the core tensor In the problem of HSI-SR, the sparsity in the spatial and spectral domains has been proven. The sparsity in the spectral dimension has been extensively used for HSI-SR problems. Based on the classical Tucker decomposition, Li et al. in work [12] apply the sparsity regularization to both the spectral and spatial modes by assuming that the core tensor exhibits sparsity. Hence, in this work, we impose the ℓ_1-norm on core tensor to characterize the sparsity on the spatial and spectral dimensions, simultaneously.

The piecewise smoothness of the three dictionaries In the problem of HSI-SR, the piecewise smoothness of the target HR-HSI has been demonstrated. Based on the classical Tucker decomposition, we assume that it could be regarded as the three dictionaries that exhibit the piecewise smoothness property. Meanwhile, it can be depicted by the total variation (TV) well. Therefore, in this work, we apply the unidirectional TV to the three dictionaries to characterize the piecewise smoothness of the target HR-HSI.

III. Proposed Method

A. Proposed Model

Based on the motivations noted in Section II-C, we give the following model to solve the HSI-SR problem, i.e.,

$$\begin{aligned} \underset{\mathbf{U_1},\mathbf{U_2},\mathbf{U_3},\mathcal{G}}{\operatorname{argmin}} \quad & ||\mathcal{B} - \mathcal{G} \times_1 \widehat{\mathbf{U_1}} \times_2 \widehat{\mathbf{U_2}} \times_3 \mathbf{U_3}||_F^2 \\ & + ||\mathcal{C} - \mathcal{G} \times_1 \mathbf{U_1} \times_2 \mathbf{U_2} \times_3 \widehat{\mathbf{U_3}}||_F^2 \\ & + \lambda_1 ||\mathcal{G}||_1 + \lambda_2 ||\mathbf{D_y U_1}||_1 \\ & + \lambda_3 ||\mathbf{D_y U_2}||_1 + \lambda_4 ||\mathbf{D_y U_3}||_1, \end{aligned} \tag{12}$$

where $\lambda_i, (i = 1, 2, 3, 4)$ are positive regularization parameters, $\mathbf{D_y}$ is a finite difference operator along the vertical direction and given by

$$\mathbf{D_y} = \begin{bmatrix} 1 & -1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 & -1 \end{bmatrix}.$$

Next, we will give the algorithm to solve the proposed model (12) efficiently.

IV. Proposed Algorithm

The proposed model (12) is non-convex because of the coupling variables $\mathcal{G}$, $\mathbf{U_1}$, $\mathbf{U_2}$, and $\mathbf{U_3}$. However, the optimization problem is convex for each variable when we keep the other variables fixed. Here, we employ the PAO scheme [41], [42] to solve it, the PAO iteration is simply shown as follows,

$$\begin{cases} \mathbf{U_1} = \underset{\mathbf{U_1}}{\operatorname{argmin}} f(\mathbf{U_1}, \mathbf{U_2}, \mathbf{U_3}, \mathcal{G}) + \beta ||\mathbf{U_1} - \mathbf{U_1}^{pre}||_F^2, \\ \mathbf{U_2} = \underset{\mathbf{U_2}}{\operatorname{argmin}} f(\mathbf{U_1}, \mathbf{U_2}, \mathbf{U_3}, \mathcal{G}) + \beta ||\mathbf{U_2} - \mathbf{U_2}^{pre}||_F^2, \\ \mathbf{U_3} = \underset{\mathbf{U_3}}{\operatorname{argmin}} f(\mathbf{U_1}, \mathbf{U_2}, \mathbf{U_3}, \mathcal{G}) + \beta ||\mathbf{U_3} - \mathbf{U_3}^{pre}||_F^2, \\ \mathcal{G} = \underset{\mathcal{G}}{\operatorname{argmin}} f(\mathbf{U_1}, \mathbf{U_2}, \mathbf{U_3}, \mathcal{G}) + \beta ||\mathcal{G} - \mathcal{G}^{pre}||_F^2, \end{cases} \tag{13}$$

where the function $f(\mathbf{U_1}, \mathbf{U_2}, \mathbf{U_3}, \mathcal{G})$ is the implicit definition of (12), $(\cdot)^{pre}$ and β denote the last iteration result and a positive number, respectively.

Next, we will report the solution of the four optimization problems in (13) in detail.

A. The optimization problem of $\mathbf{U_1}$

With fixing $\mathbf{U_2}$, $\mathbf{U_3}$, and $\mathcal{G}$, the optimization problem of $\mathbf{U_1}$ in (13) is given by

$$\begin{aligned} \underset{\mathbf{U_1}}{\operatorname{argmin}} \quad & ||\mathcal{B} - \mathcal{G} \times_1 \widehat{\mathbf{U_1}} \times_2 \widehat{\mathbf{U_2}} \times_3 \mathbf{U_3}||_F^2 \\ & + ||\mathcal{C} - \mathcal{G} \times_1 \mathbf{U_1} \times_2 \mathbf{U_2} \times_3 \widehat{\mathbf{U_3}}||_F^2 \\ & + \lambda_2 ||\mathbf{D_y U_1}||_1 + \beta ||\mathbf{U_1} - \mathbf{U_1}^{pre}||_F^2, \end{aligned} \tag{14}$$

where $\mathbf{U_1}^{pre}$ is the last estimated dictionary of 1-mode and $\mathbf{D_y} \in \mathbb{R}^{(I_1-1) \times I_1}$ denotes the difference matrix along the vertical direction of $\mathbf{U_1}$.

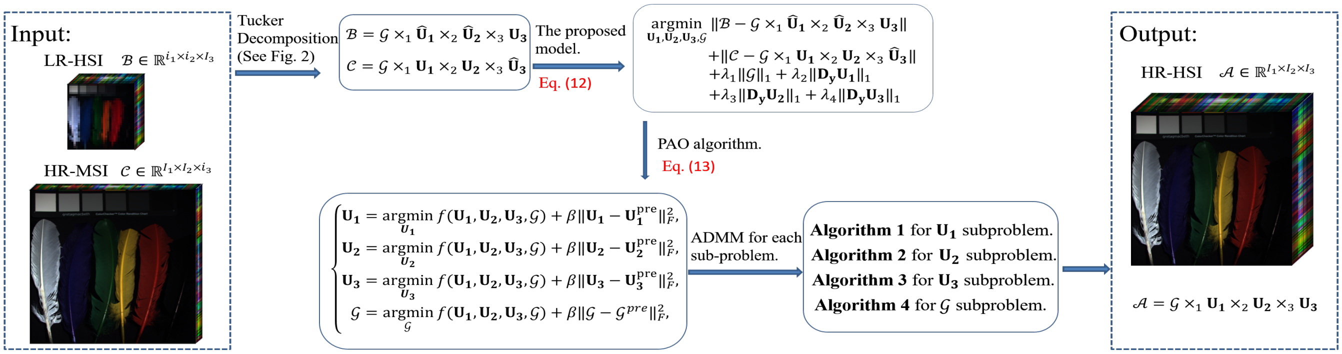


Fig. 3. An overall flow chart is presented to better grasp the ideas of the work.

By 1-mode matrix unfolding, problem (14) can be represented by

$$\underset{\mathbf{U_1}}{\operatorname{argmin}} \quad ||\mathbf{B}_{(1)} - \mathbf{D_1 U_1 X_1}||_F^2 + ||\mathbf{C}_{(1)} - \mathbf{U_1 Y_1}||_F^2 + \lambda_2||\mathbf{D_y U_1}||_1 + \beta||\mathbf{U_1} - \mathbf{U_1}^{pre}||_F^2, \tag{15}$$

where $\mathbf{B}_{(1)}$ and $\mathbf{C}_{(1)}$ are 1-mode unfolding matrixes of $\mathcal{B}$ and $\mathcal{C}$, respectively; $\mathbf{X_1}$ and $\mathbf{Y_1}$ are denoted by $\mathbf{X_1} = (\mathcal{G} \times_2 \widehat{\mathbf{U_2}} \times_3 \mathbf{U_3})_{(1)}$ and $\mathbf{Y_1} = (\mathcal{G} \times_2 \mathbf{U_2} \times_3 \widehat{\mathbf{U_3}})_{(1)}$, respectively.

B. The optimization problem of $\mathbf{U_2}$

With fixing $\mathbf{U_1}$, $\mathbf{U_3}$, and $\mathcal{G}$, the optimization problem of $\mathbf{U_2}$ in (13) is given by

$$\begin{aligned}\underset{\mathbf{U_2}}{\operatorname{argmin}} \quad & ||\mathcal{B} - \mathcal{G} \times_1 \widehat{\mathbf{U_1}} \times_2 \widehat{\mathbf{U_2}} \times_3 \mathbf{U_3}||_F^2 \\ & + ||\mathcal{C} - \mathcal{G} \times_1 \mathbf{U_1} \times_2 \mathbf{U_2} \times_3 \widehat{\mathbf{U_3}}||_F^2 \\ & + \lambda_3||\mathbf{D_y U_2}||_1 + \beta||\mathbf{U_2} - \mathbf{U_2}^{pre}||_F^2,\end{aligned} \tag{16}$$

where $\mathbf{U_2}^{pre}$ is the last estimated dictionary of 2-mode and $\mathbf{D_y} \in \mathbb{R}^{(I_2-1)\times I_2}$ denotes the difference matrix along the vertical direction of $\mathbf{U_2}$.

By 2-mode matrix unfolding, problem (16) can be represented by

$$\underset{\mathbf{U_2}}{\operatorname{argmin}} \quad ||\mathbf{B}_{(2)} - \mathbf{D_2 U_2 X_2}||_F^2 + ||\mathbf{C}_{(2)} - \mathbf{U_2 Y_2}||_F^2 + \lambda_3||\mathbf{D_y U_2}||_1 + \beta||\mathbf{U_2} - \mathbf{U_2}^{pre}||_F^2, \tag{17}$$

where $\mathbf{B}_{(2)}$ and $\mathbf{C}_{(2)}$ are 2-mode unfolding matrixes of $\mathcal{B}$ and $\mathcal{C}$, respectively; $\mathbf{X_2}$ and $\mathbf{Y_2}$ are denoted by $\mathbf{X_2} = (\mathcal{G} \times_1 \widehat{\mathbf{U_1}} \times_3 \mathbf{U_3})_{(2)}$ and $\mathbf{Y_2} = (\mathcal{G} \times_1 \mathbf{U_1} \times_3 \widehat{\mathbf{U_3}})_{(2)}$, respectively.

C. The optimization problem of $\mathbf{U_3}$

With fixing $\mathbf{U_1}$, $\mathbf{U_2}$, and $\mathcal{G}$, the optimization problem of $\mathbf{U_3}$ in (13) can be represented as follows:

$$\begin{aligned}\underset{\mathbf{U_3}}{\operatorname{argmin}} \quad & ||\mathcal{B} - \mathcal{G} \times_1 \widehat{\mathbf{U_1}} \times_2 \widehat{\mathbf{U_2}} \times_3 \mathbf{U_3}||_F^2 \\ & + ||\mathcal{C} - \mathcal{G} \times_1 \mathbf{U_1} \times_2 \mathbf{U_2} \times_3 \widehat{\mathbf{U_3}}||_F^2 \\ & + \lambda_4||\mathbf{D_y U_3}||_1 + \beta||\mathbf{U_3} - \mathbf{U_3}^{pre}||_F^2,\end{aligned} \tag{18}$$

where $\mathbf{U_3}^{pre}$ is the last estimated dictionary of 3-mode and $\mathbf{D_y} \in \mathbb{R}^{(I_3-1)\times I_3}$ denotes the difference matrix along the vertical direction of $\mathbf{U_3}$.

By using 3-mode matrix unfolding, (18) can be represented by

$$\underset{\mathbf{U_3}}{\operatorname{argmin}} \quad ||\mathbf{B}_{(3)} - \mathbf{U_3 X_3}||_F^2 + ||\mathbf{C}_{(3)} - \mathbf{D_3 U_3 Y_3}||_F^2 + \lambda_4||\mathbf{D_y U_3}||_1 + \beta||\mathbf{U_3} - \mathbf{U_3}^{pre}||_F^2, \tag{19}$$

where $\mathbf{B}_{(3)}$ and $\mathbf{C}_{(3)}$ are 3-mode unfolding matrixes of $\mathcal{B}$ and $\mathcal{C}$, respectively; $\mathbf{X_3}$ and $\mathbf{Y_3}$ are denoted by $\mathbf{X_3} = (\mathcal{G} \times_1 \widehat{\mathbf{U_1}} \times_2 \widehat{\mathbf{U_2}})_{(3)}$ and $\mathbf{Y_3} = (\mathcal{G} \times_1 \mathbf{U_1} \times_2 \mathbf{U_2})_{(3)}$, respectively.

D. The optimization problem of $\mathcal{G}$

With fixing dictionary matrices $\mathbf{U_1}$, $\mathbf{U_2}$, and $\mathbf{U_3}$, the optimization problem of the core tensor $\mathcal{G}$ in (13) is given by

$$\begin{aligned}\underset{\mathcal{G}}{\operatorname{argmin}} \quad & ||\mathcal{B} - \mathcal{G} \times_1 \widehat{\mathbf{U_1}} \times_2 \widehat{\mathbf{U_2}} \times_3 \mathbf{U_3}||_F^2 \\ & + ||\mathcal{C} - \mathcal{G} \times_1 \mathbf{U_1} \times_2 \mathbf{U_2} \times_3 \widehat{\mathbf{U_3}}||_F^2 \\ & + \lambda_1||\mathcal{G}||_1 + \beta||\mathcal{G} - \mathcal{G}^{pre}||_F^2,\end{aligned} \tag{20}$$

where $\mathcal{G}^{pre}$ is the last estimated core tensor.

Note that, problem (15), (17), (19), and (20) are all convex. Therefore, we utilize ADMM to solve them. Since the solving process of the problem (15), (17), and (19) are similar, to look more concise, we put the solving details of the four problems and each variable updating's computational complexity to Section VII as an appendix.

In Section VII, **Algorithm 1-Algorithm 4** summarize the solving process of the four subproblems in (13), respectively. Fig. 3 displays the overall flowchart, which could better grasp the ideas of the work.

*E. The termination criterion for **Algorithm 1** to **Algorithm 4***

The relative change ($RelCha$) is defined as

$$RelCha = \frac{||\mathbf{X}^{(t+1)} - \mathbf{X}^{(t)}||_F}{||\mathbf{X}^{(t)}||_F}. \tag{21}$$

In this paper, for **Algs. 1-4**, we use the condition, that is, i) the algorithm reaches the maximum number of iterations, or ii) $RelCha$ is less than the tolerance, as the stopping criterion. Since the characteristics of different datasets are not always the same, thus for better performance of our method, we need to empirically set corresponding suitable tolerances for different datasets. Table I summaries the termination conditions and how many iterations the related algorithms need.

