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 presented. 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 ℓ1-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 ℓ1-norm on core tensor to characterize the sparsity, and the unidirectional TV on the corresponding coefficients 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 spatial resolution but abundant spatial resolution compared with HSIs. Therefore, fusing a LR-HSI combined with the corresponding HR-MSI to generate 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. Simões et al. in work [25] 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

Fig. 1. The fused results reconstructed by CSTF [12] and the proposed method.
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 \( x \), boldface capital letters \( X \), and calligraphic letters \( \mathcal{X} \) to represent vector, matrix and tensor, respectively. \( \mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N} \) denotes an N-mode tensor and \( x_{i_1 \ldots i_N} \) (\( 1 \leq i_n \leq I_n \)) denote its elements. \( X_{(n)} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_{n-1} \times I_{n+1} \times \ldots \times I_N} \) denotes the n-mode unfolding matrix of \( \mathcal{X} \). \( ||X||_F = \sqrt{\sum_{i_1, \ldots, i_N} |x_{i_1 \ldots i_N}|^2} \) denote the Frobenius norm, respectively.

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

\[
y_{i_1 \ldots i_{n-1} i_n + 1 \ldots i_N} = \sum_{i_n} x_{i_1 \ldots i_{n-1} i_n + 1 \ldots i_N} c_{j_n i_n},
\]

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

\[
\mathcal{X} \times_m E \times_n F = \mathcal{X} \times_n F \times_m E \ (m \neq n),
\]

especially,

\[
\mathcal{X} \times_m E \times_n F = \mathcal{X} \times_n FE \ (m = n).
\]

Given the set of matrices \( B_{nj} \in \mathbb{R}^{I_n \times I_n} (n = 1, 2, \ldots, N) \), we define the tensor \( C \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N} \) as

\[
C = \mathcal{X} \times_1 B_1 \times_2 B_2 \times_3 \ldots \times_N B_N,
\]

then, we have

\[
c = (B_N \otimes B_{N-1} \otimes \ldots \otimes B_1)x,
\]

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

B. Related Works

As mentioned in Li et al.’s work [12], the target HR-HSI \( A \in \mathbb{R}^{I_1 \times I_2 \times I_3} \) 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 \( A \in \mathbb{R}^{I_1 \times I_2 \times I_3} \) could be decomposed as

\[
A = G \times_1 U_1 \times_2 U_2 \times_3 U_3,
\]

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

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

\[
B = A \times_1 D_1 \times_2 D_2,
\]
where $D_1 \in \mathbb{R}^{I_1 \times I_1}$ and $D_2 \in \mathbb{R}^{I_2 \times I_2}$ are downsampling matrices of 1-mode and 2-mode, respectively. Meanwhile, superseding $A$ by (6), $B$ can be rewritten by

$$B = \mathcal{G} \times_1 (D_1 U_1) \times_2 (D_2 U_2) \times_3 U_3 = \mathcal{G} \times_1 \hat{U}_1 \times_2 \hat{U}_2 \times_3 U_3,$$

(8)

where $\hat{U}_1 = D_1 U_1 \in \mathbb{R}^{i_1 \times n_1}$ and $\hat{U}_2 = D_2 U_2 \in \mathbb{R}^{i_2 \times n_2}$ denote the downsampled form of $U_1$ and of $U_2$, respectively. Similarly, the HR-MSI $C \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ $(0 < i_3 < I_3)$ could be treated as a spectrally downsampled form of $A$, i.e.,

$$C = A \times_3 D_3,$$

(9)

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

$$C = \mathcal{G} \times_1 U_1 \times_2 U_2 \times_3 (D_3 U_3) = \mathcal{G} \times_1 U_1 \times_2 U_2 \times_3 \hat{U}_3,$$

(10)

where $\hat{U}_3 = D_3 U_3 \in \mathbb{R}^{i_3 \times I_3}$ is the downsampled form of $U_3$. Therefore, the estimation of $A$ is equivalent to the estimation of $U_1$, $U_2$, $U_3$, and $\mathcal{G}$, as visualized in Fig. 2.

### C. Motivations

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

$$\arg\min_{U_1, U_2, U_3, \mathcal{G}} ||B - \mathcal{G} \times_1 \hat{U}_1 \times_2 \hat{U}_2 \times_3 U_3||_F^2 + ||C - \mathcal{G} \times_1 U_1 \times_2 U_2 \times_3 \hat{U}_3||_F^2.$$  

(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 $A$ 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 $\ell_1$-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.,

$$\arg\min_{U_1, U_2, U_3, \mathcal{G}} ||B - \mathcal{G} \times_1 \hat{U}_1 \times_2 \hat{U}_2 \times_3 U_3||_F^2 + ||C - \mathcal{G} \times_1 U_1 \times_2 U_2 \times_3 \hat{U}_3||_F^2$$

$$+ \lambda_1||\mathcal{G}||_1 + \lambda_2||D_y U_1||_1 + \lambda_3||D_y U_2||_1 + \lambda_4||D_y U_3||_1,$$

(12)

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

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

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 $\mathcal{G}$, $U_1$, $U_2$, and $U_3$. 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,

$$\begin{align*}
U_1 &= \arg\min_{\hat{U}_1} f(U_1, U_2, U_3, \mathcal{G}) + \beta||U_1 - U_1^{pre}||_F^2, \\
U_2 &= \arg\min_{\hat{U}_2} f(U_1, U_2, U_3, \mathcal{G}) + \beta||U_2 - U_2^{pre}||_F^2, \\
U_3 &= \arg\min_{\hat{U}_3} f(U_1, U_2, U_3, \mathcal{G}) + \beta||U_3 - U_3^{pre}||_F^2, \\
\mathcal{G} &= \arg\min_{G} f(U_1, U_2, U_3, \mathcal{G}) + \beta||\mathcal{G} - G^{pre}||_F^2, \\
\end{align*}$$

(13)

where the function $f(U_1, U_2, U_3, \mathcal{G})$ is the implicit definition of (12), $f^{pre}$ and $\beta$ 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 $U_1$

