A Generalized Model for Robust Tensor Factorization With Noise Modeling by Mixture of Gaussians

Xi'ai Chen, Zhi Han, *Member, IEEE*, Yao Wang, Qian Zhao, Deyu Meng, *Member, IEEE*,
Lin Lin, and Yandong Tang, *Member, IEEE*

***Abstract*— The low-rank tensor factorization (LRTF) technique has received increasing attention in many computer vision applications. Compared with the traditional matrix factorization technique, it can better preserve the intrinsic structure information and thus has a better low-dimensional subspace recovery performance. Basically, the desired low-rank tensor is recovered by minimizing the least square loss between the input data and its factorized representation. Since the least square loss is most optimal when the noise follows a Gaussian distribution, L_1-norm-based methods are designed to deal with outliers. Unfortunately, they may lose their effectiveness when dealing with real data, which are often contaminated by complex noise. In this paper, we consider integrating the noise modeling technique into a generalized weighted LRTF (GWLRTF) procedure. This procedure treats the original issue as an LRTF problem and models the noise using a mixture of Gaussians (MoG), a procedure called MoG GWLRTF. To extend the applicability of the model, two typical tensor factorization operations, i.e., CANDECOMP/PARAFAC factorization and Tucker factorization, are incorporated into the LRTF procedure. Its parameters are updated under the expectation–maximization framework. Extensive experiments indicate the respective advantages of these two versions of MoG GWLRTF in various applications and also demonstrate their effectiveness compared with other competing methods.**

***Index Terms*— Expectation–maximization (EM) algorithm, generalized weighted low-rank tensor factorization (GWLRTF), mixture of Gaussians (MoG) model, tensor factorization.**

I. INTRODUCTION

IN RECENT years, the data to be processed have become increasingly massive and high dimensional. The problem of how to learn the low-dimensional subspaces from such high-dimensional data has arisen in many areas, such as computer vision, machine learning, and data mining. To solve this issue, efforts have been made in many applications, e.g., face recognition with robustness to varying expressions, illumination and occlusion [1]–[4], camera motion and scene geometry recovery from image sequences [5], 3-D object recognition and pose estimation from its image appearance [6], and segmentation of multiple rigid-body motions from point correspondences [7]. Traditionally, by employing the matricizing technique, matrix-based approaches, including low-rank matrix factorization (LRMF) and robust principal component analysis (RPCA) [8], are used to deal with these applications. However, in these applications, the data to be analyzed have intrinsically high-dimensional multilinear structures, and these matrix-based approaches fail to exploit the structure information, as mentioned in [9]. Therefore, it would be more reasonable to formulate the data as higher order tensors, which can better preserve the intrinsic structure. Moreover, a higher order tensor can be viewed as the natural generalization of a vector and matrix. To better illustrate the procedure of our tensor method and how it differs from the matrix technique, Fig. 1 provides a framework of these two techniques when dealing with higher order data. Compared with the matrix technique, which needs to first matricize the original higher order data to obtain the corresponding matrix form, the tensor technique is directly applied to its high-order tensor form.

The low-rank tensor factorization (LRTF) is an effective way to learn the low-dimensional subspace of a higher order tensor. Unlike the LRMF framework, which has the ability to obtain a unique factorization and rank definition, the corresponding forms in the tensor case are nonunique. Basically, there are two definitions of tensor factorizations: CANDECOMP/PARAFAC (CP) factorization and Tucker factorization.

CP factorization can be regarded as a higher order generalization of the matrix singular value decomposition (SVD) [10]–[12]. It has already been used in many areas, such as image inpainting [13], [14], collaborative filtering [15], and data mining [16]. The idea of CP factorization is to

Manuscript received January 17, 2017; revised July 25, 2017 and January 11, 2018; accepted January 16, 2018. Date of publication March 1, 2018; date of current version October 16, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61773367, Grant 61303168, Grant 11501440, Grant 61333019, and Grant 61373114, in part by the Youth Innovation Promotion Association of the Chinese Academy of Sciences under Grant 2016183, in part by the China Postdoctoral Science Foundation under Grant 2017M610628, and in part by the Key Research Program of Hunan Province, China, under Grant 2017GK2273. *(Corresponding author: Zhi Han.)*

X. Chen is with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: chenxiai@sia.cn).

Z. Han and Y. Tang are with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China (e-mail: hanzhi@sia.cn; ytang@sia.cn).

Y. Wang, Q. Zhao, D. Meng, and L. Lin are with the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: yao.s.wang@gmail.com; timmy.zhaoqian@gmail.com; 610674737@qq.com; dymeng@mail.xjtu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2018.2796606

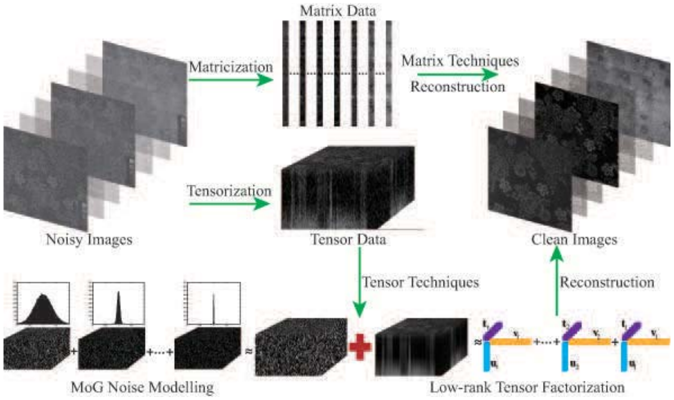


Fig. 1. Framework of higher order data processed by our tensor technique and the traditional matrix technique. Compared with the matrix technique, which needs to first matricize the original data to obtain the corresponding matrix data, the tensor technique is directly applied to its high-order tensor form. In our tensor method, mixture of Gaussians (MoG) noise modeling is incorporated into the procedure of LRTF. Here, CP is taken for illustration.

represent a tensor as the summation of a finite number of rank-1 tensors, and the corresponding rank is defined as the smallest number of rank-1 tensors [17], [18]. Mathematically, assuming an N-order tensor $\mathcal{X}$ of size $I_1 \times I_2 \times \cdots \times I_N$, the CP factorization is defined as

$$\mathcal{X} = \sum_{d=1}^{r} \mathbf{u}_d \circ \mathbf{v}_d \circ \cdots \circ \mathbf{t}_d \tag{1}$$

where $\circ$ and r are the vector outer product and the rank of tensor $\mathcal{X}$, respectively. The mode matrix is defined as the combination of vectors in the rank-1 tensors, e.g., $U = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r]$; the same is true for the others. Accordingly, the elements in tensor CP factorization can be represented as

$$x_{i_1,i_2,\ldots,i_N} = \sum_{d=1}^{r} u_{i_1,d} v_{i_2,d} \ldots t_{i_N,d}. \tag{2}$$