TABLE I
THE DISCUSSION OF STOPPING CRITERION FOR **ALGORITHM 1-ALGORITHM 4**.

	CAVE dataset [43]	Harvard dataset [44]	Pavia University dataset [45]
Maximum number of iterations	100 for **Algs. 1-4**	100 for **Algs. 1-4**	100 for **Algs. 1-4**
RelCha	10^{-2} for **Algs. 1-4**	10^{-2} for **Algs. 1-4**	10^{-4} for **Algs. 1-4**
Number of iterations (no more than)	25 for **Alg. 1** 34 for **Alg. 2** 3 for **Alg. 3** 9 for **Alg. 4**	73 for **Alg. 1** 29 for **Alg. 2** 3 for **Alg. 3** 35 for **Alg. 4**	6 for **Alg. 1** 6 for **Alg. 2** 100 for **Alg. 3** 45 for **Alg. 4**

F. Convergence of the proposed algorithm

Same with Li et al.'s work [12], we initialize $\mathbf{U_1}$ and $\mathbf{U_2}$ from the HR-MSI by dictionary-updates-cycles KSVD (DUC-KSVD) algorithm [46], and $\mathbf{U_3}$ from the LR-HSI by simplex identification split augmented Lagrangian (SISAL) algorithm [47].

An outline of the proposed algorithm is presented in **Algorithm 5**.

Algorithm 5 Solving the proposed model (12) by PAO scheme.

Initializing $\mathbf{U_1}$, $\mathbf{U_2}$ via DUC-KSVD algorithm [46];
Initializing $\mathbf{U_3}$ via SISAL algorithm [47];
Initializing $\mathcal{G}$ with **Alg. 4**;
While not converged **do**
  Step 1 Updating the dictionary matrix $\mathbf{U_1}$ by **Alg. 1**, $\widehat{\mathbf{U_1}} = \mathbf{D_1}\mathbf{U_1}$, ${\mathbf{U_1}}^{pre} = \mathbf{U_1}$;
  Step 2 Updating the dictionary matrix $\mathbf{U_2}$ by **Alg. 2**, $\widehat{\mathbf{U_2}} = \mathbf{D_2}\mathbf{U_2}$, ${\mathbf{U_2}}^{pre} = \mathbf{U_2}$;
  Step 3 Updating the dictionary matrix $\mathbf{U_3}$ by **Alg. 3**, $\widehat{\mathbf{U_3}} = \mathbf{D_3}\mathbf{U_3}$, ${\mathbf{U_3}}^{pre} = \mathbf{U_3}$;
  Step 4 Updating the core tensor $\mathcal{G}$ by **Alg. 4**, $\mathcal{G}^{pre} = \mathcal{G}$;
end while
Estimate the target HR-HSI $\mathcal{A}$ by (6).

To provide the convergence of **Alg. 5** conveniently, we rewrite the function f as

$$f(\mathbf{U_1},\mathbf{U_2},\mathbf{U_3},\mathcal{G}) = Q(\mathbf{U_1},\mathbf{U_2},\mathbf{U_3},\mathcal{G}) + f_1(\mathcal{G}) + f_2(\mathbf{U_1}) + f_3(\mathbf{U_2}) + f_4(\mathbf{U_3}), \tag{22}$$

where

$$Q(\mathbf{U_1},\mathbf{U_2},\mathbf{U_3},\mathcal{G}) = ||\mathcal{B} - \mathcal{G}\times_1 \widehat{\mathbf{U_1}} \times_2 \widehat{\mathbf{U_2}} \times_3 \mathbf{U_3}||_F^2 + ||\mathcal{C} - \mathcal{G}\times_1 \mathbf{U_1} \times_2 \mathbf{U_2} \times_3 \widehat{\mathbf{U_3}}||_F^2, \tag{23}$$

and

$$f_1(\mathcal{G}) = \lambda_1||\mathcal{G}||_1, \tag{24}$$

$$f_2(\mathbf{U_1}) = \lambda_2||\mathbf{D_y}\mathbf{U_1}||_1, \tag{25}$$

$$f_3(\mathbf{U_2}) = \lambda_3||\mathbf{D_y}\mathbf{U_2}||_1, \tag{26}$$

$$f_4(\mathbf{U_3}) = \lambda_4||\mathbf{D_y}\mathbf{U_3}||_1. \tag{27}$$

Proposition 1: Assume that problem (14) ($\mathbf{U_1}$-subproblem), problem (16) ($\mathbf{U_2}$-subproblem), problem (18) ($\mathbf{U_3}$-subproblem), and problem (20) ($\mathcal{G}$-subproblem) have exact solutions [1].

Proposition 2: We assume that the sequence $({\mathbf{U_1}}^{(t)}, {\mathbf{U_2}}^{(t)}, {\mathbf{U_3}}^{(t)}, \mathcal{G}^{(t)})_{t\in\mathbb{N}}$ obtained by **Alg. 5**, is bounded. Then, the sequence converges to some critical points of $f(\mathbf{U_1},\mathbf{U_2},\mathbf{U_3},\mathcal{G})$.

Proof: Firstly, function $Q(\mathbf{U_1},\mathbf{U_2},\mathbf{U_3},\mathcal{G})$ is C^1 with Lipschitz continuous gradient (the result of the boundedness of $({\mathbf{U_1}}^{(t)}, {\mathbf{U_2}}^{(t)}, {\mathbf{U_3}}^{(t)}, \mathcal{G}^{(t)})_{t\in\mathbb{N}}$); $f_1(\mathcal{G})$, $f_2(\mathbf{U_1})$, $f_3(\mathbf{U_2})$, and $f_4(\mathbf{U_3})$ are lower semicontinuous and proper. Secondly, $f(\mathbf{U_1},\mathbf{U_2},\mathbf{U_3},\mathcal{G}) : \mathbb{R}^{I_1 n_1} \times \mathbb{R}^{I_2 n_2} \times \mathbb{R}^{I_3 n_3} \times \mathbb{R}^{n_1 n_2 n_3} \to \mathbb{R}$ is bounded below and Kurdyka-Lojasiewicz (see [[41], Sec. 2.2]). With the two conditions, we find that **Alg. 5** is an exemplar of (61), (62) and (63) shown in [41] with $\mathbf{B}_i = \beta\mathbf{I}$, $\beta > 0$ (see [[41], Remark 6.1]). Therefore, the proof of convergence of **Alg. 5** is an application of [[41], Th. 6.2].

G. Computational Complexity of the Proposed Algorithm

In this work, we solve the $\mathbf{U_1}$-subproblem, $\mathbf{U_2}$-subproblem, $\mathbf{U_3}$-subproblem, and $\mathcal{G}$-subproblem in (13) by ADMM. For the $\mathbf{U_1}$-subproblem, during each iteration of ADMM, the most time-consuming step is updating $\mathbf{U_1}$ (i.e., (33)) by CG. During each iteration of CG, the most time-consuming is that multiplying the system matrix by a vector, it's time complexity is $O(n_1^2 I_1^2)$, which could be reduced to $O(n_1 I_1^2 + n_1^2 I_1)$ due to the matrix representation. Similar to the update of $\mathbf{U_1}$, in each iteration of CG algorithm, the complexity of $\mathbf{U_2}$ and $\mathbf{U_3}$ are $O(n_2 I_2^2 + n_2^2 I_2)$ and $O(n_3 I_3^2 + n_3^2 I_3)$, respectively. For the $\mathcal{G}$-subproblem, in each iteration of ADMM, the most time-consuming are (60) and (65), whose complexity is $O(n_1^3 n_2^3 n_3^3)$. Fortunately, the complexity could be reduced to $O(n_1^2 n_2 n_3 + n_1 n_2^2 n_3 + n_1 n_2 n_3^2)$ when we utilize (61) and (66) to do these computations. In summary, the computational complexity of each PAO iteration is

$$\begin{aligned}
&O(N_{ADMM}(N_{CG}(n_1 I_1^2 + n_1^2 I_1)))+\\
&O(N_{ADMM}(N_{CG}(n_2 I_2^2 + n_2^2 I_2)))+\\
&O(N_{ADMM}(N_{CG}(n_3 I_3^2 + n_3^2 I_3)))+\\
&O(N_{ADMM}(n_1^2 n_2 n_3 + n_1 n_2^2 n_3 + n_1 n_2 n_3^2)),
\end{aligned}$$

where N_{ADMM} and N_{CG} are the iteration number of ADMM and CG, respectively.

[1]Actually, it is difficult to efficiently solve the closed-form solution for each subproblem, so we use ADMM (the convergence guaranteed) to get the approximate solution for each subproblem efficiently.

V. EXPERIMENTS

A. Compared Algorithms

In this section, we present the comparisons between the proposed approach and current state-of-the-art methods including 1) a convex formulation for hyperspectral image superresolution via subspace-based regularization (HySure) proposed by Simões et al. [23], 2) Hyperspectral super-resolution by coupled spectral unmixing (CSU) proposed by Lanaras et al. [28], 3) fast fusion of multi-band images based on solving a Sylvester equation (FUSE) proposed by Wei et al. [29], and 4) fusing hyperspectral and multispectral images via coupled sparse tensor factorization (CSTF) proposed by Li et al. [12].

B. Datasets

We introduce three simulated HSI datasets used for this work.

The first is the Pavia University dataset with a size of $610 \times 340 \times 115$ [45]. In this work, we reduce spectral bands as 93 bands by removing low signal-to-noise ratio bands and pick the up-left 256×256 block of each band as the band of GT (i.e., the size of GT is $256 \times 256 \times 93$). In order to obtain the simulated LR-HSI with the size of $16{\times}16{\times}93$, we downsample the GT by averaging the $16{\times}16$ disjoint spatial blocks. Besides, we acquire the four-band simulated HR-MSI by the IKONOS-like reflectance spectral response filter.

The second is the Columbia Computer Vision Laboratory (CAVE) [43]. It has 32 indoor HSIs and each of them is with the size of $512 \times 512 \times 31$. In this work, we only choose six HSIs from the CAVE dataset as the ground truth (GT) used for experiments reported in Section . In order to obtain the simulated LR-HSI with the size of $32{\times}32{\times}31$, we downsample each of them by averaging the $16{\times}16$ disjoint spatial blocks. Besides, the three-band simulated HR-MSIs are generated by a Nikon D700 camera.

The third is the Harvard dataset [44], which includes 50 HSIs of both indoor and outdoor scenes featuring a diversity of objects, materials, and scale under daylight illumination. Each HSI has a spatial resolution of $1392{\times}1040$ and 31 spectral bands. The HSIs of the scenes are acquired at a wavelength interval 10 nm in the range of 420-720 nm. We randomly choose three HSIs from the Harvard dataset used for experiments. We pick up the up-left $512 \times 512 \times 31$ block of each selected HSI as the GT. To obtain the simulated LR-HSI with the size of $32{\times}32{\times}31$, we downsample each of them by averaging the $16{\times}16$ disjoint spatial blocks. Besides, the three-band simulated HR-MSIs are generated by a Nikon D700 camera.

C. Parameters Discussion

The proposed method is mainly related to nine key parameters, i.e., the number of PAO iterations K, sparsity regularization parameter λ_1, smoothness regularization parameters λ_2, λ_3, λ_4, the weight of proximal term β, and the number of atoms of dictionaries n_1, n_2, n_3. Next, we discuss them in detail.

As reported in **Algorithm 5**, we take the proximal alternative optimization scheme to solve the problem (12). To evaluate the influence of the number of iterations K, we run the proposed method for different K. Fig. 4 shows the PSNR values of the reconstructed HSI of the Pavia University, CAVE, and Harvard dataset concerning different K. By Fig. 4, the PSNR for the Harvard dataset has a slight increase when K varies from 1 to 20, then has a fluctuation when K varies from 21 to 40, finally, remains stable for $K > 40$. For the CAVE dataset, the PSNR has a slow increase when K varies from 1 to 40 and remains stable for $K > 40$. For the Pavia University dataset, the PSNR has a sharp increase when K varies from 1 to 12, and then has a slight fluctuation when K varies from 13 to 50, finally, keeps stable for $K > 50$. Thus, we set the maximum number of iterations as 60 for the proposed algorithm.

Parameter β is the weight of proximal term in (13). To evaluate the influence of β, we run the proposed method for different β. Fig. 5 shows the PSNR values of the fused HSI of the Pavia University dataset for different $\log\beta$ (log is base 10). In this work, we set the range of $\log\beta$ to [-6, 0]. As we saw in Fig. 5, the PSNR increases in waves when $\log\beta$ varies from -6 to -2, remains stable when $\log\beta$ belongs to [-2, -1], and has a sharp drop when $\log\beta$ is greater than -1. Therefore, we set $\log\beta$ to -2, that is, β=0.01 for the Pavia University dataset. Similarly, the value of β for the CAVE and Harvard dataset could be determined in the same way.

The regularization parameter λ_1 in (12) controls the sparsity of the core tensor and, therefore, affects the estimation of HR-HSI. Higher values of λ_1 yield sparser core tensor. Fig. 6 shows the PSNR values of the reconstructed HSI of the Pavia University dataset to different $\log\lambda_1$. In this work, we set the range of $\log\lambda_1$ to [-9, -1]. As Fig. 6, the PSNR remains stable when $\log\lambda_1$ belongs to [-9, -5], and decreases sharply for $\log\lambda_1$ >-5. Therefore, we set $\log\lambda_1$ as -6, that is, $\lambda_1 = 10^{-6}$ for the Pavia University dataset. Similarly, the value of λ_1 for the CAVE and Harvard dataset could be determined in the same way.