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

$$\arg\min_{U_1} ||B - \mathcal{G} \times_1 \hat{U}_1 \times_2 \hat{U}_2 \times_3 U_3||_F^2 + ||C - \mathcal{G} \times_1 U_1 \times_2 U_2 \times_3 \hat{U}_3||_F^2$$

$$+ \lambda_2||D_y U_1||_1 + \beta||U_1 - U_1^{pre}||_F^2,$$

(14)

where $U_1^{pre}$ is the last estimated dictionary of 1-mode and $D_y \in \mathbb{R}^{(I_1 - 1) \times I_1}$ denotes the difference matrix along the vertical direction of $U_1$. 
By 1-mode matrix unfolding, problem (14) can be represented by
\[
\begin{aligned}
\argmin_{U_1} & \quad ||B(1) - D_1 U_1 X_1||_F^2 + ||C(1) - U_1 Y_1||_F^2 \\
& + \lambda_2 ||D_y U_1||_1 + \beta ||U_1 - U_1^{pre}||_F^2,
\end{aligned}
\tag{15}
\]
where \(B(1)\) and \(C(1)\) are 1-mode unfolding matrices of \(B\) and \(C\), respectively; \(X_1\) and \(Y_1\) are denoted by \(X_1 = (G \times_2 \hat{U}_2 \times_3 U_3)_{(1)}\) and \(Y_1 = (\hat{G} \times_2 U_2 \times_3 \hat{U}_3)_{(1)}\), respectively.

### B. The optimization problem of \(U_2\)

With fixing \(U_1\), \(U_3\), and \(G\), the optimization problem of \(U_2\) in (13) is given by
\[
\begin{aligned}
\argmin_{U_2} & \quad ||B - G \times_1 \hat{U}_2 \times_2 \hat{U}_3 \times_3 U_3||_F^2 \\
& + ||C - G \times_1 U_1 \times_2 U_2 \times_3 \hat{U}_3||_F^2 \\
& + \lambda_3 ||D_y U_2||_1 + \beta ||U_2 - U_2^{pre}||_F^2,
\end{aligned}
\tag{16}
\]
where \(U_2^{pre}\) is the last estimated dictionary of 2-mode and \(D_y \in \mathbb{R}^{(l_z-1)x\ell_z}\) denotes the difference matrix along the vertical direction of \(U_2\).

By 2-mode matrix unfolding, problem (16) can be represented by
\[
\begin{aligned}
\argmin_{U_2} & \quad ||B(2) - D_2 U_2 X_2||_F^2 + ||C(2) - U_2 Y_2||_F^2 \\
& + \lambda_3 ||D_y U_2||_1 + \beta ||U_2 - U_2^{pre}||_F^2,
\end{aligned}
\tag{17}
\]
where \(B(2)\) and \(C(2)\) are 2-mode unfolding matrices of \(B\) and \(C\), respectively; \(X_2\) and \(Y_2\) are denoted by \(X_2 = (G \times_1 \hat{U}_1 \times_3 U_3)_{(2)}\) and \(Y_2 = (\hat{G} \times_1 U_1 \times_3 \hat{U}_3)_{(2)}\), respectively.

### C. The optimization problem of \(U_3\)

With fixing \(U_1\), \(U_2\), and \(G\), the optimization problem of \(U_3\) in (13) can be represented as follows:
\[
\begin{aligned}
\argmin_{U_3} & \quad ||B - G \times_1 \hat{U}_1 \times_2 \hat{U}_2 \times_3 U_3||_F^2 \\
& + ||C - G \times_1 U_1 \times_2 U_2 \times_3 \hat{U}_3||_F^2 \\
& + \lambda_4 ||D_y U_3||_1 + \beta ||U_3 - U_3^{pre}||_F^2,
\end{aligned}
\tag{18}
\]
where \(U_3^{pre}\) is the last estimated dictionary of 3-mode and \(D_y \in \mathbb{R}^{(l_z-1)x\ell_z}\) denotes the difference matrix along the vertical direction of \(U_3\).

By using 3-mode matrix unfolding, (18) can be represented by
\[
\begin{aligned}
\argmin_{\hat{G}} & \quad ||B(3) - U_3 X_3||_F^2 + ||C(3) - D_3 U_3 Y_3||_F^2 \\
& + \lambda_4 ||D_y U_3||_1 + \beta ||U_3 - U_3^{pre}||_F^2,
\end{aligned}
\tag{19}
\]
where \(B(3)\) and \(C(3)\) are 3-mode unfolding matrices of \(B\) and \(C\), respectively; \(X_3\) and \(Y_3\) are denoted by \(X_3 = (G \times_1 \hat{U}_1 \times_2 U_2)_{(3)}\) and \(Y_3 = (\hat{G} \times_1 U_1 \times_2 U_2)_{(3)}\), respectively.

### D. The Optimization problem of \(\hat{G}\)

With fixing dictionary matrices \(U_1\), \(U_2\), and \(U_3\), the optimization problem of the core tensor \(\hat{G}\) in (13) is given by
\[
\begin{aligned}
\argmin_{\hat{G}} & \quad ||B - G \times_1 \hat{U}_1 \times_2 \hat{U}_2 \times_3 U_3||_F^2 \\
& + ||C - G \times_1 U_1 \times_2 U_2 \times_3 \hat{U}_3||_F^2 \\
& + \lambda_1 ||\hat{G}||_1 + \beta ||\hat{G} - \hat{G}^{pre}||_F^2,
\end{aligned}
\tag{20}
\]
where \(\hat{G}^{pre}\) 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
\[
\text{RelCha} = \frac{||X^{(t+1)} - X^{(t)}||_F}{||X^{(t)}||_F}.
\tag{21}
\]

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.
F. Convergence of the proposed algorithm

Same with Li et al.’s work [12], we initialize $U_1$ and $U_2$ from the HR-MSI by dictionary-updates-cycles KSVD (DUC-KSVD) algorithm [46], and $U_3$ 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.