Tucker factorization can be viewed as a form of higher order principal component analysis [10]–[12], [19], and it has been widely used in many applications [20]–[26]. The tensor Tucker factorization is achieved by multiplying a core tensor by its mode matrices, that is,

$$\begin{aligned}\mathcal{X} &= \mathcal{G} \times_1 U \times_2 V \cdots \times_N T \\ &= \sum_{d_1=1}^{r_1} \sum_{d_2=1}^{r_2} \cdots \sum_{d_N=1}^{r_N} g_{d_1,d_2,\ldots,d_N} \mathbf{u}_{d_1} \circ \mathbf{v}_{d_2} \circ \cdots \circ \mathbf{t}_{d_N}\end{aligned} \tag{3}$$

where $\times_n$ denotes the n-mode matrix product and $\mathcal{G}$ is the core tensor, which is used to control the interaction between the mode matrices. The rank of Tucker factorization is defined by the n-rank of $\mathcal{X}$, denoted as $\text{rank}_n(\mathcal{X})$. It is composed of the mode-n rank r_n of the tensor. Therefore, tensor $\mathcal{X}$ is also called a rank-$(r_1, r_2, \ldots, r_N)$ tensor. Elementwise, the Tucker factorization in (3) is

$$\begin{aligned}&x_{i_1,i_2,\ldots,i_N} \\ &\quad= \sum_{d_1=1}^{r_1} \sum_{d_2=1}^{r_2} \cdots \sum_{d_N=1}^{r_N} g_{d_1,d_2,\ldots,d_N} u_{i_1,d_1} v_{i_2,d_2} \ldots t_{i_N,d_N}.\end{aligned} \tag{4}$$

Traditionally, the Frobenius norm function (i.e., the L_F-norm) is employed to perform LRTF. It is effective when the noise variable follows a Gaussian distribution according to the maximum likelihood estimation. However, the Frobenius-norm-function-based LRTF approach loses its effectiveness when the data are corrupted by gross outliers, which affect only a small fraction of the data but are large in magnitude. This kind of noise has arisen in many real applications, such as functional magnetic resonance imaging neuroimaging [27] and video surveillance [28]. Commonly, the L_1-norm [29], [30] is an alternative to the L_F-norm when dealing with gross outliers. From the perspective of modeling, the observation model is represented as a superposition of the low-rank term, the sparse corruption term, or/and the Gaussian noise term, which was widely considered in [8] and [31]–[33]. It works well only when the noise follows a simple mixture of the Gaussian distribution and the Laplacian distribution. Unfortunately, the noise often follows very complex statistical distributions in many real applications [34]–[36], which motivates us to introduce a more robust noise modeling technique into the LRTF procedure.

Considering that an MoG can be used to perform a universal approximation of a continuous distribution, Meng and De la Torre [34] considered applying the MoG noise model to the LRMF framework. Zhao *et al.* [35] further introduced it into the RPCA model under the Bayesian framework. Inspired by the fine results they obtained, along this line, we introduce the MoG noise model into the framework of low-rank tensor CP factorization [37], which can be regarded as a higher order generalization of the LRMF case. To better explore the underlying structure of tensors and further enhance the applicability of the model, in this paper, we apply the MoG noise model to the two typical tensor factorization strategies and derive a generalized model for low-rank subspace estimation.

The major contributions of this paper are summarized as follows: 1) MoG distribution is introduced into the framework of LRTF to model complex noise; 2) a generalized weighted LRTF (GWLRTF) is proposed for low-rank subspace learning, i.e., the GWLRTF integrated with CP (GWLRTF-CP) and the GWLRTF integrated with Tucker (GWLRTF-Tucker); 3) the parameters of the MoG are calculated using the maximum log-likelihood function, and the low-rank subspace parameters are estimated by solving the GWLRTF model; and 4) the effectiveness of MoG GWLRTF compared with other competing methods is validated via extensive experiments, with the two proposed factorization versions exhibiting different advantages in various applications. The source codes of this paper can be obtained at *http://vision.sia.cn/our%20team/Hanzhi-homepage/vision-ZhiHan(English).html*.

The remainder of this paper is organized as follows. Section II introduces the notations and common tensor operations used in this paper. In Section III, a generalized low-rank tensor model integrated with the MoG is introduced. The corresponding algorithms are derived under the expectation–maximization (EM) framework in Section IV. Section V demonstrates the results of extensive experiments. The conclusion and discussion are provided in Section VI.

II. NOTATIONS AND PRELIMINARIES

In this section, we introduce the notations and common tensor operations used throughout this paper. Lowercase letters $(a, b, \ldots)$ and their bold forms $(\mathbf{a}, \mathbf{b}, \ldots)$ are used to denote scalars and vectors, respectively. Uppercase letters $(A, B, \ldots)$ and their corresponding calligraphic forms $(\mathcal{A}, \mathcal{B}, \ldots)$ denote the matrices and higher order tensors, respectively. Vector and matrix elements are represented as $(a_i, b_j, \ldots)$ and $(a_{i,j}, b_{i,j}, \ldots)$, respectively.

Assuming an N-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, the positive integer $I_n (n = 1, 2, \ldots N)$ is used to denote the size of each order (or dimension or mode). An element of the tensor can be expressed as $x_{i_1,\ldots,i_n,\ldots,i_N}$. The tensor slices can be obtained by fixing every index but two. The mode matrix of a tensor is obtained by unfolding the tensor along the corresponding mode, which is denoted as $\mathcal{X}_{(n)} = \text{unfold}_n(\mathcal{X})$. Its inverse operation is the mode-n folding, represented as $\mathcal{X} = \text{fold}_n(\mathcal{X}_{(n)})$. The mode-$n$ rank of $\mathcal{X}$ is defined by its mode-n matrix, i.e., $\mathcal{X}_{(n)} : r_n = \text{rank}(\mathcal{X}_{(n)})$. The matrix vectorizing operation is defined as $\text{vec} : \mathbb{R}^{w \times h} \rightarrow \mathbb{R}^{wh}$.