The regularization parameter λ_2, λ_3, and λ_4 control the piecewise smoothness of the dictionary of width mode, height mode, and spectral mode, respectively. Higher values of λ_2, λ_3, and λ_4 yield smoother dictionaries. Fig. 7 shows the PSNR values of the reconstructed HSI of the Pavia University dataset under different $\log\lambda_2$, $\log\lambda_3$, and $\log\lambda_4$. In this work, we set the range of $\log\lambda_2$, $\log\lambda_3$, $\log\lambda_4$ all to [-9, 3]. As Fig. 7, the PSNR reaches the maximum value when $\log\lambda_2 = -8$, $\log\lambda_3 = -7$, $\log\lambda_4 = 3$. Therefore, we set $\log\lambda_2$ as -8, $\log\lambda_3$ as -7, $\log\lambda_4$ as 3, that is, $\lambda_2 = 10^{-8}$, $\lambda_3 = 10^{-7}$, and $\lambda_4 = 10^3$ for the Pavia University dataset. Note that, compared with λ_2 and λ_3, the best value of λ_4 is relatively large, that is because hyperspectral images are continuous in the spectral dimension, which leads that the total variation of the dictionary along the spectral direction could be small. Therefore, the best value of its regularization parameter should be relatively large. Similarly, the value of λ_2, λ_3, and λ_4 for the CAVE and Harvard dataset could be determined in the same way.

n_1, n_2, and n_3 are the number of dictionary atoms. Fig. 8 shows the PSNR values of the fused HSI of the Pavia University dataset with different n_1 and n_2 and Fig. 9 shows

TABLE II
THE DISCUSSION OF THE MAIN PARAMETERS

Parameters	Tuning Ranges	CAVE dataset [43]	Pavia University dataset [45]	Harvard dataset [44]	Suggested Ranges
K	[1, 60]	55	12	20	[12, 60]
β	[10^{-6}, 10^{0}]	10^{-2}	10^{-2}	10^{-4}	[10^{-4}, 10^{-2}]
λ_1	[10^{-9}, 10^{-1}]	10^{-7}	10^{-6}	10^{-4}	[10^{-7}, 10^{-4}]
λ_2	[10^{-9}, 10^{3}]	10^{-8}	10^{-7}	10^{-7}	[10^{-8}, 10^{-7}]
λ_3	[10^{-9}, 10^{3}]	10^{-4}	10^{-6}	10^{-4}	[10^{-6}, 10^{-4}]
λ_4	[10^{-9}, 10^{3}]	10^{2}	10^{3}	10^{2}	[10^{2}, 10^{3}]
n_1	[180, 320]	280	300	320	[280, 320]
n_2	[180, 320]	300	300	280	[280, 320]
n_3	[3, 21]	12	12	9	[9, 12]

the PSNR values of the fused HSI of the Pavia University dataset concerning different n_3. In this work, we set the range of n_1 and n_2 both to [180, 320], set n_3 as [3, 21]. The reason is that the spectral signatures of HSIs live on low dimensional subspaces. As Fig. 8, the PSNR has a sharp increase when n_1 varies within [180, 300], and reaches the maximum value as $n_1 = 300$. Similarly, the PSNR gets the maximum value at $n_2 = 300$. From Fig. 9, the PSNR decreases as $n_3 > 15$. Therefore, we set $n_1 = 300$, $n_2 = 300$, and $n_3 = 12$ for the Pavia University dataset. Note that the PSNR reaches the maximum value when n_3 is 15, however, we set it as 12 in our experiments, because we also need to consider the total performance of the other four quality indexes (i.e., ERGAS, SAM, DD, and RMSE). Similarly, the value of n_1, n_2, and n_3 for the CAVE and Harvard dataset could be determined in the same way.

In Table II, we present the tuning ranges of the nine main parameters, give each parameter value used for the three simulated HSI datasets mentioned in Section V-B, and also show the suggested ranges of each parameter to adjust the parameters conveniently.

Specifically, for the HySure [23], we set $\lambda_m = 1$ and $\lambda_\varphi = 5\times10^{-5}$. For the CSTF [12], according to the parameters used in [38], we set $n_1 = 260$, $n_2 = 260$, $n_3 = 15$, and $\lambda = 10^{-5}$. For CSU [28] and FUSE [29], we take the default optimal parameters used in their source codes.

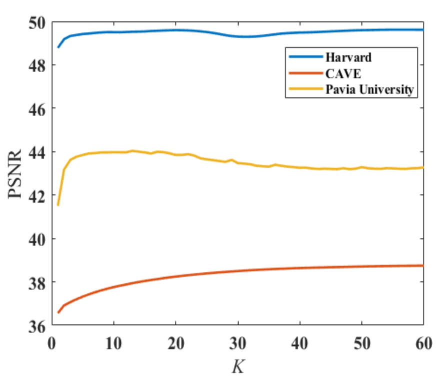


Fig. 4. The PSNR values concerning different K.

D. Quantitative Metrics

To evaluate the fused outputs from the numerical results, we utilize five metrics, namely the root mean square error (RMSE)

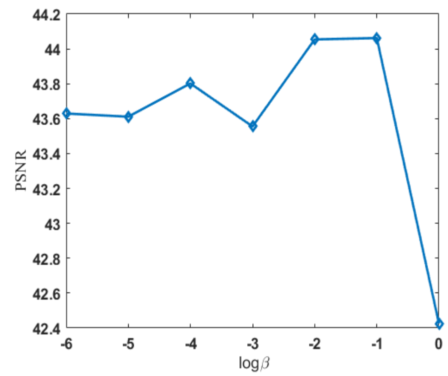


Fig. 5. The PSNR values for different $\log \beta$.

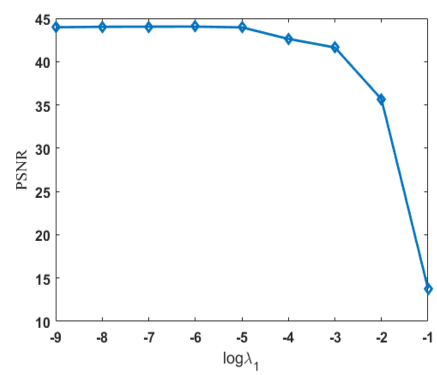


Fig. 6. The PSNR values to different $\log \lambda_1$.

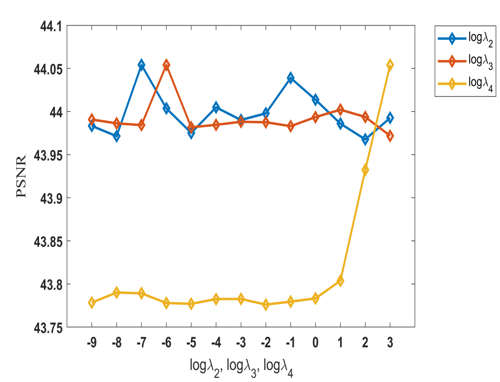


Fig. 7. The PSNR values under different $\log \lambda_2$, $\log \lambda_3$, and $\log \lambda_4$.

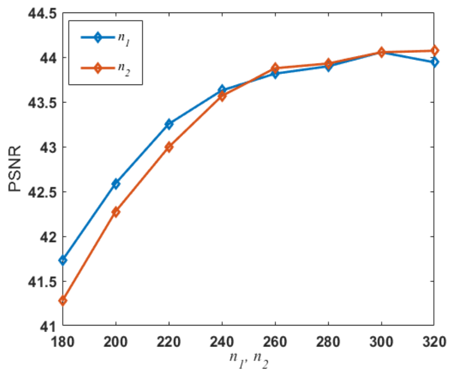


Fig. 8. The PSNR values with different n_1 and n_2.

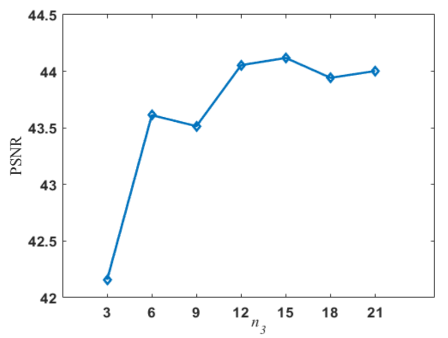


Fig. 9. The PSNR value concerning different n_3.

to estimate the error, the degree of distortion (DD) used to measure the spectral quality of the reconstructed outputs, the spectral angle mapper (SAM) used to reflect the degree of spectral distortions of the fused outputs, the peak signal to noise ratio (PSNR) and the relative dimensionless global error in synthesis (ERGAS) to measure the comprehensive quality of the fused results. All of them are defined as follows,

$$\begin{aligned} \mathrm{RMSE}(\mathcal{A},\widehat{\mathcal{A}}) &= \sqrt{\frac{||\mathcal{A}-\widehat{\mathcal{A}}||_F^2}{I_1I_2I_3}}, \\ \mathrm{DD}(\mathcal{A},\widehat{\mathcal{A}}) &= \frac{1}{I_1I_2I_3}||\mathrm{vec}(\mathcal{A})-\mathrm{vec}(\widehat{\mathcal{A}})||_1, \\ \mathrm{SAM}(\mathcal{A},\widehat{\mathcal{A}}) &= \frac{1}{I_1I_2}\sum_{i=1}^{I_1I_2}\mathrm{arcos}\frac{\widehat{\mathbf{a}}_i{}^T\mathbf{a}_i}{||\widehat{\mathbf{a}}_i||_2||\mathbf{a}_i||_2}, \\ \mathrm{PSNR}(\mathcal{A},\widehat{\mathcal{A}}) &= \frac{1}{I_3}\sum_{i=1}^{I_3}\mathrm{PSNR}(\mathbf{A}_i,\widehat{\mathbf{A}_i}), \\ \mathrm{ERGAS}(\mathcal{A},\widehat{\mathcal{A}}) &= \frac{100}{d}\sqrt{\frac{1}{I_3}\sum_{i=1}^{I_3}\frac{\mathrm{MSE}(\mathbf{A}_i,\widehat{\mathbf{A}_i})}{\mu^2_{(\widehat{\mathbf{A}}_i)}}}, \end{aligned} \tag{28}$$

where $\mathbf{A}_i$ and $\widehat{\mathbf{A}_i}$ denote the i-th band of $\mathcal{A}\in\mathbb{R}^{I_1\times I_2\times I_3}$ and reconstructed result $\widehat{\mathcal{A}}\in\mathbb{R}^{I_1\times I_2\times I_3}$, respectively; I_j, (j=1, 2, 3) represent the dimension of the j-th mode of $\mathcal{A}$; d denotes the spatial downsampling factor; MSE($\cdot$) and $\mu_{(\cdot)}$ denote the mean square error operator and the mean value operator, respectively.

Note that the best value of the four metrics (i.e., RMSE, SAM, DD, and ERGAS) is zero and of the metric PSNR is positive infinity.

TABLE III
QUALITY ASSESSMENT OF THE PROPOSED AND COMPARED METHODS ON THE PAVIA UNIVERSITY DATASET

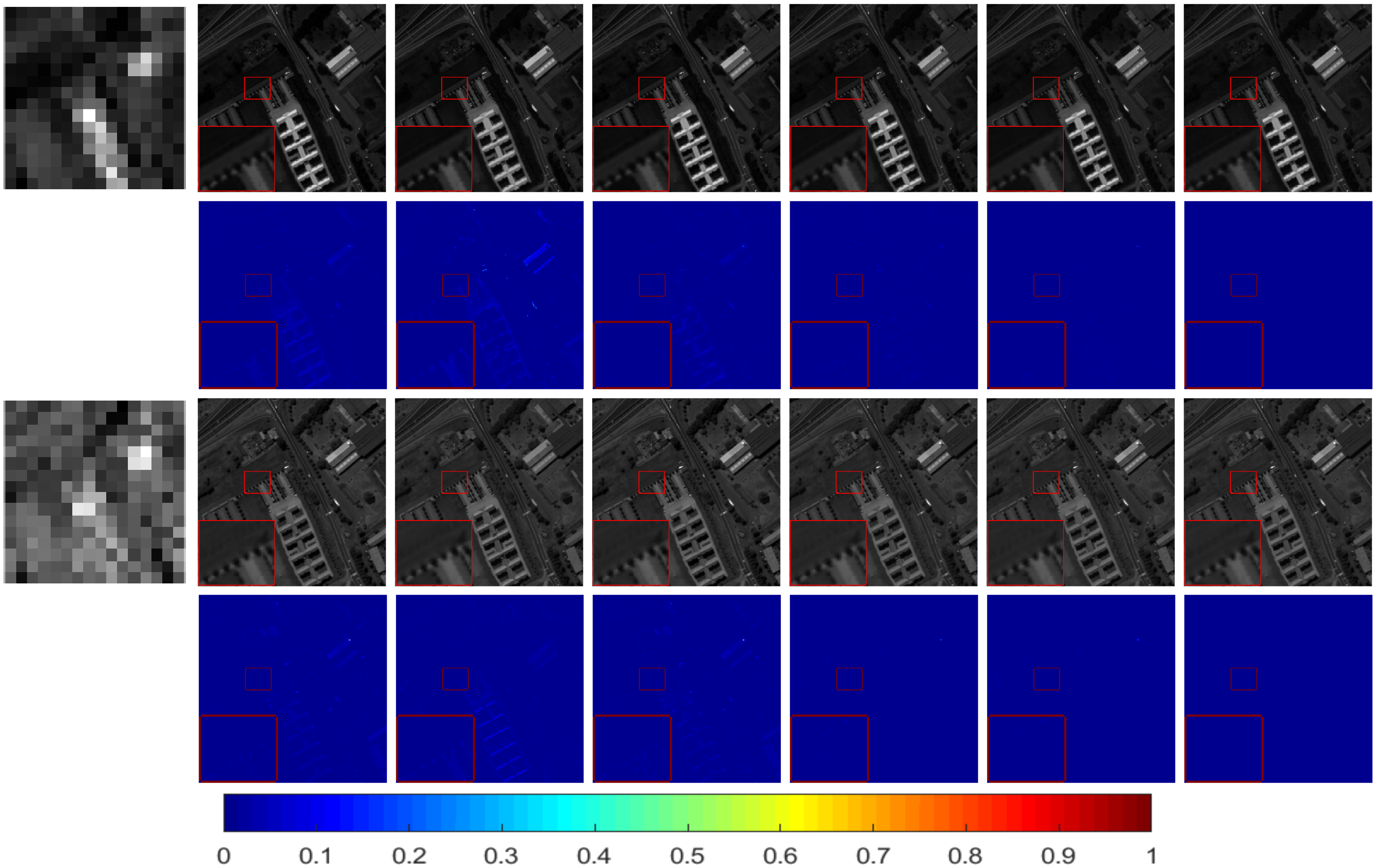
Method	Pavia University dataset [45]				
	RMSE	PSNR	DD	SAM	ERGAS
Best Values	0	$+\infty$	0	0	0
HySure [23]	3.243	38.344	2.208	2.838	1.841
FUSE [29]	3.128	38.73	2.115	2.862	1.768
CSU [28]	4.026	36.233	2.297	2.781	2.403
CSTF [12]	1.959	43.581	1.213	2.035	0.277
Proposed	**1.856**	**44.052**	**1.144**	**1.919**	**0.249**

E. Experimental Results

In this part, we test the proposed algorithm and the other four compared algorithms mentioned in Section V-A on three simulated HSI datasets noted in Section V-B. Table III presents the quality assessment of the five methods on the Pavia University dataset. The best values are highlighted by bold entries. From it, we could find that our algorithm has better performance than other compared algorithms in terms of five quality metrics mentioned in Section V-D. The reconstructed 5^{th} and 60^{th} bands and corresponding error images are shown in Fig. 10. From the 2nd and 4th row of Fig. 10, we could easily find that the fused results generated by the proposed approach and CSTF contain fewer errors than that by other compared methods. Besides, the spectral curves at different locations (i.e., (120, 120), (124, 124), (128,128)) are shown in Fig. 11, in which, we zoom some of the spectral bands for better comparison. We find that the spectral curve of HSI reconstructed by our algorithm match better with the ground truth (GT) compared with the other four algorithms.