Initialize $U_1$, $U_2$ via DUC-KSVD algorithm [46];
Initialize $U_3$ via SISAL algorithm [47];
Initialize $G$ with Alg. 4;
While not converged do
   Step 1 Updating the dictionary matrix $U_1$ by Alg. 1,
   $\hat{U}_1 = D_1 U_1$, $U_1^{pre} = U_1$;
   Step 2 Updating the dictionary matrix $U_2$ by Alg. 2,
   $\hat{U}_2 = D_2 U_2$, $U_2^{pre} = U_2$;
   Step 3 Updating the dictionary matrix $U_3$ by Alg. 3,
   $\hat{U}_3 = D_3 U_3$, $U_3^{pre} = U_3$;
   Step 4 Updating the core tensor $G$ by Alg. 4,
   $G^{pre} = G$;
end while

Estimate the target HR-HSI $A$ by (6).

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

$$f(U_1, U_2, U_3, G) = Q(U_1, U_2, U_3, G) + f_1(G) + f_2(U_1) + f_3(U_2) + f_4(U_3),$$

where

$$Q(U_1, U_2, U_3, G) = \|B - G \times_1 \hat{U}_1 \times_2 \hat{U}_2 \times_3 U_3\|_F^2 + \|C - G \times_1 U_1 \times_2 U_2 \times_3 \hat{U}_3\|_F^2,$$

and

$$f_1(G) = \lambda_1 \|G\|_1,$$

$$f_2(U_1) = \lambda_2 \|D_2 U_1\|_1,$$

$$f_3(U_2) = \lambda_3 \|D_2 U_2\|_1,$$

$$f_4(U_3) = \lambda_4 \|D_2 U_3\|_1.$$  

Proposition 1: Assume that problem (14) ($U_1$-subproblem), problem (16) ($U_2$-subproblem), problem (18) ($U_3$-subproblem), and problem (20) ($G$-subproblem) have exact solutions \(^1\).

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

Proof: First, function $Q(U_1, U_2, U_3, G)$ is $C^1$ with Lipschitz continuous gradient (the result of the boundness of $(U_1^{(t)}, U_2^{(t)}, U_3^{(t)}, G^{(t)})_{t \in \mathbb{N}}$; $f_1(G)$, $f_2(U_1)$, $f_3(U_2)$, and $f_4(U_3)$ are lower semicontinuous and proper. Second, $f(U_1, U_2, U_3, G) : \mathbb{R}^{I_3 n_3} \times \mathbb{R}^{I_2 n_2} \times \mathbb{R}^{I_1 n_1} \times \mathbb{R}^{n_1 n_2 n_3} \to \mathbb{R}$ 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 $B_i = \beta I$, $\beta > 0$ (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 $U_1$-subproblem, $U_2$-subproblem, $U_3$-subproblem, and $G$-subproblem in (13) by ADMM. For the $U_1$-subproblem, during each iteration of ADMM, the most time-consuming step is updating $U_1$ (i.e., (33)) by CG. During each iteration of CG, the most time-consuming is that multiplying the system matrix by a vector, its time complexity is $O(n_1^2 I_1^2)$, which could be reduced to $O(n_1^2 I_1^2 + n_2^2 I_2)$ due to the matrix representation. Similar to the update of $U_1$, in each iteration of CG algorithm, the complexity of $U_2$ and $U_3$ are $O(n_2^2 I_2^2 + n_3^2 I_3^2)$ and $O(n_2^2 I_2^2 + n_3^2 I_3^2)$, respectively. For the $G$-subproblem, in each iteration of ADMM, the most time-consuming are (60) and (65), whose complexity is $O(n_1^2 n_2^2 n_3^2)$. 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{align*}
&O(N_{ADMM}(N_{CG}(n_1^2 I_1^2 + n_2^2 I_2)) + O(N_{ADMM}(N_{CG}(n_2^2 I_2^2 + n_3^2 I_3))) + O(N_{ADMM}(n_2^2 I_2^2 + n_3^2 I_3)),
\end{align*}

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 × 340 × 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 × 256 × 93). In order to obtain the simulated LR-HSI with the size of 16 × 16 × 93, we downsample the GT by averaging the 16 × 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) [45]. It has 32 indoor HSIs and each of them is with the size of 512 × 512 × 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 × 32 × 31, we downsample each of them by averaging the 16 × 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 × 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 × 512 × 31 block of each selected HSI as the GT. To obtain the simulated LR-HSI with the size of 32 × 32 × 31, we downsample each of them by averaging the 16 × 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 λ1, smoothness regularization parameters λ2, λ3, λ4, the weight of proximal term β, and the number of atoms of dictionaries n1, n2, n3. 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 10, 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 β (log is base 10). In this work, we set the range of log β to [-6, 0]. As we saw in Fig. 5, the PSNR increases in waves when log β varies from -6 to -2, remains stable when log β belongs to [-2, -1], and has a sharp drop when log β is greater than -1. Therefore, we set log β 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 λ1 in (12) controls the sparsity of the core tensor and, therefore, affects the estimation of HR-HSI. Higher values of λ1 yield sparser core tensor. Fig. 6 shows the PSNR values of the reconstructed HSI of the Pavia University dataset to different log λ1. In this work, we set the range of log λ1 to [-9, -1]. As we saw in Fig. 6, the PSNR remains stable when log λ1 belongs to [-9, -5], and decreases sharply for log λ1 >-5. Therefore, we set log λ1 as -6, that is, λ1 = 10^{-6} for the Pavia University dataset. Similarly, the value of λ1 for the CAVE and Harvard dataset could be determined in the same way.

The regularization parameter λ2, λ3, and λ4 control the piecewise smoothness of the dictionary of width mode, height mode, and spectral mode, respectively. Higher values of λ2, λ3, and λ4 yield smoother dictionaries. Fig. 7 shows the PSNR values of the reconstructed HSI of the Pavia University dataset under different log λ2, log λ3, and log λ4. In this work, we set the range of log λ2, log λ3, log λ4 all to [-9, 3]. As Fig. 7, the PSNR reaches the maximum value when log λ2 = -8, log λ3 = -7, log λ4 = 3. Therefore, we set log λ2 as -8, log λ3 as -7, log λ4 as 3, that is, λ2 = 10^-8, λ3 = 10^{-7}, and λ4 = 10^3 for the Pavia University dataset. Note that, compared with λ2 and λ3, the best value of λ4 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 λ2, λ3, and λ4 for the CAVE and Harvard dataset could be determined in the same way.