Mathematically, a rank-1 tensor is defined by the outer product of N vectors as follows: $\mathcal{X} = \mathbf{u} \circ \mathbf{v} \circ \cdots \circ \mathbf{t}$, with elements $x_{i_1,i_2,\ldots,i_N} = u_{i_1} v_{i_2} \ldots t_{i_N}$. The mode-$n$ product of tensor $\mathcal{X}$ and its mode matrix U is defined by $\mathcal{X} \times_n U \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N}$ with element $(\mathcal{X} \times_n U)_{i_1,\ldots,i_{n-1},j_n,i_{n+1},\ldots,i_N} = \sum_{i_n} x_{i_1,\ldots,i_n,\ldots,i_N} u_{j_n,i_n}$. The inner product of two tensors with the same size can be represented as $\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} x_{i_1,\ldots,i_N} y_{i_1,\ldots,i_N}$. The three commonly used types of norm, i.e., L_F-norm, L_0-norm, and L_1-norm, are defined as: $\|\mathcal{X}\|_F = \langle \mathcal{X}, \mathcal{X} \rangle^{1/2}$. $\|\mathcal{X}\|_0$ counts the number of nonzero entries in $\mathcal{X}$, and $\|\mathcal{X}\|_1 = \sum_{i_1,\ldots,i_N} |x_{i_1,\ldots,i_N}|$.

III. GENERALIZED LOW-RANK TENSOR MODEL INTEGRATED WITH MOG

In this section, we propose a generalized low-rank tensor model integrated with the MoG for modeling complex noise. To optimize the parameters, we aim to maximize the log-likelihood form of the model.

Considering the noise case (denoted as $\mathcal{E}$), the observed tensor $\mathcal{X}$ can be modeled as

$$\mathcal{X} = \mathcal{L} + \mathcal{E} \tag{5}$$

where $\mathcal{L}$ denotes the low-rank tensor, the corresponding elementwise form of which is

$$x_{i_1,i_2,\ldots,i_N} = l_{i_1,i_2,\ldots,i_N} + \varepsilon_{i_1,i_2,\ldots,i_N}. \tag{6}$$

Suppose that each noise element in (5) follows the MoG distribution, which is defined as:

$$p(\varepsilon) \sim \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\varepsilon | \mu_k, \sigma_k^2\big) \tag{7}$$

where $\{\pi_k | \pi_k \geq 0, \sum_{k=1}^{K} \pi_k = 1\}$ is the mixing proportion and (μ_k, σ_k) are the Gaussian distribution parameters. Each element $x_{i_1,i_2,\ldots,i_N}$ of $\mathcal{X}$ follows:

$$p(x_{i_1,i_2,\ldots,i_N} | \Pi, \Lambda, \Sigma) = \sum_{k=1}^{K} \pi_k \mathcal{N}\big(x_{i_1,i_2,\ldots,i_N} | \Lambda_k, \sigma_k^2\big) \tag{8}$$

where $\Pi = \{\pi_1, \pi_2, \ldots, \pi_K\}, \Lambda = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_K\}$, $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_K\}$, and $\Lambda_k = l_{i_1,i_2,\ldots,i_N} + \mu_k$. Correspondingly, we have the following likelihood form of the observed tensor $\mathcal{X}$:

$$p(\mathcal{X} | \Pi, \Lambda, \Sigma) = \prod_{i_1,i_2,\ldots,i_N \in \Omega} \sum_{k=1}^{K} \pi_k \mathcal{N}\big(x_{i_1,i_2,\ldots,i_N} | \Lambda_k, \sigma_k^2\big) \tag{9}$$

where Ω is used to indicate the nonmissing entries in $\mathcal{X}$.

Alternatively, we choose to maximize the logarithmic form of (9) as follows:

$$\begin{aligned}(\Pi^*, \Lambda^*, \Sigma^*) &= \underset{\Pi,\Lambda,\Sigma}{\text{argmax}} \ \log p(\mathcal{X} | \Pi, \Lambda, \Sigma) \\ &= \underset{\Pi,\Lambda,\Sigma}{\text{argmax}} \sum_{i_1,i_2,\ldots,i_N \in \Omega} \log \sum_{k=1}^{K} \pi_k \mathcal{N}\big(x_{i_1,i_2,\ldots,i_N} | \Lambda_k, \sigma_k^2\big). \end{aligned} \tag{10}$$

Assuming that the mean value in (7) is zero, the original problem becomes a Gaussian scale mixture problem [38], [39]. Thus, our goal is to maximize

$$\begin{aligned}(\Pi^*, \mathcal{L}^*, \Sigma^*) &= \underset{\Pi,\Lambda,\Sigma}{\text{argmax}} \ \log p(\mathcal{X} | \Pi, \mathcal{L}, \Sigma) \\ &= \underset{\Pi,\Lambda,\Sigma}{\text{argmax}} \sum_{i_1,i_2,\ldots,i_N \in \Omega} \log \sum_{k=1}^{K} \pi_k \mathcal{N}\big(x_{i_1,i_2,\ldots,i_N} | l_{i_1,i_2,\ldots,i_N}, \sigma_k^2\big). \end{aligned} \tag{11}$$

IV. ALGORITHMS UNDER EM FRAMEWORK

It is known that an effective method for solving the problem of the maximum log-likelihood function is the EM method [40]. Therefore, by assuming a higher order latent variable, we employ the EM method to solve (10). The parameters of the MoG model are estimated using the maximum log-likelihood method under the EM framework, and the low-rank subspace parameters are computed using the newly developed algorithms. Thus, the whole algorithm is achieved by iteratively updating between the computation of Gaussian responsibility (**E Step**) and the estimation of parameters $\Pi, \Sigma, \mathcal{L}$ in the model (**M Step**).

E Step: Assuming that higher order latent variable $\{z_{i_1,i_2,\ldots,i_N,k} | z_{i_1,i_2,\ldots,i_N,k} \in \{0, 1\}, \sum_{k=1}^{K} z_{i_1,i_2,\ldots,i_N,k} = 1\}$ represents the assigned value of each MoG distribution, the posterior responsibility of the kth Gaussian mixture for generating $x_{i_1,i_2,\ldots,i_N}$ can be computed as

$$\begin{aligned}E(z_{i_1,i_2,\ldots,i_N,k}) &= \gamma_{i_1,i_2,\ldots,i_N,k} \\ &= \frac{\pi_k \mathcal{N}(x_{i_1,i_2,\ldots,i_N} | l_{i_1,i_2,\ldots,i_N}, \sigma_k^2)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_{i_1,i_2,\ldots,i_N} | l_{i_1,i_2,\ldots,i_N}, \sigma_k^2)}. \end{aligned} \tag{12}$$

M Step: Maximize the upper bound given in the **E Step** with respect to parameters $\mathcal{L}$, Π, and Σ

$$\begin{aligned} &E_Z p(\mathcal{X}, Z|\mathcal{L}, \Pi, \Sigma)\\ &\quad = \sum_{i_1,i_2,\ldots,i_N\in\Omega}\sum_{k=1}^{K}\gamma_{i_1,i_2,\ldots,i_N,k}\\ &\quad\quad \times\left(\log\pi_k - \log\sqrt{2\pi\sigma_k} - \frac{(x_{i_1,i_2,\ldots,i_N}-l_{i_1,i_2,\ldots,i_N})^2}{2\pi\sigma_k^2}\right). \end{aligned} \tag{13}$$

The maximization procedure for this problem can be alternatively achieved by iteratively computing the MoG parameters and estimating the low-rank tensor in the model.