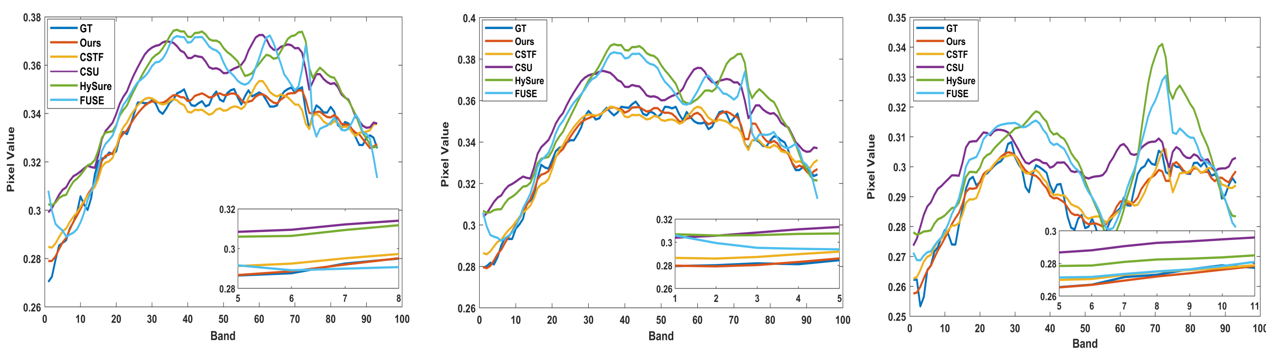
For the CAVE dataset, Table IV reports the average and standard deviation values of five quality metrics on the six reconstructed HSIs. The best values are emphasized by bold font. From it, we find that our approach has better performance than other compared approaches in all the five quality metrics. For visual comparison, the reconstructed 10^{th} band of *paints* (a HSI in CAVE dataset), the error images at the 10^{th} band, the reconstructed 29^{th} band, and the error images at the 29^{th} band are presented in Fig. 12. From the 2nd and 4th row of Fig. 12, we observe that the fused HSIs generated by the HySure, CSU, FUSE, and CSTF have more flaws than the proposed method. To describe the situation of spectral recovery, we show the spectral curves in Fig. 13 for *paints* at different locations (i.e., (256, 256), (361, 339), (366, 477)), in which, we zoom some of the spectral bands for better comparison. We could find that the spectral curve of HSI reconstructed by our algorithm match better with the ground truth (GT) compared with the other four algorithms.

For the Harvard dataset, Table V reports the average and standard deviation values of the five quality metrics on the three fused HSIs. The bold font represents the best values. From it, we find that our approach has better performance than other compared approaches in terms of four quality metrics except for ERGAS. For visual comparison, the reconstructed



(a) LR-HSI (b) HySure (c) CSU (d) FUSE (e) CSTF (f) Proposed (g) Ground Truth

Fig. 10. The 1st row presents the reconstructed results of the Pavia University for the 5^{th} band. The 2nd row presents the error results at the 5^{th} band. The 3rd row presents the reconstructed 60^{th} band. The 4th row presents the error results at the 60^{th} band. The 5th row shows the color bar. (a) LR-HSI, (b) HySure [23], (c) CSU [28], (d) FUSE [29], (e) CSTF [12], (f) Proposed, (g) Ground Truth.

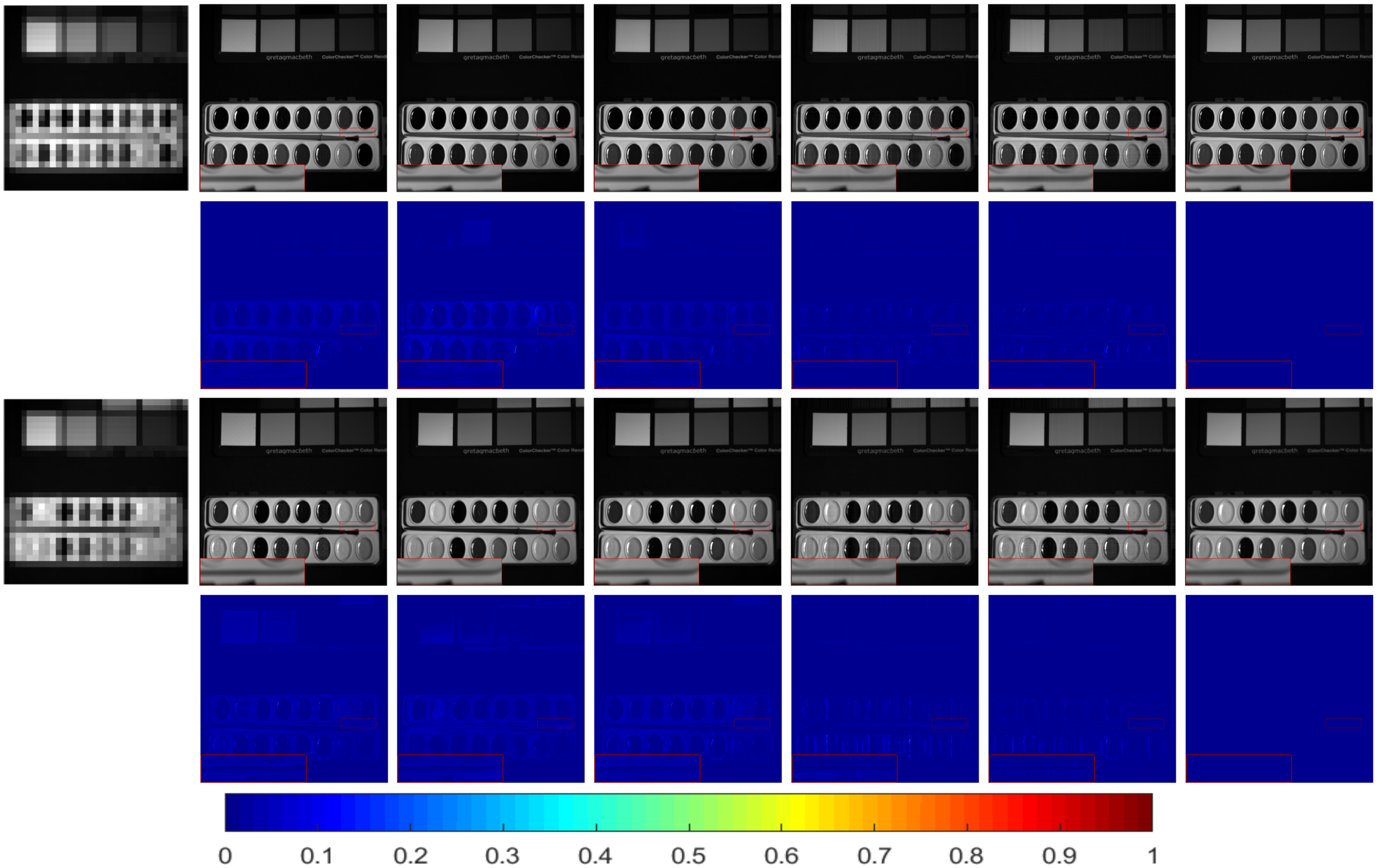


(a) (120, 120) (b) (124, 124) (c) (128, 128)

Fig. 11. The spectral curves between each fused output and GT (i.e., Pavia University) at different locations. Located at (a) (120, 120), (b) (124, 124), (c) (128, 128).

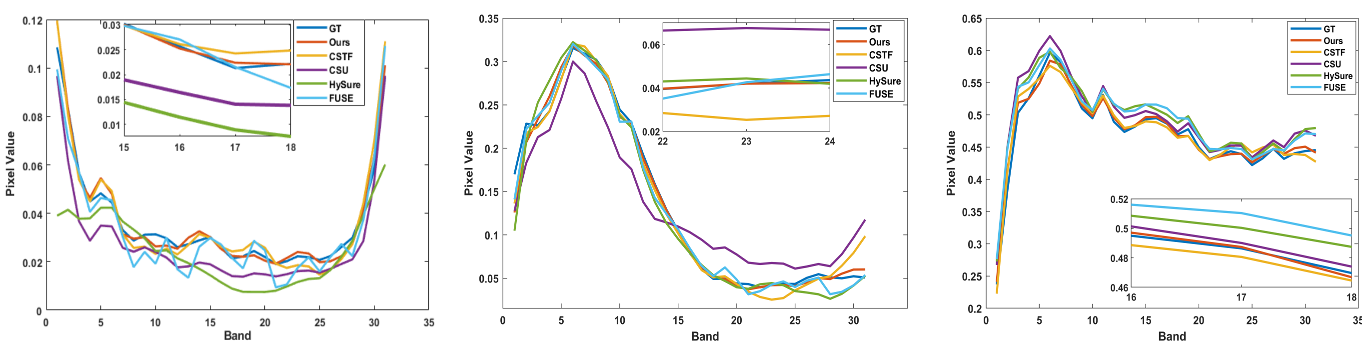
TABLE IV
QUALITY ASSESSMENT OF THE PROPOSED AND COMPARED METHODS ON THE CAVE DATASET

Method	CAVE dataset [43]				
	RMSE($\pm$std)	ERGAS($\pm$std)	DD($\pm$std)	SAM($\pm$std)	PSNR($\pm$std)
Best Values	0	0	0	0	$+\infty$
HySure [23]	3.322($\pm$1.238)	0.886($\pm$0.281)	1.976($\pm$1.011)	14.964($\pm$4.845)	39.512($\pm$3.085)
CSU [28]	3.306($\pm$1.378)	0.870($\pm$0.261)	1.624($\pm$0.943)	8.391($\pm$2.490)	39.001($\pm$3.656)
FUSE [29]	2.436($\pm$1.068)	0.625($\pm$0.141)	1.270($\pm$0.747)	8.122($\pm$1.476)	42.188($\pm$3.410)
CSTF [12]	2.722($\pm$1.487)	0.671($\pm$0.195)	1.351($\pm$0.790)	8.632($\pm$1.459)	42.240($\pm$4.027)
Proposed	**2.306($\pm$1.031)**	**0.590($\pm$0.136)**	**1.167($\pm$0.581)**	**7.934($\pm$1.388)**	**43.126($\pm$3.301)**



(a) LR-HSI (b) HySure (c) CSU (d) FUSE (e) CSTF (f) Proposed (g) Ground Truth

Fig. 12. The 1st row presents the results generated by the proposed and the other four compared methods for *paints* at the 10^{th} band. The 2nd row presents the error results at the 10^{th} band. The 3rd row presents the reconstructed 29^{th} band. The 4th row presents the error results at the 29^{th} band. (a) LR-HSI, (b) HySure [23], (c) CSU [28], (d) FUSE [29], (e) CSTF [12], (f) Proposed, (g) Ground Truth.



(a) (256, 256) (b) (361, 339) (c) (366, 477)

Fig. 13. The spectral curves between each fused result and GT (i.e., *paints*) at different locations. Located at (a) (256, 256), (b) (361,339), (c) (366,477).

TABLE V
QUALITY ASSESSMENT OF THE PROPOSED AND COMPARED METHODS ON THE HARVARD DATASET

Methods	Harvard dataset [44]				
	RMSE(±std)	ERGAS(±std)	SAM(±std)	DD(±std)	PSNR(±std)
Best Values	0	0	0	0	$+\infty$
HySure [23]	1.445(±0.634)	0.951(±0.566)	5.654(±1.434)	0.948(±0.499)	46.717(±2.982)
CSU [28]	1.639(±0.553)	0.866(±0.447)	4.553(±1.157)	0,844(±0.401)	45.669(±3.47)
FUSE [29]	1.179(±0.385)	0.760(±0.497)	4.330(±1.401)	0.741(±0.341)	47.973(±2.686)
CSTF [12]	1.184(±0.532)	**0.740(±0.480)**	4.460(±1.156)	0.758(±0.429)	48.141(±3.342)
Proposed	**1.142(±0.479)**	0.745(±0.486)	**4.425(±1.254)**	**0.735(±0.398)**	**48.262(±3.201)**

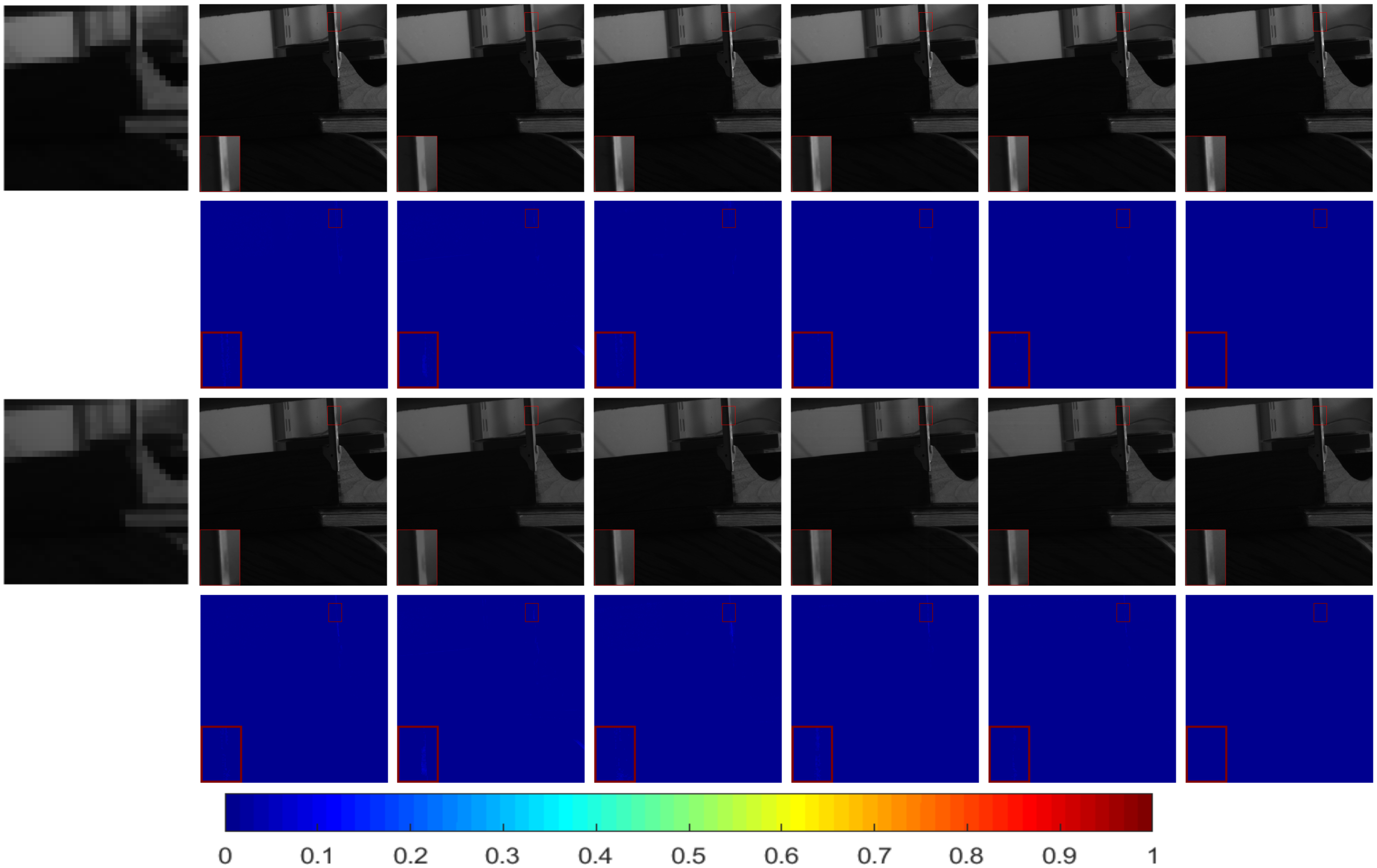


Fig. 14. The 1st row presents the results generated by the proposed and the other four compared methods for a HSI in the Harvard dataset at the 25^{th} band. The 2nd row presents the error results at the 25^{th} band. The 3rd row presents the reconstructed 31^{th} band. The 4th row presents the error results at the 31^{th} band. (a) LR-HSI (b) HySure [23], (c) CSU [28], (d) FUSE [29], (e) CSTF [12], (f) Proposed, (g) Ground Truth.