n1, n2, and n3 are the number of dictionary atoms. Fig. 8 shows the PSNR values of the fused HSI of the Pavia University dataset with different n1 and n2 and Fig. 9 shows
TABLE II
THE DISCUSSION OF THE MAIN PARAMETERS

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$K$</td>
<td>[1, 60]</td>
<td>$10^{-2}$</td>
<td>$10^{-4}$</td>
<td>20</td>
<td>[12, 60]</td>
</tr>
<tr>
<td>$\beta$</td>
<td>$10^{-6}, 10^{0}$</td>
<td>$10^{-2}$</td>
<td>$10^{-4}$</td>
<td>$10^{-1}, 10^{-4}$</td>
<td></td>
</tr>
<tr>
<td>$\lambda_1$</td>
<td>$10^{-6}, 10^{-1}$</td>
<td>$10^{-3}$</td>
<td>$10^{-4}$</td>
<td>$10^{-1}, 10^{-4}$</td>
<td></td>
</tr>
<tr>
<td>$\lambda_2$</td>
<td>$10^{-7}, 10^{1}$</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>$10^{-3}, 10^{-4}$</td>
<td></td>
</tr>
<tr>
<td>$\lambda_3$</td>
<td>$10^{-5}, 10^{1}$</td>
<td>$10^{-5}$</td>
<td>$10^{-5}$</td>
<td>$10^{-3}, 10^{-4}$</td>
<td></td>
</tr>
<tr>
<td>$n_1$</td>
<td>[180, 320]</td>
<td>280</td>
<td>300</td>
<td>320</td>
<td>[280, 320]</td>
</tr>
<tr>
<td>$n_2$</td>
<td>[180, 320]</td>
<td>300</td>
<td>300</td>
<td>280</td>
<td>[280, 320]</td>
</tr>
<tr>
<td>$n_3$</td>
<td>[3, 21]</td>
<td>12</td>
<td>12</td>
<td>9</td>
<td>[9, 12]</td>
</tr>
</tbody>
</table>

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 $\lambda_p = 5 \times 10^{-5}$. 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.

D. Quantitative Metrics

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

Fig. 4. The PSNR values concerning different $K$.

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 $\log \lambda_4$. 
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,

\[
\begin{align*}
\text{RMSE}(A, \hat{A}) &= \sqrt{\frac{||A - \hat{A}||_F^2}{I_1I_2I_3}}, \\
\text{DD}(A, \hat{A}) &= \frac{1}{I_1I_2I_3}||\text{vec}(A) - \text{vec}(\hat{A})||_1, \\
\text{SAM}(A, \hat{A}) &= \frac{1}{I_1I_2I_3} \sum_{i=1}^{I_3} \arccos \frac{\hat{a}_i^T a_i}{||\hat{a}_i||_2||a_i||_2}, \\
\text{PSNR}(A, \hat{A}) &= \frac{1}{I_3} \sum_{i=1}^{I_3} \text{PSNR}(A_i, \hat{A}_i), \\
\text{ERGAS}(A, \hat{A}) &= \frac{100}{d} \sqrt{\frac{1}{I_3} \sum_{i=1}^{I_3} \frac{\text{MSE}(A_i, \hat{A}_i)}{\mu_i^2}},
\end{align*}
\]

where \(A_i\) and \(\hat{A}_i\) denote the \(i\)-th band of \(A \in \mathbb{R}^{I_1 \times I_2 \times I_3}\) and reconstructed result \(\hat{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}\), respectively; \(I_j\), \((j=1, 2, 3)\) represent the dimension of the \(j\)-th mode of \(A\); \(d\) denotes the spatial downsampling factor; \(\text{MSE}(\cdot)\) and \(\mu_i(\cdot)\) 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.

**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 6\(^{th}\) 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), (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
Fig. 10. The 1st row presents the reconstructed results of the Pavia University for the 5th band. The 2nd row presents the error results at the 5th band. The 3rd row presents the reconstructed 60th band. The 4th row presents the error results at the 60th 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.

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

<table>
<thead>
<tr>
<th>Method</th>
<th>CAVE dataset [43]</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE($\pm$std)</td>
<td>ERGAS($\pm$std)</td>
<td>DD($\pm$std)</td>
<td>SAM($\pm$std)</td>
<td>PSNR($\pm$std)</td>
</tr>
<tr>
<td>Best Values</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$+\infty$</td>
</tr>
<tr>
<td>HySure [23]</td>
<td>3.322($\pm$1.238)</td>
<td>0.886($\pm$0.281)</td>
<td>1.976($\pm$1.011)</td>
<td>14.964($\pm$4.845)</td>
<td>39.512($\pm$3.085)</td>
</tr>
<tr>
<td>CSU [28]</td>
<td>3.306($\pm$1.378)</td>
<td>0.870($\pm$0.261)</td>
<td>1.624($\pm$0.943)</td>
<td>8.391($\pm$2.490)</td>
<td>39.001($\pm$3.656)</td>
</tr>
<tr>
<td>FUSE [29]</td>
<td>2.436($\pm$1.068)</td>
<td>0.625($\pm$0.141)</td>
<td>1.270($\pm$0.747)</td>
<td>8.122($\pm$1.476)</td>
<td>42.188($\pm$3.410)</td>
</tr>
<tr>
<td>CSTF [12]</td>
<td>2.722($\pm$1.487)</td>
<td>0.671($\pm$0.195)</td>
<td>1.551($\pm$0.790)</td>
<td>8.632($\pm$1.459)</td>
<td>42.240($\pm$4.027)</td>
</tr>
<tr>
<td>Proposed</td>
<td>2.306($\pm$1.031)</td>
<td>0.590($\pm$0.136)</td>
<td>1.167($\pm$0.581)</td>
<td>7.934($\pm$1.388)</td>
<td>43.126($\pm$3.301)</td>
</tr>
</tbody>
</table>
Fig. 12. The 1st row presents the results generated by the proposed and the other four compared methods for \textit{paints} at the 10\textsuperscript{th} band. The 2nd row presents the error results at the 10\textsuperscript{th} band. The 3rd row presents the reconstructed 29\textsuperscript{th} band. The 4th row presents the error results at the 29\textsuperscript{th} band. (a) LR-HSI, (b) HySure [23], (c) CSU [28], (d) FUSE [29], (e) CSTF [12], (f) Proposed, (g) Ground Truth.