Update Π, Σ*:* The MoG parameters have the following closed-form solutions:

$$\begin{aligned} m_k &= \sum_{i_1,i_2,\ldots,i_N}\gamma_{i_1,i_2,\ldots,i_N,k},\ \pi_k = \frac{m_k}{\sum_k m_k}\\ \sigma_k^2 &= \frac{1}{m_k}\sum_{i_1,i_2,\ldots,i_N}\gamma_{i_1,i_2,\ldots,i_N,k}(x_{i_1,i_2,\ldots,i_N}-l_{i_1,i_2,\ldots,i_N})^2. \end{aligned} \tag{14}$$

Update $\mathcal{L}$*:* Once the MoG parameters are known, (13) can be rewritten with respect to $\mathcal{L}$ as

$$\begin{aligned} &\sum_{i_1,i_2,\ldots i_N\in\Omega}\sum_{k=1}^{K}\gamma_{i_1,i_2,\ldots i_N,k}\left(-\frac{(x_{i_1,i_2,\ldots,i_N}-l_{i_1,i_2,\ldots,i_N})^2}{2\pi\sigma_k^2}\right)\\ &= -\sum_{i_1,i_2,\ldots,i_N\in\Omega}\sum_{k=1}^{K}\left(\frac{\gamma_{i_1,i_2,\ldots,i_N,k}}{2\pi\sigma_k^2}\right)(x_{i_1,i_2,\ldots,i_N}-l_{i_1,i_2,\ldots,i_N})^2\\ &= -\|\mathcal{W}\odot(\mathcal{X}-\mathcal{L})\|_{L_F}^2 \end{aligned} \tag{15}$$

where $\mathcal{L}$ is evaluated by solving the GWLRTF model $\min_{\mathcal{L}}\|\mathcal{W}\odot(\mathcal{X}-\mathcal{L})\|_{L_F}^2$. We use $\odot$ to denote the Hadamard product (componentwise multiplication). The element $w_{i_1,i_2,\ldots,i_N}$ in weighted tensor $\mathcal{W}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ is computed using

$$w_{i_1,i_2,\ldots,i_N} = \begin{cases}\sqrt{\displaystyle\sum_{k=1}^{K}\frac{\gamma_{i_1,i_2,\ldots,i_N,k}}{2\pi\sigma_k^2}}, & i_1,i_2,\ldots,i_N\in\Omega\\ 0, & i_1,i_2,\ldots,i_N\notin\Omega.\end{cases} \tag{16}$$

The whole procedure for solving the proposed model is summarized in Algorithm 1.

In this paper, the two typical factorizations are introduced into the GWLRTF model, and the corresponding optimization algorithms are proposed.

A. GWLRTF Integrated With CP

Assuming that tensor $\mathcal{X}$ has a size of $\mathbb{R}^{I\times J\times K}$ in the GWLRTF model, the low-rank CP factorization form is

$$\min_{U,V,\,T}\left\|\mathcal{W}\odot\left(\mathcal{X}-\sum_{d=1}^{r}\mathbf{u}_{:,d}\circ\mathbf{v}_{:,d}\circ\mathbf{t}_{:,d}\right)\right\|_{L_F} \tag{17}$$

where $\mathcal{L}=\sum_{d=1}^{r}\mathbf{u}_{:,d}\circ\mathbf{v}_{:,d}\circ\mathbf{t}_{:,d}$ and $U\in\mathbb{R}^{I\times r}$, $V\in\mathbb{R}^{J\times r}$, and $T\in\mathbb{R}^{K\times r}$ are mode matrices with rank r. The element of weighted tensor $\mathcal{W}$ is dependent on the above-obtained standard variance.

Algorithm 1 MoG GWLRTF

```
Input:  observed data in tensor form $\mathcal{X}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$
Output: the recovered low-rank tensor $\mathcal{L}$
Initialize the parameters $\Pi, \Sigma, \mathcal{L}$ in the model, and preset the number of MoGs $K$, the iteration and the value of threshold $\epsilon$.
while not converged do
    E Step for Gaussian responsibility:
    Evaluate $\gamma_{i_1,i_2,\cdots,i_N,k}$ using Eq. (12).
    M Step for MoG parameters $\Pi, \Sigma$:
    Evaluate $\pi_k, \sigma_k^2$ using Eq. (14)
    M Step for low-rank tensor $\mathcal{L}$:
    Estimate $\mathcal{L}$ by solving the following GWLRTF model
    $\min_{\mathcal{L}}\|\mathcal{W}\odot(\mathcal{X}-\mathcal{L})\|_{L_F}^2$,
    in which $\mathcal{W}$ is calculated using Eq. (16).
end while
```

We adopt the alternating least squares (ALS) technique to iteratively update the parameters of the low-rank tensor. Considering that the low-rank tensor is the summation of finite rank-1 tensors, the tensor slice can be represented by the linear combination of the corresponding slice of the finite rank-1 tensors. We use matrices M_F, M_H, and M_L to denote the vectorized form of tensor frontal, horizontal, and lateral slices.

The first term in the GWLRTF-CP model can be represented by

$$\mathcal{X}^{\text{weight}} = \mathcal{W}\odot\mathcal{X}. \tag{18}$$

To update mode matrix T, we first rewrite (18) in the vectorized form

$$M_F = \left[\operatorname{vec}\left(X^{\text{weight}}_{:,:,1}\right)|\ldots|\operatorname{vec}\left(X^{\text{weight}}_{:,:,K}\right)\right]\in\mathbb{R}^{IJ\times K}. \tag{19}$$

The ith vectorized frontal slice of the second term in the GWLRTF-CP model with respect to mode matrix T can be expressed as

$$\begin{aligned} F_i = \Big[&\operatorname{vec}\left(W_{:,:,i}\odot\left(\mathbf{u}^{\text{old}}_{:,1}\circ\mathbf{v}^{\text{old}}_{:,1}\right)\right)|\\ &\ldots|\operatorname{vec}\left(W_{:,:,i}\odot\left(\mathbf{u}^{\text{old}}_{:,r}\circ\mathbf{v}^{\text{old}}_{:,r}\right)\right)\Big]\in\mathbb{R}^{IJ\times r}. \end{aligned} \tag{20}$$

The ith vector of mode matrix T can be updated using

$$T^{\text{new}}_{i,:} = \left(F_i^{\dagger}M_{F\,:,i}\right)^T\in\mathbb{R}^{1\times r} \tag{21}$$

where † and T denote the pseudoinverse form and the transposed form of the matrix, respectively.