TABLE VI
QUALITY ASSESSMENT OF THE THREE REGULARIZATION TERMS

Method	CAVE dataset [43]				
	RMSE	PSNR	DD	SAM	ERGAS
Best Values	0	$+\infty$	0	0	0
Without $\lambda_2\|\|\mathbf{D_y U_1}\|\|_1$	2.358($\pm$1.111)	43.109($\pm$3.317)	1.186($\pm$0.609)	8.064($\pm$1.240)	0.599($\pm$0.127)
Without $\lambda_3\|\|\mathbf{D_y U_2}\|\|_1$	2.460($\pm$1.207)	42.723($\pm$3.735)	1.242($\pm$0.668)	8.142($\pm$1.460)	0.618($\pm$0.149)
Without $\lambda_4\|\|\mathbf{D_y U_3}\|\|_1$	2.316($\pm$1.031)	43.076($\pm$3.291)	1.175($\pm$0.582)	7.979($\pm$1.365)	0.593($\pm$0.138)
CSTF [12]	2.722($\pm$1.487)	42.240($\pm$4.027)	1.351($\pm$0.790)	8.632($\pm$1.459)	0.671($\pm$0.195)
Proposed	**2.306($\pm$1.031)**	**43.126($\pm$3.301)**	**1.167($\pm$0.581)**	**7.934($\pm$1.388)**	**0.590($\pm$0.136)**

25^{th} band of a HSI in the Harvard dataset, the error images at the 25^{th} band, the reconstructed 31^{th} band, and the error images at the 31^{th} band are presented in Fig. 14. From the 2nd and 4th row of Fig. 14, we observe that the fused HSIs generated by the HySure, CSU, FUSE, and CSTF have more flaws than the proposed method.

F. Discussion

1) The effectiveness of the three regularization terms: In this part, we discuss the effectiveness of the three regularization terms (i.e., $\lambda_2||\mathbf{D_y U_1}||_1$, $\lambda_3||\mathbf{D_y U_2}||_1$ and $\lambda_4||\mathbf{D_y U_3}||_1$). In the proposed model (12), we use the three regularization terms to character the piecewise smoothness of the target HR-HSI. How do the three regularization terms individually affect the fusion results? Are among the components essential in the proposed approach? In this part, we will consider three variants of our model: $\lambda_2||\mathbf{D_y U_1}||_1$, $\lambda_3||\mathbf{D_y U_2}||_1$ and $\lambda_4||\mathbf{D_y U_3}||_1$. We set λ_2 equal to zero to demonstrate the effectiveness of $||\mathbf{D_y U_1}||_1$, set λ_3 equal to zero to demonstrate the effectiveness of $||\mathbf{D_y U_2}||_1$, and λ_4 equal to zero to demonstrate the effectiveness of $||\mathbf{D_y U_3}||_1$, it is worth note that the proposed model degenerates to Dian et al. [12] when we set λ_2, λ_3, and λ_4 equal to zero at the same time. We test the above five situations on six HSIs from CAVE dataset mentioned in Section V-B, Table VI reports the quality assessment of the fused results. The bold entries denote the best values. From it, we observe that each regularization term has positive effect on the fused outputs, especially, when they combine together to form the proposed method, which gives excellent performance.

2) The spatial downsampling ways and factors: In experiments, we use the same spatial downsampling ways and factors for both CAVE dataset and Pavia University dataset, i.e., downsampling factor 16 with the average filter. In this part, we test the proposed approach and LTMR [38] by different downsampling ways and different downsampling factors on the Pavia University dataset noted in Section V-B. For downsampling ways, we choose the average filter and Gaussian filter with 7×7 and standard deviation 2. The downsampling factors are set as 4, 8, 16, and 32, respectively. Table VII and Table VIII report the quality assessment for the average filter and Gaussian filter, respectively, with different downsampling factors. The best values are highlighted by bold entries. From them, we can observe that LTMR and the proposed method are both stable for different downsampling ways. However, each value of the quality metrics for LTMR is changed dramatically as the downsampling factor increasing, while the proposed method always keeps relative stability. Just to make it a little bit more intuitive, we show the variation trend of PSNR of the two methods (i.e., the proposed approach and LTMR [38]) under different downsampling factors and filters in Fig. 15.

3) The convergence of the proposed model: In this work, we utilize the PAO algorithm to deal with the proposed model (12), in which each subproblem is solved by the ADMM. In this part, we further verify the convergence of the proposed model in (12) and four subproblem in (13). Fig. 16 shows the corresponding convergence curves, in which the X-axis represents the number of iterations (for the problem (12), which means the number of PAO iterations; for the four subproblems in (13), which means the number of ADMM iterations.), and the Y-axis denotes the value of $RelCha$. From Fig. 16, we can see that not only the proposed model is convergent, but also each subproblem is convergent.

TABLE VII
AVERAGE FILTER WITH DIFFERENT DOWNSAMPLING FACTOR (DF).

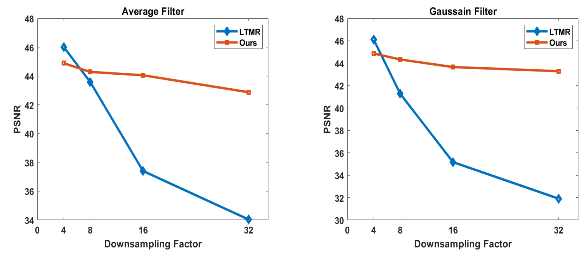
Method	Average Filter				
	PSNR	SAM	DD	ERGAS	DF
Best Values	$+\infty$	0	0	0	
LTMR [38]	**46.006**	**1.422**	**0.814**	**0.793**	4
	43.585	2.084	1.202	0.541	8
	37.405	4.725	2.857	0.636	16
	34.035	6.796	4.419	0.469	32
Ours	44.887	1.682	0.993	0.907	4
	44.283	**1.836**	**1.097**	**0.486**	8
	44.052	**1.919**	**1.144**	**0.249**	16
	42.873	**2.224**	**1.366**	**0.145**	32

VI. CONCLUSION

In this work, we have presented a novel unidirectional total variation based approach for the HSI-SR problem. We first consider that the target HR-HSIs exhibit both the sparsity and the piecewise smoothness on the three modes and then utilize the classical Tucker decomposition to decompose the target HR-HSI as a sparse core tensor multiplied by the dictionary matrices along with the three modes. In the proposed model, we apply the unidirectional TV on three dictionaries to characterize the piecewise smoothness, and the ℓ_1-norm on core tensor to characterize the sparsity. Experiments on three public HSI datasets illustrate the superiority of the proposed approach over some state-of-the-art approaches, and also emphasize the robust performance on the downsampling ways and downsampling factors. Especially, the proposed approach also has one unavoidable weakness that is its computation. For the solution of the proposed model under the framework of PAO, each subproblem has to be addressed by an inner iterative approach, i.e., ADMM, which increases the computation and the number of parameters. In the future, we will try to find a closed-form solution for each subproblem even though it is very challenging, aiming to reduce or overcome the mentioned weakness.

TABLE VIII
GAUSSIAN FILTER WITH DIFFERENT DOWNSAMPLING FACTOR (DF).

Method	Gaussian Filter				
	PSNR	SAM	DD	ERGAS	DF
Best Values	$+\infty$	0	0	0	
LTMR [38]	**46.089**	**1.426**	**0.816**	**0.781**	4
	41.274	2.594	1.517	0.735	8
	35.167	5.450	3.595	0.816	16
	31.908	8.197	5.544	0.611	32
Ours	44.866	1.686	0.996	0.902	4
	44.332	**1.797**	**1.075**	**0.484**	8
	43.655	**1.972**	**1.195**	**0.263**	16
	43.284	**2.083**	**1.276**	**0.137**	32



(a) Average Filter (b) Gaussian Filter

Fig. 15. The variation trend of PSNR of the proposed method and LTMR [38] under different downsampling factors and filters. (a) Average Filter, (b) Gaussian Filter.

VII. APPENDIX

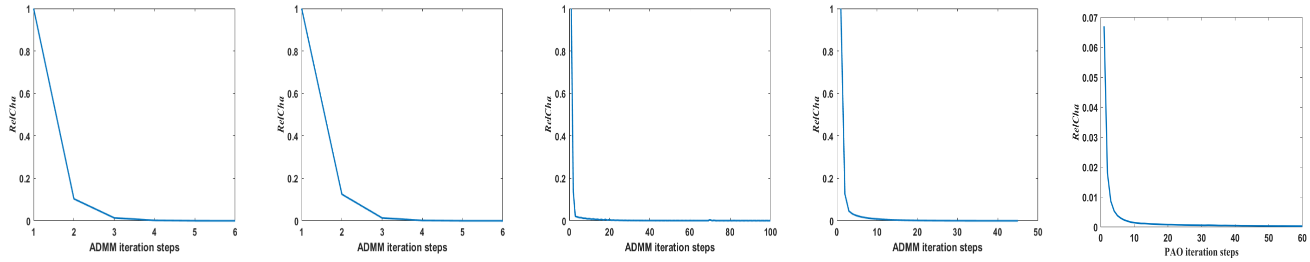
A. The optimization problem of $\mathbf{U_1}$

Problem (15) is convex and can be solved by ADMM efficiently [48], [49]. Hence, we introduce the splitting variable $\mathbf{V_1} = \mathbf{D_y U_1}$ and then rewrite the problem (15) as follows,

$$
\begin{aligned}
\underset{\mathbf{U_1}}{\operatorname{argmin}} \quad & ||\mathbf{B}_{(1)} - \mathbf{D_1 U_1 X_1}||_F^2 + ||\mathbf{C}_{(1)} - \mathbf{U_1 Y_1}||_F^2 \\
& + \lambda_2 ||\mathbf{V_1}||_1 + \beta ||\mathbf{U_1} - \mathbf{U_1}^{pre}||_F^2, \\
& s.t. \mathbf{V_1} = \mathbf{D_y U_1}.
\end{aligned} \tag{29}
$$

The augmented Lagrangian form of equation (29) is represented by

$$
\begin{aligned}
L(\mathbf{U_1}, \mathbf{V_1}, \mathbf{L_1}) = & ||\mathbf{B}_{(1)} - \mathbf{D_1 U_1 X_1}||_F^2 \\
& + ||\mathbf{C}_{(1)} - \mathbf{U_1 Y_1}||_F^2 \\
& + \lambda_2 ||\mathbf{V_1}||_1 + \beta ||\mathbf{U_1} - \mathbf{U_1}^{pre}||_F^2 \\
& + \mu_1 ||\mathbf{D_y U_1} - \mathbf{V_1} - \mathbf{L_1}||_F^2,
\end{aligned} \tag{30}
$$



(a) $\mathbf{U_1}$-subproblem (b) $\mathbf{U_2}$-subproblem (c) $\mathbf{U_3}$-subproblem (d) $\mathcal{G}$-subproblem (e) Problem(12)

Fig. 16. The convergence curve of the problem (12) and four subproblems. (a) $\mathbf{U_1}$-subproblem (14), (b) $\mathbf{U_2}$-subproblem (16), (c) $\mathbf{U_3}$-subproblem (18), (d) $\mathcal{G}$-subproblem (20), (e) Problem (12).

where $\mathbf{L_1}$ denotes the Lagrangian multiplier and μ_1 represents a positive penalty parameter.

ADMM iterations of problem (30) are given by

$$\begin{cases} \mathbf{U_1}^{(t+1)} = \underset{\mathbf{U_1}}{\operatorname{argmin}}\, L(\mathbf{U_1}, \mathbf{V_1}^{(t)}, \mathbf{L_1}^{(t)}), \\ \mathbf{V_1}^{(t+1)} = \underset{\mathbf{V_1}}{\operatorname{argmin}}\, L(\mathbf{U_1}^{(t+1)}, \mathbf{V_1}, \mathbf{L_1}^{(t)}), \\ \mathbf{L_1}^{(t+1)} = \underset{\mathbf{L_1}}{\operatorname{argmin}}\, L(\mathbf{U_1}^{(t+1)}, \mathbf{V_1}^{(t+1)}, \mathbf{L_1}). \end{cases} \tag{31}$$

Next, we present the solving process of (31).

1) The $\mathbf{U_1}$-subproblem: From (30), we have

$$\underset{\mathbf{U_1}}{\operatorname{argmin}} \quad ||\mathbf{B}_{(1)} - \mathbf{D_1 U_1 X_1}||_F^2 + ||\mathbf{C}_{(1)} - \mathbf{U_1 Y_1}||_F^2 + \beta||\mathbf{U_1} - \mathbf{U_1}^{pre}||_F^2 + \mu_1||\mathbf{D_y U_1} - \mathbf{V_1} - \mathbf{L_1}||_F^2. \tag{32}$$

Problem (32) is a quadratic optimization, which has a unique solution, and it amounts to compute the following Sylvester matrix equation, i.e.,

$$\begin{aligned} &\mathbf{D_1}^T\mathbf{D_1 U_1 X_1 X_1}^T + \mathbf{U_1}(\mathbf{Y_1 Y_1}^T + \beta\mathbf{I}) + \mu_1\mathbf{D_y}^T\mathbf{D_y U_1} \\ &= \mathbf{D_1}^T\mathbf{B}_{(1)}\mathbf{X_1}^T + \mathbf{C}_{(1)}\mathbf{Y_1}^T + \beta\mathbf{U_1}^{pre} \\ &+ \mu_1\mathbf{D_y}^T(\mathbf{V_1} + \mathbf{L_1}). \end{aligned} \tag{33}$$

We apply the conjugate gradient (CG) [50] to solving (33) efficiently. Similar to Li et al.'s work [27], we give the following remarks: 1) the system matrix associated with (33) is symmetric and positive definite, which is a necessary and sufficient condition to directly apply CG; 2) the step of the heaviest computation in applying CG is the multiplication of the system matrix times a vector, which can be carried out very efficiently in the matrix representation, and finally, 3) CG converges in only a few iterations. In our experiments, we have systematically observed that 40 iterations have a pretty good approximation of the solution of (33).