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

**TABLE V**

<table>
<thead>
<tr>
<th>Methods</th>
<th>Harvard dataset [44]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE(±std)</td>
</tr>
<tr>
<td><strong>Best Values</strong></td>
<td>0</td>
</tr>
<tr>
<td>HySure [23]</td>
<td>1.445(±0.634)</td>
</tr>
<tr>
<td>CSU [28]</td>
<td>1.639(±0.553)</td>
</tr>
<tr>
<td>FUSE [29]</td>
<td>1.179(±0.385)</td>
</tr>
<tr>
<td>CSTF [12]</td>
<td>1.184(±0.532)</td>
</tr>
<tr>
<td><strong>Proposed</strong></td>
<td><strong>1.142(±0.479)</strong></td>
</tr>
</tbody>
</table>
25th band of a HSI in the Harvard dataset, the error images at the 25th band, the reconstructed 31th band, and the error images at the 31th 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., $\lambda_2 ||D_y U_1||_1$, $\lambda_3 ||D_y U_2||_1$ and $\lambda_4 ||D_y U_3||_1$). In the proposed model (12), we use the three regularization terms to characterize 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: $\lambda_2 ||D_y U_1||_1$, $\lambda_3 ||D_y U_2||_1$ and $\lambda_4 ||D_y U_3||_1$. We set $\lambda_2$ equal to zero to demonstrate the effectiveness of $||D_y U_1||_1$, set $\lambda_3$ equal to zero to demonstrate the effectiveness of $||D_y U_2||_1$, and $\lambda_4$ equal to zero to demonstrate the effectiveness of $||D_y U_3||_1$. It is worth note that the proposed model degenerates to Dian et al. [12] when we set $\lambda_2$, $\lambda_3$, and $\lambda_4$ 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 proposed method 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](image)

### Table VII
**Average Filter with Different Downsampling Factor (DF).**

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR</th>
<th>SAM</th>
<th>DD</th>
<th>ERGAS</th>
<th>DF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best Values</td>
<td>+∞</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LTMR [38]</td>
<td>46.089</td>
<td>1.426</td>
<td>0.816</td>
<td>0.781</td>
<td>4</td>
</tr>
<tr>
<td>43.585</td>
<td>2.084</td>
<td>1.517</td>
<td>0.735</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>35.167</td>
<td>5.450</td>
<td>3.595</td>
<td>0.816</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>31.908</td>
<td>8.197</td>
<td>5.544</td>
<td>0.611</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>44.866</td>
<td>1.686</td>
<td>0.996</td>
<td>0.902</td>
<td>4</td>
</tr>
<tr>
<td>43.322</td>
<td>1.797</td>
<td>1.075</td>
<td>0.484</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>43.655</td>
<td>1.972</td>
<td>1.195</td>
<td>0.263</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>43.284</td>
<td>2.083</td>
<td>1.276</td>
<td>0.137</td>
<td>32</td>
<td></td>
</tr>
</tbody>
</table>

![Fig. 15](image)

**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.

![Table VIII](image)

### Table VIII
**Gaussian Filter with Different Downsampling Factor (DF).**

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR</th>
<th>SAM</th>
<th>DD</th>
<th>ERGAS</th>
<th>DF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best Values</td>
<td>+∞</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LTMR [38]</td>
<td>46.089</td>
<td>1.426</td>
<td>0.816</td>
<td>0.781</td>
<td>4</td>
</tr>
<tr>
<td>41.274</td>
<td>2.594</td>
<td>1.517</td>
<td>0.735</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>35.167</td>
<td>5.450</td>
<td>3.595</td>
<td>0.816</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>31.908</td>
<td>8.197</td>
<td>5.544</td>
<td>0.611</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>44.866</td>
<td>1.686</td>
<td>0.996</td>
<td>0.902</td>
<td>4</td>
</tr>
<tr>
<td>43.322</td>
<td>1.797</td>
<td>1.075</td>
<td>0.484</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>43.655</td>
<td>1.972</td>
<td>1.195</td>
<td>0.263</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>43.284</td>
<td>2.083</td>
<td>1.276</td>
<td>0.137</td>
<td>32</td>
<td></td>
</tr>
</tbody>
</table>

### 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 ℓ1-norm on the 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.

### VII. APPENDIX

#### A. The optimization problem of $U_1$

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

$$
\arg\min_{U_1} \|B_1 - D_1 U_1 X_1\|_F^2 + \|C_1 - U_1 Y_1\|_F^2 + \lambda_1 \|V_1\|_1 + \beta \|U_1 - U_1^{pre}\|_F^2,
$$

subject to $V_1 = D_y U_1$.

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

$$
L(U_1, V_1, L_1) = \|B_1 - D_1 U_1 X_1\|_F^2 + \|C_1 - U_1 Y_1\|_F^2 + \lambda_1 \|V_1\|_1 + \beta \|U_1 - U_1^{pre}\|_F^2 + \mu_1 \|D_y U_1 - V_1 - L_1\|_F^2,
$$

\[ (30) \]
where $L_1$ denotes the Lagrangian multiplier and $\mu_1$ represents a positive penalty parameter.

ADMM iterations of problem (30) are given by

\[
\begin{align*}
U_1^{(t+1)} &= \arg\min_{U_1} L(U_1, V_1^{(t)}, L_1^{(t)}), \\
V_1^{(t+1)} &= \arg\min_{V_1} L(U_1^{(t+1)}, V_1, L_1^{(t)}), \\
L_1^{(t+1)} &= \arg\min_{L_1} L(U_1^{(t+1)}, V_1^{(t+1)}, L_1).
\end{align*}
\]

(31)

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

1) The $U_1$-subproblem: From (30), we have

\[
\arg\min_{U_1} \|B_1 - D_1 U_1 X_1\|_F^2 + \|C_1 - U_1 Y_1\|_F^2 + \beta\|U_1 - U_1^{pre}\|_F^2 + \mu_1\|D_y U_1 - V_1 - L_1\|_F^2.
\]

(32)

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

\[
D_1^T D_1 U_1 X_1 X_1^T + U_1 (Y_1 Y_1^T + \beta I) + \mu_1 D_y^T D_y U_1 = D_1^T B_1 X_1^T + C_1 Y_1^T + \beta U_1^{pre} + \mu_1 D_y^T (V_1 + L_1).
\]

(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 $V_1$-subproblem: From (30), we have

\[
\arg\min_{V_1} \lambda_2 \|V_1\|_1 + \mu_1\|D_y U_1 - V_1 - L_1\|_F^2,
\]

(34)

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

\[
V_1 = \text{soft}(D_y U_1 - L_1, \frac{\lambda_2}{2\mu_1}),
\]

(35)

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

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

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

\[
L_1 = L_1 - (D_y U_1^* - V_1).
\]

(36)

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

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 $U_1$-subproblem (14) by ADMM.