Similarly, we have the ith vector of mode matrices V and U, which are updated as follows:

$$M_L = \left[\operatorname{vec}\left(X^{\text{weight}}_{:,1,:}\right)|\ldots|\operatorname{vec}\left(X^{\text{weight}}_{:,J,:}\right)\right]\in\mathbb{R}^{IK\times J} \tag{22}$$

$$\begin{aligned} L_i = \Big[&\operatorname{vec}\left(W_{:,i,:}\odot\left(\mathbf{t}^{\text{new}}_{:,1}\circ\mathbf{u}^{\text{old}}_{:,1}\right)\right)|\\ &\ldots|\operatorname{vec}\left(W_{:,i,:}\odot\left(\mathbf{t}^{\text{new}}_{:,r}\circ\mathbf{u}^{\text{old}}_{:,r}\right)\right)\Big]\in\mathbb{R}^{IK\times r} \end{aligned} \tag{23}$$

$$V^{\text{new}}_{i,:} = \left(L_i^{\dagger}M_{L\,:,i}\right)^T\in\mathbb{R}^{1\times r} \tag{24}$$

$$M_H = \left[\operatorname{vec}\left(X^{\text{weight}}_{1,:,:}\right)|\ldots|\operatorname{vec}\left(X^{\text{weight}}_{I,:,:}\right)\right]\in\mathbb{R}^{JK\times I} \tag{25}$$

$$\begin{aligned} H_i = \Big[&\operatorname{vec}\left(W_{i,:,:}\odot\left(\mathbf{v}^{\text{new}}_{:,1}\circ\mathbf{t}^{\text{new}}_{:,1}\right)\right)|\\ &\ldots|\operatorname{vec}\left(W_{i,:,:}\odot\left(\mathbf{v}^{\text{new}}_{:,r}\circ\mathbf{t}^{\text{new}}_{:,r}\right)\right)\Big]\in\mathbb{R}^{JK\times r} \end{aligned} \tag{26}$$

$$U^{\text{new}}_{i,:} = \left(H_i^{\dagger}M_{H\,:,i}\right)^T\in\mathbb{R}^{1\times r}. \tag{27}$$

The whole optimization procedure is summarized in Algorithm 2.

Algorithm 2 GWLRTF-CP

```
Input: the original tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ and the weighted tensor $\mathcal{W}$, initialized mode matrices $U, V, T$, the number of iterations and the value of threshold $\epsilon$.
Output: mode matrices $U, V, T$.
while not converged do
    update mode matrix $T$ with Eqs. (19), (20), and (21);
    update mode matrix $V$ with Eqs. (22), (23), and (24);
    update mode matrix $U$ with Eqs. (25), (26), and (27).
end while
```

B. GWLRTF Integrated With Tucker

The GWLRTF model with Tucker factorization has the following form:

$$\min_{\mathcal{G},U,V,\ldots,T} \|\mathcal{W} \odot (\mathcal{X} - \mathcal{G} \times_1 U \times_2 V \cdots \times_N T)\|_{L_F}. \tag{28}$$

We apply the coordinate descent technique to solve this optimization problem, considering its effectiveness in solving the convex optimization problems [41]–[43]. This problem can be reduced to solving a series of scalar minimization subproblems, with the goal of updating the entries of the mode matrices and the core tensor with respect to the coordinates.

1) Update the Mode Matrices $U, V, \ldots, T$: First, we reformulate (28), such that the function is minimized against only one of the unknown mode matrices (take H for example) at a time, with the others fixed, as follows:

$$\|\mathcal{W} \odot (\mathcal{X} - \mathcal{D} \times_n H)\|_{L_F} \tag{29}$$

where

$$\mathcal{D} = \mathcal{G} \times_1 U \times_2 V \cdots \times_{n-1} F \times_{n+1} K \cdots \times_N T. \tag{30}$$

Unfolding the tensors along mode-n, it can be reformulated as the following subproblem:

$$\begin{aligned}
&\|W_{(n)} \odot (X_{(n)} - H D_{(n)})\|_{L_F} \\
&= \left\| W_{(n)} \odot \left(X_{(n)} - \sum_{j=1}^{r_n} \mathbf{h}_{:,j} \mathbf{d}_{:,j}^T \right) \right\|_{L_F} \\
&= \left\| W_{(n)} \odot \left(E - \mathbf{h}_{:,k} \mathbf{d}_{:,k}^T \right) \right\|_{L_F}
\end{aligned} \tag{31}$$

where

$$E = X_{(n)} - \sum_{j \neq k} \mathbf{h}_{:,j} \mathbf{d}_{:,j}^T.$$

Then, the original problem (28) can be separated into the following single-scalar parameter optimization subproblems:

$$\begin{aligned}
&\min_{h_{i,k}} \|\mathbf{w}_{k,:} \odot (\mathbf{e}_{k,:} - \mathbf{d}_{k,:} h_{i,k})\|_{L_F} \\
&\quad = \min_{h_{i,k}} \|\mathbf{w}_{k,:} \odot \mathbf{e}_{k,:} - \mathbf{w}_{k,:} \odot \mathbf{d}_{k,:} h_{i,k}\|_{L_F}.
\end{aligned} \tag{32}$$

Algorithm 3 GWLRTF-Tucker

```
Input: input the original tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and the weighted tensor $\mathcal{W}$, initialized mode matrices $U, V, \ldots, T$ and core tensor $\mathcal{G}$, the number of iterations and the threshold $\epsilon$.
Output: mode matrices $U, V, .., T$ and core tensor $\mathcal{G}$.
while not converged do
    update the entries of mode matrices $U, V, \ldots, T$ by solving Eq. (32);
    update the entries of core tensor $\mathcal{G}$ using Eq. (35);
end while
```