For the problem (33), during each iteration of CG, the heaviest computational complexity is $O(n_1^2 I_1^2)$, which could be reduced to $O(n_1^2 I_1 + n_1 I_1^2)$ due to the matrix representation.

2) The $\mathbf{V_1}$-subproblem: From (30), we have

$$\underset{\mathbf{V_1}}{\operatorname{argmin}}\, \lambda_2||\mathbf{V_1}||_1 + \mu_1||\mathbf{D_y U_1} - \mathbf{V_1} - \mathbf{L_1}||_F^2, \tag{34}$$

whose solution $\mathbf{V_1}$ could be computed by the following column-wise vector-soft threshold function, i.e.,

$$\mathbf{V_1} = \operatorname{soft}(\mathbf{D_y U_1} - \mathbf{L_1}, \frac{\lambda_2}{2\mu_1}), \tag{35}$$

where $\operatorname{soft}(x, y) = \operatorname{sign}(x) * \max(|x| - y, 0)$.

The computational complexity of updating $\mathbf{V_1}$ by (35) is $O(n_1(I_1 - 1)(I_1 + 1))$.

3) The $\mathbf{L_1}$-subproblem: From (30), we update the Lagrangian multiplier $\mathbf{L_1}$ by

$$\mathbf{L_1} = \mathbf{L_1} - (\mathbf{D_y U_1} - \mathbf{V_1}). \tag{36}$$

The computational complexity of updating $\mathbf{L_1}$ by (36) is $O(n_1 I_1 (I_1 - 1))$.

During each iteration of ADMM, the heaviest computation steps, shown in (33), have complexity $O(N_{CG}(n_1^2 I_1 + n_1 I_1^2))$, where N_{CG} is the iteration number of CG.

In **Algorithm 1**, we summarize the process of solving $\mathbf{U_1}$-subproblem (14) by ADMM.

Algorithm 1 Solving $\mathbf{U_1}$-subproblem (14) via ADMM

Inputting: $\mathcal{B}$, $\mathcal{C}$, $\mathbf{U_2}$, $\widehat{\mathbf{U_2}}$, $\mathbf{U_3}$, $\widehat{\mathbf{U_3}}$, $\mathcal{G}$, $\mathbf{U_1}^{pre}$, $\beta > 0$, $\mu_1 > 0$, and $\lambda_2 > 0$.
Outputting: Dictionary matrix $\mathbf{U_1}$.
While not converged **do**
  Step 1 Updating the dictionary matrix $\mathbf{U_1}$ by (33);
  Step 2 Updating the variable $\mathbf{V_1}$ by (35);
  Step 3 Updating the Lagrangian multiplier $\mathbf{L_1}$ by (36);
end while

B. *The optimization problem of* $\mathbf{U_2}$

Similar to problem (15), problem (17) is convex, therefore, we use ADMM to solve it efficiently. We first introduce the splitting variable $\mathbf{V_2} = \mathbf{D_y U_2}$ and then rewrite the problem (17) as follows,

$$\begin{aligned} \underset{\mathbf{U_2}}{\operatorname{argmin}} \quad & ||\mathbf{B}_{(2)} - \mathbf{D_2 U_2 X_2}||_F^2 + ||\mathbf{C}_{(2)} - \mathbf{U_2 Y_2}||_F^2 \\ & + \lambda_3||\mathbf{V_2}||_1 + \beta||\mathbf{U_2} - \mathbf{U_2}^{pre}||_F^2, \\ & s.t. \mathbf{V_2} = \mathbf{D_y U_2}. \end{aligned} \tag{37}$$

The augmented Lagrangian form of equation (37) is represented by

$$\begin{aligned} L(\mathbf{U_2}, \mathbf{V_2}, \mathbf{L_2}) = & ||\mathbf{B}_{(2)} - \mathbf{D_2 U_2 X_2}||_F^2 \\ & + ||\mathbf{C}_{(2)} - \mathbf{U_2 Y_2}||_F^2 \\ & + \lambda_3||\mathbf{V_2}||_1 + \beta||\mathbf{U_2} - \mathbf{U_2}^{pre}||_F^2 \\ & + \mu_2||\mathbf{D_y U_2} - \mathbf{V_2} - \mathbf{L_2}||_F^2, \end{aligned} \tag{38}$$

where $\mathbf{L_2}$ denotes the Lagrangian multiplier and $\mu_2 > 0$ represents a penalty parameter.

ADMM iterations of problem (38) are given by

$$\begin{cases} \mathbf{U_2}^{(t+1)} = \underset{\mathbf{U_2}}{\text{argmin}}\, L(\mathbf{U_2}, \mathbf{V_2}^{(t)}, \mathbf{L_2}^{(t)}), \\ \mathbf{V_2}^{(t+1)} = \underset{\mathbf{V_2}}{\text{argmin}}\, L(\mathbf{U_2}^{(t+1)}, \mathbf{V_2}, \mathbf{L_2}^{(t)}), \\ \mathbf{L_2}^{(t+1)} = \underset{\mathbf{L_2}}{\text{argmin}}\, L(\mathbf{U_2}^{(t+1)}, \mathbf{V_2}^{(t+1)}, \mathbf{L_2}). \end{cases} \tag{39}$$

Next, we present the solving process of (39).

1) The $\mathbf{U_2}$-subproblem: From (38), we have

$$\begin{aligned} &\underset{\mathbf{U_2}}{\text{argmin}}\, ||\mathbf{B}_{(2)} - \mathbf{D_2U_2X_2}||_F^2 + ||\mathbf{C}_{(2)} - \mathbf{U_2Y_2}||_F^2 + \\ &\beta||\mathbf{U_2} - \mathbf{U_2}^{pre}||_F^2 + \mu_2||\mathbf{D_y U_2} - \mathbf{V_2} - \mathbf{L_2}||_F^2. \end{aligned} \tag{40}$$

Similar to problem (32), the solution of problem (40) amounts to compute the following Sylvester matrix equation, i.e.,

$$\begin{aligned} &\mathbf{D_2}^T\mathbf{D_2U_2X_2X_2}^T + \mathbf{U_2}(\mathbf{Y_2Y_2}^T + \beta\mathbf{I}) + \mu_2\mathbf{D_y}^T\mathbf{D_yU_2} \\ &= \mathbf{D_2}^T\mathbf{B}_{(2)}\mathbf{X_2}^T + \mathbf{C}_{(2)}\mathbf{Y_2}^T + \beta\mathbf{U_2}^{pre} \\ &+ \mu_2\mathbf{D_y}^T(\mathbf{V_2} + \mathbf{L_2}). \end{aligned} \tag{41}$$

Similarly, we utilize CG to solve (41). During each iteration of CG, similar to (33), the heaviest computational complexity is $O(n_2^2 I_2^2)$, which could be reduced to $O(n_2^2 I_2 + n_2 I_2^2)$ due to the matrix representation.

2) The $\mathbf{V_2}$-subproblem: From (38), we have

$$\underset{\mathbf{V_2}}{\text{argmin}}\, \lambda_3||\mathbf{V_2}||_1 + \mu_2||\mathbf{D_yU_2} - \mathbf{V_2} - \mathbf{L_2}||_F^2, \tag{42}$$

whose solution $\mathbf{V_2}$ could be given by the following column-wise vector-soft threshold function, i.e.,

$$\mathbf{V_2} = \text{soft}(\mathbf{D_yU_2} - \mathbf{L_2}, \frac{\lambda_3}{2\mu_2}), \tag{43}$$

where the definition of soft(x, y) is the same as that in (35).

The computational complexity of updating $\mathbf{V_2}$ by (43) is $O(n_2(I_2 - 1)(I_2 + 1))$.

3) The $\mathbf{L_2}$-subproblem: From (38), we update the Lagrangian multiplier $\mathbf{L_2}$ by

$$\mathbf{L_2} = \mathbf{L_2} - (\mathbf{D_yU_2} - \mathbf{V_2}). \tag{44}$$

The computational complexity of updating $\mathbf{L_2}$ by (44) is $O(n_2 I_2(I_2 - 1))$.

During each iteration of ADMM, the heaviest computation steps, shown in (41), have complexity $O(N_{CG}(n_2^2 I_2 + n_2 I_2^2))$, where N_{CG} is the iteration number of CG.

In **Algorithm 2**, we summarize the process of solving $\mathbf{U_2}$-subproblem (16) by ADMM.

Algorithm 2 Solving $\mathbf{U_2}$-subproblem (16) via ADMM

Inputting: $\mathcal{B}$, $\mathcal{C}$, $\mathbf{U_1}$, $\widehat{\mathbf{U_1}}$, $\mathbf{U_3}$, $\widehat{\mathbf{U_3}}$, $\mathcal{G}$, $\mathbf{U_2}^{pre}$, $\beta > 0$, $\mu_2 > 0$, and $\lambda_3 > 0$.
Outputting: Dictionary matrix $\mathbf{U_2}$.
While not converged **do**
  Step 1 Updating the dictionary matrix $\mathbf{U_2}$ by (41);
  Step 2 Updating the variable $\mathbf{V_2}$ by (43);
  Step 3 Updating the Lagrangian multiplier $\mathbf{L_2}$ by (44);
end while

C. *The optimization problem of* $\mathbf{U_3}$

Similar to problem (15), problem (19) is convex, therefore, we use ADMM to solve it efficiently. We first introduce the splitting variable $\mathbf{V_3} = \mathbf{D_yU_3}$ and then rewrite the problem (19) as follows,

$$\begin{aligned} \underset{\mathbf{U_3}}{\text{argmin}}\quad &||\mathbf{B}_{(3)} - \mathbf{U_3X_3}||_F^2 + ||\mathbf{C}_{(3)} - \mathbf{D_3U_3Y_3}||_F^2 \\ &+ \lambda_4||\mathbf{V_3}||_1 + \beta||\mathbf{U_3} - \mathbf{U_3}^{pre}||_F^2, \\ &s.t. \mathbf{V_3} = \mathbf{D_yU_3}. \end{aligned} \tag{45}$$

The augmented Lagrangian form of equation (45) is given by

$$\begin{aligned} L(\mathbf{U_3}, \mathbf{V_3}, \mathbf{L_3}) = &||\mathbf{B}_{(3)} - \mathbf{U_3X_3}||_F^2 + ||\mathbf{C}_{(3)} - \mathbf{D_3U_3Y_3}||_F^2 \\ &+ \lambda_4||\mathbf{V_3}||_1 + \beta||\mathbf{U_3} - \mathbf{U_3}^{pre}||_F^2 \\ &+ \mu_3||\mathbf{D_yU_3} - \mathbf{V_3} - \mathbf{L_3}||_F^2, \end{aligned} \tag{46}$$

where $\mathbf{L_3}$ denotes the Lagrangian multiplier and $\mu_3 > 0$ represents a penalty parameter.

ADMM iterations of problem (46) are given by

$$\begin{cases} \mathbf{U_3}^{(t+1)} = \underset{\mathbf{U_3}}{\text{argmin}}\, L(\mathbf{U_3}, \mathbf{V_3}^{(t)}, \mathbf{L_3}^{(t)}), \\ \mathbf{V_3}^{(t+1)} = \underset{\mathbf{V_3}}{\text{argmin}}\, L(\mathbf{U_3}^{(t+1)}, \mathbf{V_3}, \mathbf{L_3}^{(t)}), \\ \mathbf{L_3}^{(t+1)} = \underset{\mathbf{L_3}}{\text{argmin}}\, L(\mathbf{U_3}^{(t+1)}, \mathbf{V_3}^{(t+1)}, \mathbf{L_3}). \end{cases} \tag{47}$$

Next, we present the solving process of (47).

1) The $\mathbf{U_3}$-subproblem: From (46), we have

$$\begin{aligned} &\underset{\mathbf{U_3}}{\text{argmin}}\quad ||\mathbf{B}_{(3)} - \mathbf{U_3X_3}||_F^2 + ||\mathbf{C}_{(3)} - \mathbf{D_3U_3Y_3}||_F^2 + \\ &\beta||\mathbf{U_3} - \mathbf{U_3}^{pre}||_F^2 + \mu_3||\mathbf{D_yU_3} - \mathbf{V_3} - \mathbf{L_3}||_F^2. \end{aligned} \tag{48}$$

Similar to (32), the solution of problem (48) amounts to compute the following Sylvester matrix equation, i.e.,

$$\begin{aligned} &\mathbf{D_3}^T\mathbf{D_3U_3Y_3Y_3}^T + \mathbf{U_3}(\mathbf{X_3X_3}^T + \beta\mathbf{I}) + \mu_3\mathbf{D_y}^T\mathbf{D_yU_3} \\ &= \mathbf{D_3}^T\mathbf{C}_{(3)}\mathbf{Y_3}^T + \mathbf{B}_{(3)}\mathbf{X_3}^T + \beta\mathbf{U_3}^{pre} \\ &+ \mu_3\mathbf{D_y}^T(\mathbf{V_3} + \mathbf{L_3}). \end{aligned} \tag{49}$$

Similarly, we utilize CG to solve (49). During each iteration of CG, similar to (33), the heaviest computational complexity is $O(n_3^2 I_3^2)$, which could be reduced to $O(n_3^2 I_3 + n_3 I_3^2)$ due to the matrix representation.

2) The $\mathbf{V_3}$-subproblem: From (46), we have

$$\underset{\mathbf{V_3}}{\text{argmin}}\quad \lambda_4||\mathbf{V_3}||_1 + \mu_3||\mathbf{D_yU_3} - \mathbf{V_3} - \mathbf{L_3}||_F^2, \tag{50}$$

whose solution $\mathbf{V_3}$ could be given by the following column-wise vector-soft threshold function, i.e.,

$$\mathbf{V_3} = \text{soft}(\mathbf{D_yU_3} - \mathbf{L_3}, \frac{\lambda_4}{2\mu_3}), \tag{51}$$

where the definition of soft(x, y) is the same as that in (35).

The computational complexity of updating $\mathbf{V_3}$ by (51) is $O(n_3(I_3 - 1)(I_3 + 1))$.