**Algorithm 1** Solving $U_1$-subproblem (14) via ADMM

**Inputting:** $B$, $C$, $U_2$, $U_3$, $\beta$, $\mu_1 > 0$, $\lambda_2 > 0$.

**Outputting:** Dictionary matrix $U_1$.

**While** not converged **do**

**Step 1** Updating the dictionary matrix $U_1$ by (33);

**Step 2** Updating the variable $V_1$ by (35);

**Step 3** Updating the Lagrangian multiplier $L_1$ by (36);

**end while**

**B. The optimization problem of $U_2$**

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

\[
\begin{align*}
\arg\min_{U_2} & \|B_2 - D_2 U_2 X_2\|_F^2 + \|C_2 - U_2 Y_2\|_F^2 + \lambda_3 \|V_2\|_1 + \beta \|U_2 - U_2^{pre}\|_F^2, \\
\text{s.t.} & \quad V_2 = D_y U_2.
\end{align*}
\]

(37)

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

\[
L(U_2, V_2, L_2) = \|B_2 - D_2 U_2 X_2\|_F^2 + \|C_2 - U_2 Y_2\|_F^2 + \lambda_3 \|V_2\|_1 + \beta \|U_2 - U_2^{pre}\|_F^2 + \mu_2 \|D_y U_2 - V_2 - L_2\|_F^2,
\]

(38)

where $L_2$ denotes the Lagrangian multiplier and $\mu_2$ represents a positive penalty parameter.

ADMM iterations of problem (30) are given by

\[
\begin{align*}
U_2^{(t+1)} &= \arg\min_{U_2} L(U_2, V_2^{(t)}, L_2^{(t)}), \\
V_2^{(t+1)} &= \arg\min_{V_2} L(U_2^{(t+1)}, V_2, L_2^{(t)}), \\
L_2^{(t+1)} &= \arg\min_{L_2} L(U_2^{(t+1)}, V_2^{(t+1)}, L_2).
\end{align*}
\]

(39)

ADMM iterations of problem (30) are given by

\[
\begin{align*}
U_2^{(t+1)} &= \arg\min_{U_2} L(U_2, V_2^{(t)}, L_2^{(t)}), \\
V_2^{(t+1)} &= \arg\min_{V_2} L(U_2^{(t+1)}, V_2, L_2^{(t)}), \\
L_2^{(t+1)} &= \arg\min_{L_2} L(U_2^{(t+1)}, V_2^{(t+1)}, L_2).
\end{align*}
\]

(40)

ADMM iterations of problem (30) are given by

\[
\begin{align*}
U_2^{(t+1)} &= \arg\min_{U_2} L(U_2, V_2^{(t)}, L_2^{(t)}), \\
V_2^{(t+1)} &= \arg\min_{V_2} L(U_2^{(t+1)}, V_2, L_2^{(t)}), \\
L_2^{(t+1)} &= \arg\min_{L_2} L(U_2^{(t+1)}, V_2^{(t+1)}, L_2).
\end{align*}
\]

(41)
where $L_3$ denotes the Lagrangian multiplier and $\mu_2 > 0$ represents a penalty parameter.

ADMM iterations of problem (38) are given by

\[
\begin{aligned}
U_2^{(t+1)} &= \arg\min_{U_2} L(U_2, V_2^{(t)}, L_2^{(t)}), \\
V_2^{(t+1)} &= \arg\min_{V_2} L(U_2^{(t+1)}, V_2, L_2^{(t)}), \\
L_2^{(t+1)} &= \arg\min_{L_2} L(U_2^{(t+1)}, V_2^{(t+1)}, L_2).
\end{aligned}
\]

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

1) The $U_2$-subproblem: From (38), we have

\[
\begin{aligned}
&\text{argmin}_{U_2} ||B(2) - D_2 U_2 X_2 ||_F^2 + ||C(2) - U_2 Y_2 ||_F^2 + \\
&\quad \beta ||U_2 - U_2^{pre}||_F^2 + \mu_2 ||D_y U_2 - V_2 - L_2||_F^2.
\end{aligned}
\]

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

\[
D_2^T D_2 U_2 X_2 X_2^T + U_2 (Y_2 Y_2^T + \beta I) + \mu_2 D_y^T D_y U_2 =
D_2^T B(2) X_2^T + C(2) Y_2^T + \beta U_2^{pre}
+ \mu_2 D_y^T (V_2 + L_2).
\]

Similarly, we utilize CG to solve (41). During each iteration of CG, similar to (33), the heaviest computational complexity is $O(n_3^2 I_2^2)$, which could be reduced to $O(n_3^2 I_2^2 + n_2 I_2^2)$ due to the matrix representation.

2) The $V_2$-subproblem: From (38), we have

\[
\begin{aligned}
&\text{argmin}_{V_2} \lambda_3 ||V_2||_1 + \mu_2 ||D_y U_2 - V_2 - L_2||_F^2,
\end{aligned}
\]

whose solution $V_2$ could be given by the following columnwise vector-soft threshold function, i.e.,

\[
V_2 = \text{soft}(D_y U_2 - L_2, \frac{\lambda_3}{2\mu_2}),
\]

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

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

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

\[
L_2 = L_2 - (D_y U_2 - V_2).
\]

The computational complexity of updating $L_2$ 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 $O(N_{CG}(n_3^2 I_2^2 + n_2 I_2^2))$, where $N_{CG}$ is the iteration number of CG.

In Algorithm 2, we summarize the process of solving $U_2$-subproblem (16) via ADMM.