2) Update the Core Tensor $\mathcal{G}$: Likewise, for the equivalent formulation of (28),

$$\left\| \mathcal{W} \odot \left(\mathcal{X} - \sum_{d_1=1}^{r_1} \cdots \sum_{d_N=1}^{r_N} g_{d_1,d_2,\ldots,d_N} \mathbf{u}_{d_1} \circ \mathbf{v}_{d_2} \circ \cdots \circ \mathbf{t}_{d_N} \right) \right\|_{L_F} \tag{33}$$

can be rewritten as

$$\|\mathcal{W} \odot (\mathcal{E} - g_{k_1,\ldots,k_N} \mathcal{U})\|_{L_F} \tag{34}$$

where

$$\begin{aligned}
\mathcal{E} &= \mathcal{X} - \sum_{d_1 \neq k_1,\ldots,d_N \neq k_N} g_{d_1,d_2,\ldots,d_N} \mathbf{u}_{d_1} \circ \mathbf{v}_{d_2} \circ \cdots \circ \mathbf{t}_{d_N} \\
\mathcal{U} &= \mathbf{u}_{k_1} \circ \mathbf{v}_{k_2} \circ \cdots \circ \mathbf{t}_{k_N}.
\end{aligned}$$

Then, (33) can be broken down into the following optimization subproblems:

$$\begin{aligned}
&\min_{g_{k_1,\ldots,k_N}} \|\mathbf{w} \odot (\mathbf{e} - \mathbf{u} g_{k_1,\ldots,k_N})\|_{L_F} \\
&\quad = \min_{g_{k_1,\ldots,k_N}} \|\mathbf{w} \odot \mathbf{e} - \mathbf{w} \odot \mathbf{u} g_{k_1,\ldots,k_N})\|_{L_F}
\end{aligned} \tag{35}$$

where $\mathbf{w} = \text{vec}(\mathcal{W})$, $\mathbf{e} = \text{vec}(\mathcal{E})$, and $\mathbf{u} = \text{vec}(\mathcal{U})$.

Note that the solutions to (32) and (35) can be obtained using the ALS technique. The whole optimization procedure is summarized in Algorithm 3.

3) Termination Conditions: We stop Algorithm 1 when the change in the log-likelihood between the consecutive iterations is smaller than a prespecified small threshold (we set $\epsilon = 1.0e - 50$) or the maximum number of iterations (we set iter $= 15$) is reached. We stop Algorithms 2 and 3 when the change in U between the consecutive iterations is smaller than a prespecified small threshold (we set $\epsilon = 1.0e - 50$) or the maximum number of iterations (we set iter $= 15$) is reached.

V. EXPERIMENTS

Extensive experiments are conducted in this section. All experiments are implemented using a workstation with a 3.6 GHz Intel Core i7 CPU and 24 GB of RAM. The supporting software is MATLAB R2012a. The MATLAB Tensor Toolbox [44] is employed to perform tensor computations. The proposed MoG-based tensor factorization method (MoG GWLRTF) is compared with the following methods: 1) MoG-based LRMF—MoG LRMF [34]; 2) L_2-norm-based low-rank tensor completion and recovery—high accuracy

TABLE I

COMPLETION PERFORMANCE OF COMPETING METHODS WITH VARYING MISSING RATIO. THE FIRST AND SECOND BEST RESULTS ARE HIGHLIGHTED AND UNDERLINED, RESPECTIVELY

		MoG LRMF	HaLRTC	LRTA	PARAFAC	MSI DL	CWM LRTF	MoG GWLRTF-CP	MoG GWLRTF-Tucker
Missing 20%	E1	0.08	2.55e+02	4.40e+02	4.38e+02	2.98e+02	3.51e+02	1.61e-08	**2.31e-12**
	E2	1.08e-04	6.50e+04	6.47e+02	6.47e+02	3.70e+02	3.93e+02	6.82e-19	**1.43e-26**
	E3	0.09	2.68e+04	6.07e+02	5.95e+02	4.10e+02	4.63e+02	2.03e-08	**2.92e-12**
	E4	0.57	3.76e+06	9.62e+02	9.48e+02	5.82e+02	5.34e+02	8.85e-19	**1.85e-26**
Missing 40%	E1	1.24	2.55e+02	5.28e+02	5.30e+02	3.58e+02	4.38e+02	8.84e-09	**1.10e-12**
	E2	0.02	6.50e+04	1.24e+03	1.23e+03	7.95e+02	8.07e+02	3.18e-19	**3.82e-27**
	E3	7.36e+02	5.20e+04	9.93e+02	9.78e+02	6.54e+02	8.00e+02	1.90e-08	**2.07e-12**
	E4	1.41e+02	7.06e+06	2.28e+03	2.22e+03	1.50e+03	1.56e+03	9.63e-19	**8.95e-27**
Missing 60%	E1	5.01	2.55e+02	7.11e+02	6.63e+02	4.85e+02	5.57e+02	**2.49e-07**	8.84e-07
	E2	0.66	6.50e+04	2.34e+03	2.21e+03	1.67e+03	1.90e+03	2.77e-16	**3.59e-27**
	E3	1.51e+04	7.89e+04	1.86e+03	1.81e+03	1.25e+03	1.73e+03	1.12e-06	**2.48e-12**
	E4	1.57e+03	1.09e+07	6.26e+03	6.21e+03	4.32e+03	6.22e+03	4.21e-15	**1.24e-26**

TABLE II

RECONSTRUCTION RESULTS OBTAINED BY COMPETING METHODS WITH DIFFERENT NOISES. THE FIRST AND SECOND BEST RESULTS ARE HIGHLIGHTED AND UNDERLINED, RESPECTIVELY

		MoG LRMF	HaLRTC	LRTA	PARAFAC	MSI DL	CWM LRTF	MoG GWLRTF-CP	MoG GWLRTF-Tucker
Gaussian Noise	E1	32.3	2.55e+02	6.31e+02	6.41e+02	4.42e+02	4.69e+02	54.2	**14.7**
	E2	2.44	6.50e+04	1.55e+03	1.57e+03	1.17e+03	7.04e+02	5.84	**1.62**
	E3	**11.4**	1.05e+05	8.59e+02	8.53e+02	6.02e+02	6.09e+02	29.2	14.8
	E4	72.0	1.41e+07	2.10e+03	2.07e+03	1.53e+03	8.81e+02	1.52	**0.647**
Sparse Noise	E1	**4.20e+02**	5.10e+02	1.05e+03	9.63e+02	7.70e+02	8.86e+02	6.93e+02	4.36e+02
	E2	**4.96e+02**	1.30e+05	2.96e+03	2.63e+03	2.24e+03	2.47e+03	1.42e+03	1.25e+03
	E3	5.25e+03	1.02e+05	1.02e+03	1.05e+03	7.47e+02	8.41e+02	**5.10e+02**	6.82e+02
	E4	1.04e+03	1.33e+07	2.17e+03	2.34e+03	1.64e+03	1.77e+03	**4.33e+02**	1.30e+03
Mixture Noise	E1	4.63e+02	5.10e+02	1.19e+03	1.15e+03	8.17e+02	1.07e+03	6.68e+02	**4.46e+02**
	E2	**6.05e+02**	1.30e+05	3.90e+03	3.70e+03	2.58e+03	3.31e+03	1.37e+03	1.29e+03
	E3	1.23e+04	1.06e+05	1.13e+03	1.17e+03	7.85e+02	1.10e+03	**4.59e+02**	4.62e+02
	E4	1.46e+03	1.50e+07	2.84e+03	2.94e+03	1.93e+03	2.91e+03	**3.83e+02**	5.72e+02