3) The $\mathbf{L_3}$-subproblem: From (46), we update the Lagrangian multiplier $\mathbf{L_3}$ by

$$\mathbf{L_3} = \mathbf{L_3} - (\mathbf{D_y U_3} - \mathbf{V_3}). \tag{52}$$

The computational complexity of updating $\mathbf{L_3}$ by (52) is $O(n_3 I_3 (I_3 - 1))$.

During each iteration of ADMM, the heaviest computation steps, shown in (49), have complexity $O(N_{CG}(n_3^2 I_3 + n_3 I_3^2))$, where N_{CG} is the iteration number of CG.

In **Algorithm 3**, we summarize the process of solving $\mathbf{U_3}$-subproblem (18) by ADMM.

Algorithm 3 Solving $\mathbf{U_3}$-subproblem (18) via ADMM

Inputting: $\mathcal{B}$, $\mathcal{C}$, $\mathbf{U_1}$, $\widehat{\mathbf{U_1}}$, $\mathbf{U_2}$, $\widehat{\mathbf{U_2}}$, $\mathcal{G}$, $\mathbf{U_3}^{pre}$, $\beta > 0$, $\mu_3 > 0$, and $\lambda_4 > 0$.
Outputting: Dictionary matrix $\mathbf{U_3}$.
While not converged **do**
  Step 1 Updating the dictionary matrix $\mathbf{U_3}$ by (49);
  Step 2 Updating the variable $\mathbf{V_3}$ by (51);
  Step 3 Updating the Lagrangian multiplier $\mathbf{L_3}$ by (52);
end while

D. The optimization problem of $\mathcal{G}$

Problem (20) is convex, hence, we utilize ADMM to solve it. We first introduce the splitting variables $\mathcal{G}_1 = \mathcal{G}$ and $\mathcal{G}_2 = \mathcal{G}$ and then rewrite the problem (20) as follows,

$$\begin{aligned} &\underset{\mathcal{G},\mathcal{G}_1,\mathcal{G}_2}{\text{argmin}}\, f(\mathcal{G}) + f_1(\mathcal{G}_1) + f_2(\mathcal{G}_2), \\ &\quad s.t.\ \mathcal{G}_1 = \mathcal{G},\ \mathcal{G}_2 = \mathcal{G}, \end{aligned} \tag{53}$$

where

$$\begin{aligned} f(\mathcal{G}) &= \lambda_1 ||\mathcal{G}||_1 + \beta ||\mathcal{G} - \mathcal{G}^{pre}||_F^2, \\ f_1(\mathcal{G}_1) &= ||\mathcal{B} - \mathcal{G}_1 \times_1 \widehat{\mathbf{U_1}} \times_2 \widehat{\mathbf{U_2}} \times_3 \mathbf{U_3}||_F^2, \\ f_2(\mathcal{G}_2) &= ||\mathcal{C} - \mathcal{G}_2 \times_1 \mathbf{U_1} \times_2 \mathbf{U_2} \times_3 \widehat{\mathbf{U_3}}||_F^2. \end{aligned}$$

The augmented Lagrangian function of (53) is given by

$$\begin{aligned} L(\mathcal{G}, \mathcal{G}_1, \mathcal{G}_2, \mathcal{L}_4, \mathcal{L}_5) = \lambda_1 ||\mathcal{G}||_1 &+ \beta ||\mathcal{G} - \mathcal{G}^{pre}||_F^2 \\ &+ ||\mathcal{B} - \mathcal{G}_1 \times_1 \widehat{\mathbf{U_1}} \times_2 \widehat{\mathbf{U_2}} \times_3 \mathbf{U_3}||_F^2 \\ &+ \mu_4 ||\mathcal{G} - \mathcal{G}_1 - \mathcal{L}_4||_F^2 \\ &+ ||\mathcal{C} - \mathcal{G}_2 \times_1 \mathbf{U_1} \times_2 \mathbf{U_2} \times_3 \widehat{\mathbf{U_3}}||_F^2 \\ &+ \mu_4 ||\mathcal{G} - \mathcal{G}_2 - \mathcal{L}_5||_F^2, \end{aligned} \tag{54}$$

where μ_4 represents a penalty parameter, $\mathcal{L}_4$ and $\mathcal{L}_5$ denote the Lagrangian multipliers.

ADMM iterations of (54) are given by

$$\begin{cases} \mathcal{G}^{(t+1)} = \underset{\mathcal{G}}{\text{argmin}}\, L(\mathcal{G}, \mathcal{G}_1^{(t)}, \mathcal{G}_2^{(t)}, \mathcal{L}_4^{(t)}, \mathcal{L}_5^{(t)}), \\ \mathcal{G}_1^{(t+1)} = \underset{\mathcal{G}_1}{\text{argmin}}\, L(\mathcal{G}^{(t+1)}, \mathcal{G}_1, \mathcal{G}_2^{(t)}, \mathcal{L}_4^{(t)}, \mathcal{L}_5^{(t)}), \\ \mathcal{G}_2^{(t+1)} = \underset{\mathcal{G}_2}{\text{argmin}}\, L(\mathcal{G}^{(t+1)}, \mathcal{G}_1^{(t+1)}, \mathcal{G}_2, \mathcal{L}_4^{(t)}, \mathcal{L}_5^{(t)}), \\ \mathcal{L}_4^{(t+1)} = \underset{\mathcal{L}_4}{\text{argmin}}\, L(\mathcal{G}^{(t+1)}, \mathcal{G}_1^{(t+1)}, \mathcal{G}_2^{(t+1)}, \mathcal{L}_4, \mathcal{L}_5^{(t)}), \\ \mathcal{L}_5^{(t+1)} = \underset{\mathcal{L}_5}{\text{argmin}}\, L(\mathcal{G}^{(t+1)}, \mathcal{G}_1^{(t+1)}, \mathcal{G}_2^{(t+1)}, \mathcal{L}_4^{(t+1)}, \mathcal{L}_5). \end{cases} \tag{55}$$

Next, we present the solving process of (55).

1) The $\mathcal{G}$-subproblem: From (54), we have

$$\begin{aligned} &\underset{\mathcal{G}}{\text{argmin}}\ \lambda_1 ||\mathcal{G}||_1 + \beta ||\mathcal{G} - \mathcal{G}_{pre}||_F^2 + \\ &\mu_4 ||\mathcal{G} - \mathcal{G}_1 - \mathcal{L}_4||_F^2 + \mu_4 ||\mathcal{G} - \mathcal{G}_2 - \mathcal{L}_5||_F^2, \end{aligned} \tag{56}$$

whose solution $\mathcal{G}$ could be represented by the following column-wise vector-soft threshold function, i.e.,

$$\mathcal{G} = \text{soft}[\frac{\mu_4(\mathcal{G}_1 + \mathcal{L}_4 + \mathcal{G}_2 + \mathcal{L}_5) + \beta \mathcal{G}_{pre}}{2\mu_4 + \beta}, \frac{\lambda_1}{4\mu_4 + 2\beta}], \tag{57}$$

where the definition of soft(x, y) is the same as that in (35).

The computational complexity of updating $\mathcal{G}$ by (57) is $O(n_1 n_2 n_3)$.

2) The $\mathcal{G}_1$-subproblem: From (54), we have

$$\begin{aligned} &\underset{\mathcal{G}_1}{\text{argmin}}\ \mu_4 ||\mathcal{G}_1 - \mathcal{G} + \mathcal{L}_4||_F^2 + \\ &||\mathcal{B} - \mathcal{G}_1 \times_1 \widehat{\mathbf{U_1}} \times_2 \widehat{\mathbf{U_2}} \times_3 \mathbf{U_3}||_F^2. \end{aligned} \tag{58}$$

Problem (58) is equal to

$$\underset{\mathbf{g_1}}{\text{argmin}}\ \mu_4 ||\mathbf{g_1} - \mathbf{g} + \mathbf{l_4}||_F^2 + ||\mathbf{b} - \mathbf{M_1 g_1}||_F^2, \tag{59}$$

where the vectors $\mathbf{g_1} = \text{vec}(\mathcal{G}_1)$, $\mathbf{l_4} = \text{vec}(\mathcal{L}_4)$, $\mathbf{g} = \text{vec}(\mathcal{G})$, and $\mathbf{b} = \text{vec}(\mathcal{B})$ are generated by vectorizing the tensors $\mathcal{G}_1$, $\mathcal{L}_4$, $\mathcal{G}$ and $\mathcal{B}$, respectively, and $\mathbf{M_1} = \mathbf{U_3} \otimes \widehat{\mathbf{U_2}} \otimes \widehat{\mathbf{U_1}}$.

Problem (59) has the following closed-form solution, i.e.,

$$\mathbf{g_1} = (\mathbf{M_1}^T \mathbf{M_1} + \mu_4 \mathbf{I})^{-1} (\mathbf{M_1}^T \mathbf{b} + \mu_4 \mathbf{g} - \mu_4 \mathbf{l}_4). \tag{60}$$

Note that $\mathbf{M_1} \in \mathbb{R}^{i_1 i_2 I_3 \times n_1 n_2 n_3}$ is relatively large, therefore, it is difficult to solve (60). Fortunately, we find that

$$\begin{aligned} (\mathbf{M_1}^T \mathbf{M_1} + \mu_4 \mathbf{I})^{-1} = (\mathbf{D_3} \otimes \mathbf{D_2} \otimes \mathbf{D_1})(&\mathbf{\Sigma_3} \otimes \mathbf{\Sigma_2} \otimes \mathbf{\Sigma_1} \\ &+ \mu_4 \mathbf{I})^{-1} \times (\mathbf{D_3}^T \otimes \mathbf{D_2}^T \otimes \mathbf{D_1}^T), \end{aligned} \tag{61}$$

where $\mathbf{\Sigma_i}$ and $\mathbf{D_i}$ (i=1, 2, 3) are diagonal matrices and unitary matrices containing the eigenvalues and eigenvectors of, respectively, $\widehat{\mathbf{U_1}}^T \widehat{\mathbf{U_1}}$, $\widehat{\mathbf{U_2}}^T \widehat{\mathbf{U_2}}$, and $\mathbf{U_3}^T \mathbf{U_3}$. Therefore, $(\mathbf{\Sigma_3} \otimes \mathbf{\Sigma_2} \otimes \mathbf{\Sigma_1} + \mu_4 \mathbf{I})^{-1}$ is a diagonal matrix and could be computed easily. Besides, the term $\mathbf{M_1}^T \mathbf{b}$ in (60) can be computed by

$$\mathbf{M_1}^T \mathbf{b} = \text{vec}(\mathcal{B} \times_1 \widehat{\mathbf{U_1}}^T \times_2 \widehat{\mathbf{U_2}}^T \times_3 \mathbf{U_3}^T), \tag{62}$$

where vec$(\cdot)$ is the vectorization operation.

Therefore, we could compute (60) easily.

3) The $\mathcal{G}_2$-subproblem: From (54), we have

$$\begin{aligned}\underset{\mathcal{G}_2}{\operatorname{argmin}}\ & \mu_4 ||\mathcal{G}_2 - \mathcal{G} + \mathcal{L}_5||_F^2 + \\ & ||\mathcal{C} - \mathcal{G}_2 \times_1 \mathbf{U_1} \times_2 \mathbf{U_2} \times_3 \widehat{\mathbf{U_3}}||_F^2.\end{aligned} \tag{63}$$

Problem (63) is equal to

$$\underset{\mathbf{g_2}}{\operatorname{argmin}}\ \mu_4 ||\mathbf{g_2} - \mathbf{g} + \mathbf{l_5}||_F^2 + ||\mathbf{c} - \mathbf{M_2}\mathbf{g_2}||_F^2, \tag{64}$$

where the vectors $\mathbf{g_2} = \mathrm{vec}(\mathcal{G}_2)$, $\mathbf{l_5} = \mathrm{vec}(\mathcal{L}_5)$, $\mathbf{g} = \mathrm{vec}(\mathcal{G})$, and $\mathbf{c} = \mathrm{vec}(\mathcal{C})$ are obtained by vectorizing the tensors $\mathcal{G}_2$, $\mathcal{L}_5$, $\mathcal{G}$ and $\mathcal{C}$, respectively, and $\mathbf{M_2} = \widehat{\mathbf{U_3}} \otimes \mathbf{U_2} \otimes \mathbf{U_1}$.

Problem (64) has the closed-form solution, i.e.,

$$\mathbf{g_2} = (\mathbf{M_2}^T \mathbf{M_2} + \mu_4 \mathbf{I})^{-1} (\mathbf{M_2}^T \mathbf{c} + \mu_4 \mathbf{g} - \mu_4 \mathbf{l_5}). \tag{65}$$

Note that $\mathbf{M_2} \in \mathbb{R}^{I_1 I_2 i_3 \times n_1 n_2 n_3}$ is relatively large, therefore, it is difficult to solve (65). Fortunately, similar to (61), we have

$$\begin{aligned}(\mathbf{M_2}^T \mathbf{M_2} + \mu_4 \mathbf{I})^{-1} = & (\tilde{\mathbf{D_3}} \otimes \tilde{\mathbf{D_2}} \otimes \tilde{\mathbf{D_1}})(\tilde{\mathbf{\Sigma_3}} \otimes \tilde{\mathbf{\Sigma_2}} \otimes \tilde{\mathbf{\Sigma_1}} \\ & + \mu_4 \mathbf{I})^{-1} \times (\tilde{\mathbf{D_3}}^T \otimes \tilde{\mathbf{D_2}}^T \otimes \tilde{\mathbf{D_1}}^T),\end{aligned} \tag{66}$$

where $\tilde{\mathbf{\Sigma_i}}$ and $\tilde{\mathbf{D_i}}$ (i=1, 2, 3) are diagonal matrices and unitary matrices containing the eigenvalues and eigenvectors of, respectively, $\mathbf{U_1}^T \mathbf{U_1}$, $\mathbf{U_2}^T \mathbf{U_2}$, and $\widehat{\mathbf{U}_3}^T \widehat{\mathbf{U_3}}$. Therefore, $(\tilde{\mathbf{\Sigma_3}} \otimes \tilde{\mathbf{\Sigma_2}} \otimes \tilde{\mathbf{\Sigma_1}} + \mu_4 \mathbf{I})^{-1}$ is a diagonal matrix and could be computed easily.

4) The $\mathcal{L}_4$ and $\mathcal{L}_5$-subproblem: From (54), we update the Lagrangian multipliers $\mathcal{L}_4$ and $\mathcal{L}_5$ by

$$\begin{aligned}\mathcal{L}_4 &= \mathcal{L}_4 - (\mathcal{G} - \mathcal{G}_1), \\ \mathcal{L}_5 &= \mathcal{L}_5 - (\mathcal{G} - \mathcal{G}_2).\end{aligned} \tag{67}$$

Same with that in Li et al.'s work [12], during each iteration of ADMM, the two heaviest computation steps, shown in (60) and (65), have time complexity of $O(n_1^3 n_2^3 n_3^3)$. If we use (61) and (66) to carry out those computations and tensor i-mode products, the time complexity is reduced to $O(n_1^2 n_2 n_3 + n_1 n_2^2 n_3 + n_1 n_2 n_3^2)$.