**Algorithm 2** Solving $U_2$-subproblem (16) via ADMM

**Inputting:** $B, C, U_1, \widehat{U}_1, U_3, \widehat{U}_3, \mathcal{G}, U_2^{pre}, \beta > 0, \mu_2 > 0$, and $\lambda_3 > 0$.

**Outputting:** Dictionary matrix $U_2$.

**While** not converged do

1) Updating the dictionary matrix $U_2$ by (41);
2) Updating the variable $V_2$ by (43);
3) Updating the Lagrangian multiplier $L_2$ by (44);

**end while**

C. The optimization problem of $U_3$.

Similar to problem (15), problem (19) is convex, therefore, we use ADMM to solve it efficiently. We first introduce the splitting variable $V_3 = D_y U_3$ and then rewrite the problem (19) as follows,

\[
\begin{aligned}
&\text{argmin}_{U_3} ||B(3) - U_3 X_3||_F^2 + ||C(3) - D_3 U_3 Y_3||_F^2 \\
&+ \lambda_4 ||V_3||_1 + \beta ||U_3 - U_3^{pre}||_F^2,
\end{aligned}
\]

s.t. $V_3 = D_y U_3$.

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

\[
L(U_3, V_3, L_3) = ||B(3) - U_3 X_3||_F^2 + ||C(3) - D_3 U_3 Y_3||_F^2 \\
+ \lambda_4 ||V_3||_1 + \beta ||U_3 - U_3^{pre}||_F^2 \\
+ \mu_3 ||D_y U_3 - V_3 - L_3||_F^2,
\]

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

ADMM iterations of problem (46) are given by

\[
\begin{aligned}
U_3^{(t+1)} &= \arg\min_{U_3} L(U_3, V_3^{(t)}, L_3^{(t)}), \\
V_3^{(t+1)} &= \arg\min_{V_3} L(U_3^{(t+1)}, V_3, L_3^{(t)}), \\
L_3^{(t+1)} &= \arg\min_{L_3} L(U_3^{(t+1)}, V_3^{(t+1)}, L_3).
\end{aligned}
\]

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

1) The $U_3$-subproblem: From (46), we have

\[
\begin{aligned}
&\text{argmin}_{U_3} ||B(3) - U_3 X_3||_F^2 + ||C(3) - D_3 U_3 Y_3||_F^2 \\
&+ \beta ||U_3 - U_3^{pre}||_F^2 + \mu_3 ||D_y U_3 - V_3 - L_3||_F^2.
\end{aligned}
\]

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

\[
D_3^T D_3 U_3 Y_3 Y_3^T + U_3 (X_3 X_3^T + \beta I) + \mu_3 D_y^T D_y U_3 =
D_3^T C(3) Y_3^T + B(3) X_3^T + \mu_3 U_3^{pre}
+ \mu_3 D_y^T (V_3 + L_3).
\]

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 $O(n_3^2 I_3^2 + n_3 I_3^2)$ due to the matrix representation.

2) The $V_3$-subproblem: From (46), we have

\[
\begin{aligned}
&\text{argmin}_{V_3} \lambda_4 ||V_3||_1 + \mu_3 ||D_y U_3 - V_3 - L_3||_F^2,
\end{aligned}
\]

whose solution $V_3$ could be given by the following columnwise vector-soft threshold function, i.e.,

\[
V_3 = \text{soft}(D_y U_3 - L_3, \frac{\lambda_4}{2\mu_3}),
\]

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

The computational complexity of updating $V_3$ by (51) is $O(n_3(I_3 - 1)(I_3 + 1))$.
3) The $L_3$-subproblem: From (46), we update the Lagrangian multiplier $L_3$ by

$$L_3 = L_3 - (D_y U_3 - V_3).$$

(52)

The computational complexity of updating $L_3$ 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 $O(NCG(n_3^3 I_3 + n_3 I_3^2))$, where $NCG$ is the iteration number of CG.

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

**Algorithm 3 Solving $U_3$-subproblem (18) via ADMM**

**Inputting:** $B, C, U_1, \hat{U}_1, U_2, \hat{U}_2, G, U_3^{pre}$, $\beta > 0$, $\mu_4 > 0$, and $\lambda_4 > 0$.

**Outputting:** Dictionary matrix $U_3$.

**While** not converged **do**

1. **Step 1** Updating the dictionary matrix $U_3$ by (49);
2. **Step 2** Updating the variable $V_3$ by (51);
3. **Step 3** Updating the Lagrangian multiplier $L_3$

by (52);

**end while**

D. The optimization problem of $G$

Problem (20) is convex, hence, we utilize ADMM to solve it. We first introduce the splitting variables $G_1 = G$ and $G_2 = \hat{G}$ and then rewrite the problem (20) as follows,

$$\begin{align*}
\arg\min_{G_1, G_2} f(G) + f_1(G_1) + f_2(G_2), \\
\quad \text{s.t. } G_1 = G, \ G_2 = \hat{G},
\end{align*}$$

(53)

where

$$\begin{align*}
f(G) &= \lambda_1 ||G||_1 + \beta ||G - G^{pre}||_F^2, \\
f_1(G_1) &= ||B - G_1 \times_1 \hat{U}_1 \times_2 \hat{U}_2 \times_3 U_3||_F^2, \\
f_2(G_2) &= ||C - G_2 \times_1 U_1 \times_2 U_2 \times_3 \hat{U}_3||_F^2.
\end{align*}$$

The augmented Lagrangian function of (53) is given by

$$L(G, G_1, G_2, L_4, L_5) = \lambda_1 ||G||_1 + \beta ||G - G^{pre}||_F^2 + ||B - G_1 \times_1 \hat{U}_1 \times_2 \hat{U}_2 \times_3 U_3||_F^2 + \mu_4 ||G - G_1 - L_4||_F^2 + ||C - G_2 \times_1 U_1 \times_2 U_2 \times_3 \hat{U}_3||_F^2 + \mu_4 ||G - G_2 - L_5||_F^2,$$

(54)

where $\mu_4$ represents a penalty parameter, $L_4$ and $L_5$ denote the Lagrangian multipliers.