low-rank tensor completion (HaLRTC) [9]; 3) L_2-norm-based low-rank tensor CP factorization—PARAFAC [45]; 4) L_2-norm-based low-rank tensor Tucker factorization—low-rank tensor approximation approximation (LRTA) [46] and MSI DL [47], and the MSI DL method also utilizes the nonlocal prior for better recovery; and 5) L_1-norm-based low-rank tensor Tucker factorization—cyclic weighted median low-rank tensor factorization (CWM LRTF) [41].

Specifically, the results of MoG GWLRTF-Tucker and MoG GWLRTF-CP are listed to carry out a performance comparison regarding various applications.

We conduct numerical experiments on synthetic data and real applications, including single RGB image reconstruction, face modeling, multispectral image recovery, real hyperspectral image restoration, and heavy-tailed noise removal.

A. Synthetic Data

Considering a tensor $\mathcal{X}_{\text{gt}}$ of size $10 \times 10 \times 10$ and rank 5, we generate mode matrices $U \in \mathbb{R}^{10\times5}$, $V \in \mathbb{R}^{10\times5}$, and $T \in \mathbb{R}^{10\times5}$ independent of the standard normal distribution $\mathcal{N}(0, I_{\mathbb{R}})$ and set $\mathcal{X}_{\text{gt}} = [\![U, V, T]\!]$. The following four situations are considered.

1) Vary missing ratio $\rho_m = 0.2$, $\rho_m = 0.4$, and $\rho_m = 0.6$.
2) Fix missing ratio $\rho_m = 0.2$ and Gaussian noise $\mathcal{N}(0, 0.1)$ with scale ratio $\rho_s = 0.2$.
3) Fix missing ratio $\rho_m = 0.2$ and sparse noise drawn from uniform distribution $[-5,5]$ with scale ratio $\rho_s = 0.2$.
4) Fix missing ratio $\rho_m = 0.2$ and mixture noise drawn from uniform distribution $[-5,5]$ with scale ratio $\rho_s = 0.2$, Gaussian noise $\mathcal{N}(0, 0.2)$ with scale ratio $\rho_s = 0.2$, and Gaussian noise $\mathcal{N}(0, 0.01)$ with scale ratio $\rho_s = 0.4$.

We use the following measurements to assess the performance of each method:

$$E1 = \|\mathcal{W} \odot (\mathcal{X}_{\text{no}} - \mathcal{X}_{\text{rec}})\|_{L_1}, \quad E3 = \|\mathcal{X}_{\text{gt}} - \mathcal{X}_{\text{rec}}\|_{L_1}$$

$$E2 = \|\mathcal{W} \odot (\mathcal{X}_{\text{no}} - \mathcal{X}_{\text{rec}})\|_{L_F}, \quad E4 = \|\mathcal{X}_{\text{gt}} - \mathcal{X}_{\text{rec}}\|_{L_F}$$

where $\mathcal{X}_{\text{no}}$, $\mathcal{X}_{\text{rec}}$, and $\mathcal{X}_{\text{gt}}$ denote the noisy, recovered, and ground-truth tensor, respectively. The results obtained using each method are listed in Tables I and II. The first and second best results are highlighted and underlined, respectively. When the data missing rate varies, our methods perform better according to all evaluation indices. Compared with the other methods, which are optimal only when the noise follows a Gaussian distribution or uniform distribution, our methods obtain a relatively better performance under both simple and complex noise conditions.

B. Single RGB Image Reconstruction

In this section, the three-order benchmark RGB image data, i.e., a building facade of size $493\times517\times3$, are used to validate the effectiveness of each method in reconstructing a single RGB image. Two experimental cases are considered here.

In the first case, the data with the original magnitude [0,255] are randomly corrupted by a relatively small-magnitude complex noise: 20% missing entries, 20% uniform distribution over $[-35,35]$, 20% Gaussian distribution $\mathcal{N}(0, 20)$, and 40% Gaussian distribution $\mathcal{N}(0, 10)$. The visual results obtained