In **Algorithm 4**, we summarize the process of solving $\mathcal{G}$-subproblem (47) by ADMM.

Algorithm 4 Solving $\mathcal{G}$-subproblem (20) via ADMM

Inputting: $\mathcal{B}$, $\mathcal{C}$, $\mathbf{U_1}$, $\widehat{\mathbf{U_1}}$, $\mathbf{U_2}$, $\widehat{\mathbf{U_2}}$, $\mathbf{U_3}$, $\widehat{\mathbf{U_3}}$, $\mathcal{G}^{pre}$, $\beta > 0$, $\mu_4 > 0$, and $\lambda_1 > 0$.
Outputting: Core tensor $\mathcal{G}$.
While not converged **do**
  Step 1 Updating $\mathcal{G}$ by (57);
  Step 2 Updating $\mathcal{G}_1$ by (60);
  Step 3 Updating $\mathcal{G}_2$ by (65);
  Step 4 Updating $\mathcal{L}_4$ and $\mathcal{L}_5$ by (67);
end while

References

[1] N. He, M. E. Paoletti, J. M. Haut, L. Fang, S. Li, A. Plaza, and J. Plaza, "Feature extraction with multiscale covariance maps for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 755–769, Feb. 2019.

[2] J. H. Yang, X. L. Zhao, T. H. Ma, Y. Chen, T. Z. Huang, and M. Ding, "Remote sensing image destriping using unidirectional high-order total variation and nonconvex low-rank regularization," *Journal of Computational and Applied Mathematics*, vol. 363, pp. 124–144, Jan. 2020.

[3] L. Loncan, L. Almeida, J. Bioucas-Dias, X. Briottet, J. Chanussot, N. Dobigeon, S. Fabre, W. Liao, G. Licciardi, M. *Simões*, J. Tourneret, M. Veganzones, G. Vivone, and Q. Wei, "Hyperspectral pansharpening: A review," *IEEE Geosci. Remote Sens.*, vol. 3, no. 3, pp. 27–46, Sep. 2015.

[4] J. H. Yang, X. L. Zhao, J. J. Mei, S. Wang, T. H. Ma, and T. Z. Huang, "Total variation and high-order total variation adaptive model for restoring blurred images with cauchy noise," *Computer and Mathematics with Application*, vol. 77, pp. 1255–1272, Mar. 2019.

[5] Z. Y. Zha, X. Yuan, B. H. Wen, J. T. Zhou, J. C. Zhang, and C. Zhu, "From rank estimation to rank approximation: Rank residual constraint for image restoration," *IEEE Trans. Image Process.*, vol. 29, pp. 3254–3269, 2020, doi:10.1109/TIP.2019.2958309.

[6] J. Huang and T. Z. Huang, "A nonstationary accelerating alternating direction method for frame-based poissonian image deblurring," *Journal of Computational and Applied Mathematics*, vol. 352, pp. 181–193, 2019.

[7] J. Ma, X. Wang, and J. Jiang, "Image superresolution via dense discriminative network," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 7, pp. 5687–5695, 2020, doi:10.1109/TIE.2019.2934071.

[8] P. Yi, Z. Wang, K. Jiang, Z. Shao, and J. Ma, "Multi-temporal ultra dense memory network for video super-resolution," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2019, doi:10.1109/TCSVT.2019.2925844.

[9] Z. Wang, M. He, Z. Ye, K. Xu, Y. Nian, and B. Huang, "Reconstruction of hyperspectral images from spectral compressed sensing based on a multitype mixing model," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 2304–2320, 2020, doi:10.1109/JSTARS.2020.2994334.

[10] X. L. Zhao, W. Wang, T. Y. Zeng, T. Z. Huang, and M. K. Ng, "Total variation structured total least squares method for image restoration," *SIAM Journal on Scientific Computing*, vol. 35, pp. 1304–1320, 2013.

[11] J. H. Yang, X. L. Zhao, T. Y. Ji, T. H. Ma, and T. Z. Huang, "Low-rank tensor train for tensor robust principal component analysis," *Applied Mathematics and Computation*, vol. 367, pp. 1–15, 2020.

[12] S. T. Li, R. W. Dian, L. Fang, and J. M. Bioucas-Dias, "Fusing hyperspectral and multispectral images via coupled sparse tensor factorization," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 4118–4130, Aug. 2018.

[13] Y. Fu, Y. Zheng, H. Huang, I. Sato, and Y. Sato, "Hyperspectral image super-resolution with a mosaic rgb image," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5539–5552, Nov. 2018.

[14] Y. Qu, H. Qi, and C. Kwan, "Unsupervised sparse dirichlet-net for hyperspectral image super-resolution," *IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2018.

[15] R. W. Dian, S. T. Li, A. Guo, and L. Fang, "Deep hyperspectral image sharpening," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5345–5355, 2018.

[16] L. J. Deng, M. Y. Feng, and X. C. Tai, "The fusion of panchromatic and multispectral remote sensing images via tensor-based sparse modeling and hyper-laplacian prior," *Information Fusion*, vol. 52, pp. 76–89, Dec. 2019.

[17] L. J. Deng, G. Vivone, W. H. Guo, M. Dalla Mura, and J. Chanussot, "A variational pansharpening approach based on reproducible kernel hilbert space and heaviside function," *IEEE Trans. Image Process*, vol. 27, pp. 4330–4344, Sep. 2018.

[18] L. J. Deng, W. H. Guo, and T. Z. Huang, "Single image super-resolution via an iterative reproducing kernel hilbert space method," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, pp. 2001–2014, Nov. 2016.

[19] J. J. Li, R. X. Cui, B. Li, R. Song, Y. S. Li, and Q. Du, "Hyperspectral image super-resolution with 1d-2d attentional convolutional neural network." *IEEE Trans. Geosci. Remote Sens.*, vol. 11, no. 23, pp. 2859–2860, Dec. 2019.

[20] S. Mei, X. Yuan, J. Ji, Y. Zhang, S. Wan, and Q. Du, "Hyperspectral image spatial super-resolution via 3d full convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 9, no. 11, 2017.

[21] J. J. Jiang, S. T. Li, H. Sun, X. M. Liu, and J. Y. Ma, "Learning spatial-spectral prior for super-resolution of hyperspectral imagery," *IEEE Transactions on Computational Imaging*, pp. 1–1, 2020, doi:10.1109/TCI.2020.2996075.

[22] H. Chen, H. Zhang, J. Du, and B. Luo, "Unified framework for the joint super-resolution and registration of multiangle multi/hyperspectral remote sensing images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 2369–2384, 2020, doi:10.1109/JSTARS.2020.2993629.

[23] M. Simões, J. Bioucas-Dias, L. Almeida, and J. Chanussot, "A convex formulation for hyperspectral image super-resolution via subspace-based regularization," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 6, pp. 3373–3388, Jun. 2015.

[24] Q. Wei, J. M. Bioucas-Dias, N. Dobigeon, and J. Tourneret, "Hyperspectral and multispectral image fusion based on a sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 7, pp. 3658–3668, Jul. 2015.

[25] C. Yi, Y. Zhao, and J. C. Chan, "Hyperspectral image super-resolution based on spatial and spectral correlation fusion," in *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 7, Jul. 2018, pp. 4165–4177.

[26] C. Lin, F. Ma, C. Chi, and C. Hsieh, "A convex optimization-based coupled nonnegative matrix factorization algorithm for hyperspectral and multispectral data fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1652–1667, Mar. 2018.

[27] R. W. Dian, S. T. Li, L. Fang, and Q. Wei, "Multispectral and hyperspectral image fusion with spatial-spectral sparse representation," *Inf. Fus.*, vol. 49, pp. 262–270, 2019.

[28] C. Lanaras, E. Baltsavias, and K. Schindler, "Hyperspectral super-resolution by coupled spectral unmixing," *in Proc. IEEE Int. Conf. Computer Vision*, pp. 3586–3594, Dec. 2015.

[29] Q. Wei, N. Dobigeon, and J. Y. Tourneret, "Fast fusion of multiband images based on solving a sylvester equation," *IEEE Trans. Image Process.*, vol. 23, pp. 1632–1636, Nov. 2016.

[30] J. Huang, T. Z. Huang, L. J. Deng, and X. L. Zhao, "Joint-sparse-blocks and low-rank representation for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, pp. 2419–2438, Apr. 2019.

[31] J. J. Wang, T. Z. Huang, J. Huang, H. X. Dou, L. J. Deng, and X. L. Zhao, "Row-sparsity spectral unmixing via total variation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pp. 1–14, Nov. 2019.

[32] X. Li, J. Huang, L. J. Deng, and T. Z. Huang, "Bilateral filter based total variation regularization for sparse hyperspectral image unmixing," *Information Sciences*, vol. 504, pp. 334–353, 2019.

[33] X. R. Feng, H. C. Li, J. Li, Q. Du, A. Plaza, and W. J. Emery, "Hyperspectral unmixing using sparsity-constrained deep nonnegative matrix factorization with total variation," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 6245–6257, 2018.

[34] R. W. Dian, L. Fang, and S. T. Li, "Hyperspectral image super-resolution via non-local sparse tensor factorization," *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 5344–5353, Jul. 2017.

[35] K. Zhang, M. Wang, S. Yang, and L. Jiao, "Spectral-graph-regularized low-rank tensor decomposition for multispectral and hyperspectral image fusion," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, 2018, doi: 10.1109/JSTARS.2017.2785411.

[36] Y. Chang, L. Yan, H. Fang, S. Zhong, and Z. Zhang, "Weighted low-rank tensor recovery for hyperspectral image restoration," *CoRR*, vol. abs/1709.00192, 2017.

[37] Y. Xu, Z. Wu, J. Chanussot, and Z. Wei, "Nonlocal patch tensor sparse representation for hyperspectral image super-resolution," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 3034–3047, Jun. 2019.

[38] R. W. Dian, , and S. T. Li, "Hyperspectral image super-resolution via subspace-based low tensor multi-rank regularization," *IEEE Trans. Image Process.*, vol. 28, pp. 5135–5146, Oct. 2019.

[39] R. W. Dian, S. T. Li, and L. Fang, "Learning a low tensor-train rank representation for hyperspectral image super-resolution," *IEEE Trans. Neur. Net. Lear.*, vol. 30, pp. 2672–2683, Sep. 2019.

[40] C. I. Kanatsoulis, X. Fu, N. D. Sidiropoulos, and W. Ma, "Hyperspectral super-resolution: A coupled tensor factorization approach," *IEEE Transactions on Signal Processing*, vol. 66, no. 24, pp. 6503–6517, Dec. 2018.

[41] H. Attouch, J. Bolte, and B. F. Svaiter, "Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward-backward splitting, and regularized gauss-seidel methods," *Math. Program.*, vol. 137, pp. 91–129, 2013.

[42] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, "Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-ojasiewicz inequality," *Math. Oper. Res.*, vol. 35, no. 2, pp. 438–457, May. 2010.

[43] F. Yasuma, T. Mitsunaga, D. Iso, and S. Nayar, "Generalized assorted pixel camera: Postcapture control of resolution, dynamic range, and spectrum," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2241–2253, Sep. 2010.

[44] A. Chakrabarti and T. Zickler, "Statistics of real-world hyperspectral images," *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 193–200, 2011.

[45] F. Dell'Acqua, P. Gamba, A. Ferrari, J. Palmason, J. Benediktsson, and K. Arnason, "Exploiting spectral and spatial information in hyperspectral urban data with high resolution," *IEEE Geosci. Remote Sens. Lett.*, vol. 1, no. 4, pp. 322–326, Oct. 2004.

[46] L. N. Smith and M. Elad, "Improving dictionary learning:multiple dictionary updates and coefficient reuse," *IEEE Signal Process.Lett.*, vol. 20, no. 1, pp. 79–82, Jan. 2013.

[47] J. M. Bioucas−Dias, "A variable splitting augmented lagrangian approach to linear spectral unmixing," *in Proc. Workshop Hyperspectral Image Signal Process., Evolution Remote Sens.*, pp. 1–4, Aug. 2009.

[48] J. Eckstein and D. Bertsekas, "On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Program.*, vol. 55, no. 3, pp. 293–318, Jun. 1992.

[49] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

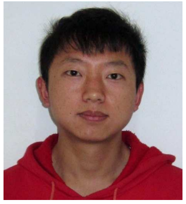
[50] G. H. Golub and C. F. V. Loan, "Matrix computations," *Baltimore, MD, USA: The Johns Hopkins Univ. Press*, 2012.

Ting Xu received the B.S. degree in mathematics and applied mathematics from the Mianyang Teachers' College, Mianyang, China, in 2018. She is pursuing the Ph. D. degree in mathematics with the School of Mathematical Sciences, University of Electronic Science and Technology of China (UESTC), Chengdu, China.

Ting-Zhu Huang received the B.S., M.S., and Ph.D. degrees in computational mathematics from the Department of Mathematics, Xi'an Jiaotong University, Xi'an, China. He is currently a Professor with the School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, China.

His current research interests include scientific computation and applications, numerical algorithms for image processing, numerical linear algebra, preconditioning technologies, and matrix analysis with applications.

Dr. Huang is an Editor of the Scientific World Journal, Advances in Numerical Analysis, the Journal of Applied Mathematics, the Journal of Pure and Applied Mathematics: Advances in Applied Mathematics, and the Journal of Electronic Science and Technology, China.



Liang-Jian Deng received the B.S. and Ph.D. degrees in applied mathematics from the School of Mathematical Sciences, University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2010 and 2016, respectively. He is currently an associate professor with the School of Mathematical Sciences, UESTC. From Sept. 2013 to Sept. 2014, he stayed at Case Western Reserve University (CWRU), Cleveland, USA, as a Joint-Training Ph.D. student. In 2017, he was working as a postdoc in HKBU. In addition, he also stayed at Isaac Newton Institute for Mathematical Sciences (Cambridge University) and HKBU for short visits. His research interest is mainly focusing on using partial differential equations (PDE), Optimization modeling and Deep learning to address several tasks in image processing and computer vision, e.g., resolution enhancement, restoration, etc.

Xi-Le Zhao received the M.S. and Ph.D. degrees in applied mathematics from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2009 and 2012, respectively. He is currently a Professor with the School of Mathematical Sciences, UESTC.

His current research interests include image processing, computer vision, and machine learning.

Jie Huang received the Ph.D. degree from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2013. Currently, she is an Associate Professor with the School of Mathematical Sciences, UESTC.

Her research interests include remote sensing and image processing.