ADMM iterations of (54) are given by

$$\begin{align*}
G^{(t+1)} &= \arg\min_{G} L(G, G_1^{(t)}, G_2^{(t)}, L_4^{(t)}, L_5^{(t)}), \\
G_1^{(t+1)} &= \arg\min_{G_1} L(G^{(t+1)}, G_1, G_2^{(t)}, L_4^{(t)}, L_5^{(t)}), \\
G_2^{(t+1)} &= \arg\min_{G_2} L(G^{(t+1)}, G_1^{(t+1)}, G_2, L_4^{(t+1)}, L_5^{(t+1)}), \\
L_4^{(t+1)} &= \arg\min_{L_4} L(G^{(t+1)}, G_1^{(t+1)}, G_2^{(t+1)}, L_4, L_5^{(t+1)}), \\
L_5^{(t+1)} &= \arg\min_{L_5} L(G^{(t+1)}, G_1^{(t+1)}, G_2^{(t+1)}, L_4^{(t+1)}, L_5).
\end{align*}$$

(55)

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

1) The $G$-subproblem: From (54), we have

$$\begin{align*}
\arg\min_{G} & \lambda_1 ||G||_1 + \beta ||G - G^{pre}||_F^2 + \mu_4 ||G - G_1 - L_4||_F^2 + \mu_4 ||G - G_2 - L_5||_F^2, \\
\leq & \frac{\lambda_1}{2 \mu_4 + \beta}, \frac{4 \mu_4 + 2 \beta}{4 \mu_4 + 2 \beta},
\end{align*}$$

(56)

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

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

2) The $G_1$-subproblem: From (54), we have

$$\begin{align*}
\arg\min_{G_1} & \mu_4 ||G_1 - \hat{G} + L_4||_F^2 + ||B - G_1 \times_1 \hat{U}_1 \times_2 \hat{U}_2 \times_3 U_3||_F^2, \\
\leq & \frac{\mu_4}{\lambda_1} ||G_1 - \hat{G} + L_4||_F^2 + ||B - G_1 \times_1 \hat{U}_1 \times_2 \hat{U}_2 \times_3 U_3||_F^2.
\end{align*}$$

(58)

Problem (58) is equal to

$$\begin{align*}
\arg\min_{G_1} & \mu_4 ||G_1 - \hat{G} + L_4||_F^2 + ||B - G_1 \times_1 \hat{U}_1 \times_2 \hat{U}_2 \times_3 U_3||_F^2,
\end{align*}$$

(59)

where the vectors $g_1 = \text{vec}(G_1)$, $l_4 = \text{vec}(L_4)$, $g = \text{vec}(G)$, and $b = \text{vec}(B)$ are generated by vectorizing the tensors $G_1$, $L_4$, $G$ and $B$, respectively, and $M_1 = U_3 \otimes \hat{U}_2 \otimes \hat{U}_1$.

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

$$g_1 = (M_1^T M_1 + \mu_4 I)^{-1} (M_1^T b + \mu_4 g - \mu_4 l_4).$$

(60)

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

$$(M_1^T M_1 + \mu_4 I)^{-1} = (D_3 \otimes D_2 \otimes D_1) (\Sigma_3 \otimes \Sigma_2 \otimes \Sigma_1 + \mu_4 I)^{-1} \times (D_3^T \otimes D_2^T \otimes D_1^T),$$

(61)

where $\Sigma_i$ and $D_i$ (i=1, 2, 3) are diagonal matrices and unitary matrices containing the eigenvalues and eigenvectors of, respectively, $\hat{U}_1^T \hat{U}_1$, $\hat{U}_2^T \hat{U}_2$, and $U_3^T U_3$. Therefore, $(\Sigma_3 \otimes \Sigma_2 \otimes \Sigma_1 + \mu_4 I)^{-1}$ is a diagonal matrix and could be computed easily. Besides, the term $M_1^T b$ in (60) can be computed by

$$M_1^T b = \text{vec}(B \times_1 \hat{U}_1^T \times_2 \hat{U}_2^T \times_3 U_3^T),$$

(62)

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

Therefore, we could compute (60) easily.
3) The $G_2$-subproblem: From (54), we have

$$\begin{align*}
\text{argmin}_{\hat{g}_2} \, & \mu_4 \| \hat{g}_2 - G + L_5 \|_2^2 + \\
& \| C - G_2 \times 1 \, U_1 \times 1 \, U_2 \times 3 \, \hat{U}_3 \|_2^2.
\end{align*}$$

(63)

Problem (63) is equal to

$$\begin{align*}
\text{argmin}_{\hat{g}_2} \, & \mu_4 \| \hat{g}_2 - g + l_5 \|_2^2 + \| C - M_2 \hat{g}_2 \|_2^2,
\end{align*}$$

(64)

where the vectors $g_2 = \text{vec}(G_2)$, $l_5 = \text{vec}(L_5)$, $g = \text{vec}(G)$, and $c = \text{vec}(C)$ are obtained by vectorizing the tensors $G_2$, $L_5$, $G$ and $C$, respectively, and $M_2 = \hat{U}_3 \otimes U_2 \otimes U_1$.

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

$$\hat{g}_2 = (M_2^T M_2 + \mu_4 I)^{-1}(M_2^T c + \mu_4 g - \mu_4 l_5).$$

(65)

Note that $M_2 \in \mathbb{R}^{l_1 l_2 l_3 \times n_1 n_2 n_3}$ is relatively large, therefore, it is difficult to solve (65). Fortunately, similar to (61), we have

$$(M_2^T M_2 + \mu_4 I)^{-1} = (D_3 \otimes D_2 \otimes D_1)(\Sigma_3 \otimes \Sigma_2 \otimes \Sigma_1 + \mu_4 I)^{-1}(D_3^T \otimes D_2^T \otimes D_1^T),$$

(66)

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

4) The $L_4$ and $L_5$-subproblem: From (54), we update the Lagrangian multipliers $L_4$ and $L_5$ by

$$L_4 = L_4 - (G - G_1),$$

(67)

$$L_5 = L_5 - (G - G_2).$$

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 $O(n_1^3 n_2^2 n_3 + n_1 n_2^3 n_3 + n_1 n_2 n_3^3)$.

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

---

### References


Ting Xu received the B.S. degree in mathematics and applied mathematics from the Manyang 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.

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.