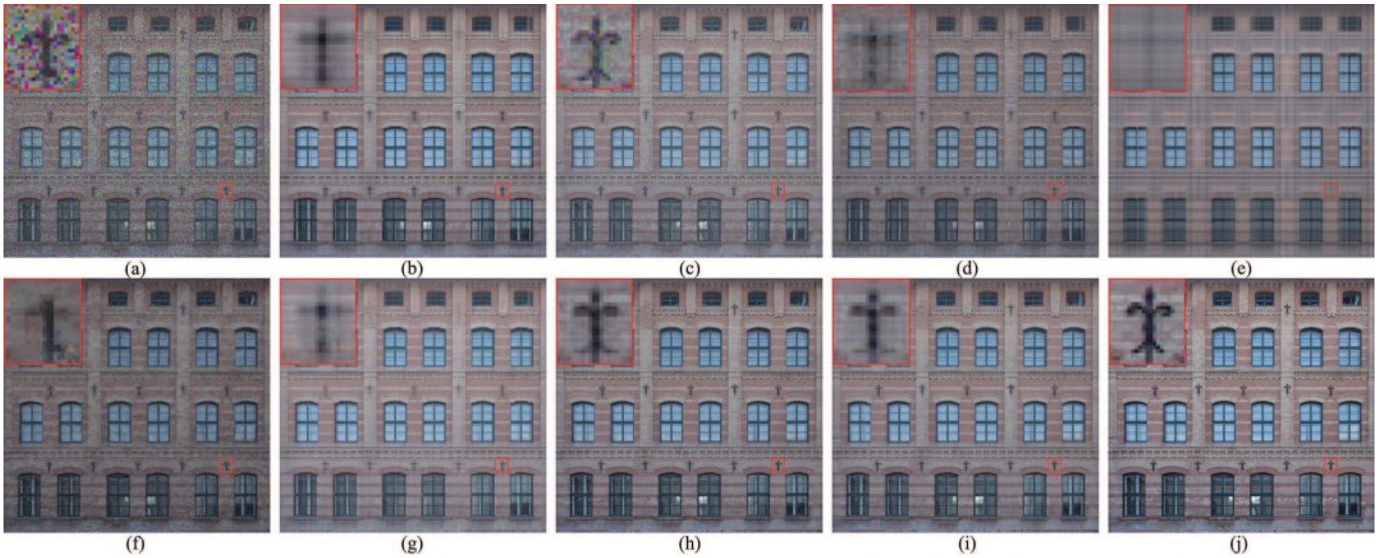


Fig. 2. Building facade with a relatively small-magnitude complex noise. (a) Noisy image. (b) MoG LRMF. (c) HaLRTC. (d) LRTA. (e) PARAFAC. (f) MSI DL. (g) CWM LRTF. (h) MoG GWLRTF-CP. (i) MoG GWLRTF-Tucker. (j) Original image.

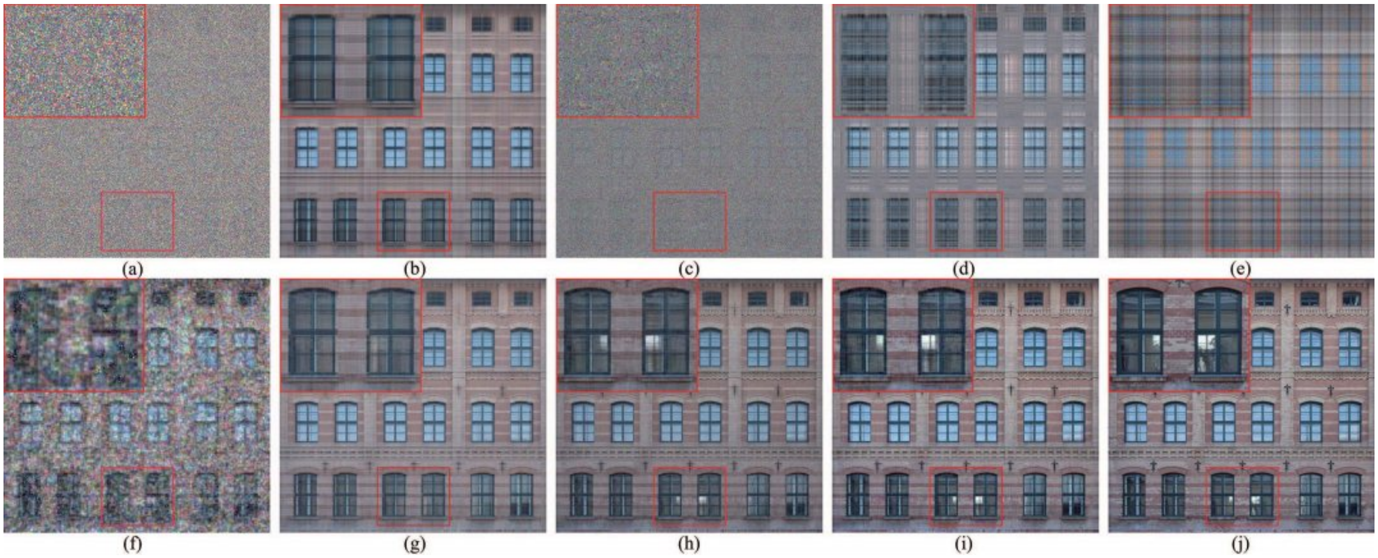


Fig. 3. Building facade with a relatively large-magnitude complex noise. (a) Noisy image. (b) MoG LRMF. (c) HaLRTC. (d) LRTA. (e) PARAFAC. (f) MSI DL. (g) CWM LRTF. (h) MoG GWLRTF-CP. (i) MoG GWLRTF-Tucker. (j) Original image.

using each method are given in Fig. 2; the result of a local region is also enlarged in this demonstration. The results show that our proposed MoG GWLRTF method has a better performance in reconstructing the image details compared with the typical matrix-based and tensor-based methods.

To further quantitatively evaluate the quality of the competing methods, feature similarity (FSIM) [48], peak signal-to-noise ratio (PSNR), and relative standard error (RSE) are employed to measure the image quality. It is known that larger FSIM and PSNR and smaller RSE values indicate better image quality. The quantitative results obtained using each method are given in Table III (the small magnitude case). Compared with the other methods, our proposed methods have better performances in reconstructing the image details. Specifically, MoG GWLRTF-Tucker and MoG GWLRTF-CP have comparable results both qualitatively and quantitatively.

In the second case, we continue to increase the magnitude of the noise to further test the effectiveness in robust image construction. Here, the image is rescaled to [0,1]. A relatively large-magnitude complex noise is added as follows: 20% missing entries, 20% uniform distribution over [-5,5], 20% Gaussian noise $\mathcal{N}(0, 0.2)$, and 40% Gaussian distribution $\mathcal{N}(0, 0.01)$. The qualitative and quantitative results are given in Fig. 3 and Table III (the large magnitude case), respectively. Similarly, the enlarged results of a local region are demonstrated in Fig. 3. From the results, we can see that with the increase in noise magnitude, the MoG GWLRTF-Tucker method performs better than the MoG GWLRTF-CP method in reconstructing image details. Both are still superior to the

TABLE III

FACADE RECONSTRUCTION PERFORMANCE OF COMPETING METHODS WITH MIXTURE NOISE. IN EACH EXPERIMENT, THE FIRST BEST PERFORMANCE IS HIGHLIGHTED IN BOLD AND THE SECOND BEST IS UNDERLINED

Facade		MoG LRMF	HaLRTC	LRTA	PARAFAC	MSI DL	CWM LRTF	MoG GWLRTF-CP	MoG GWLRTF-Tucker
small magnitude	PSNR	24.34	23.43	13.59	13.37	13.53	24.80	25.65	**25.67**
	RSE	0.1169	0.1298	0.4026	0.4129	0.4062	0.1109	0.1005	**0.1003**
	FSIM	0.8954	0.9407	0.8318	0.7402	0.8258	0.9435	**0.9539**	0.9454
large magnitude	PSNR	22.18	14.20	18.51	16.95	16.71	22.82	23.69	**24.46**
	RSE	0.1499	0.3755	0.2287	0.2737	0.2817	0.1393	0.1260	**0.1127**
	FSIM	0.8525	0.6003	0.7667	0.7101	0.7310	0.9117	0.9268	**0.9310**

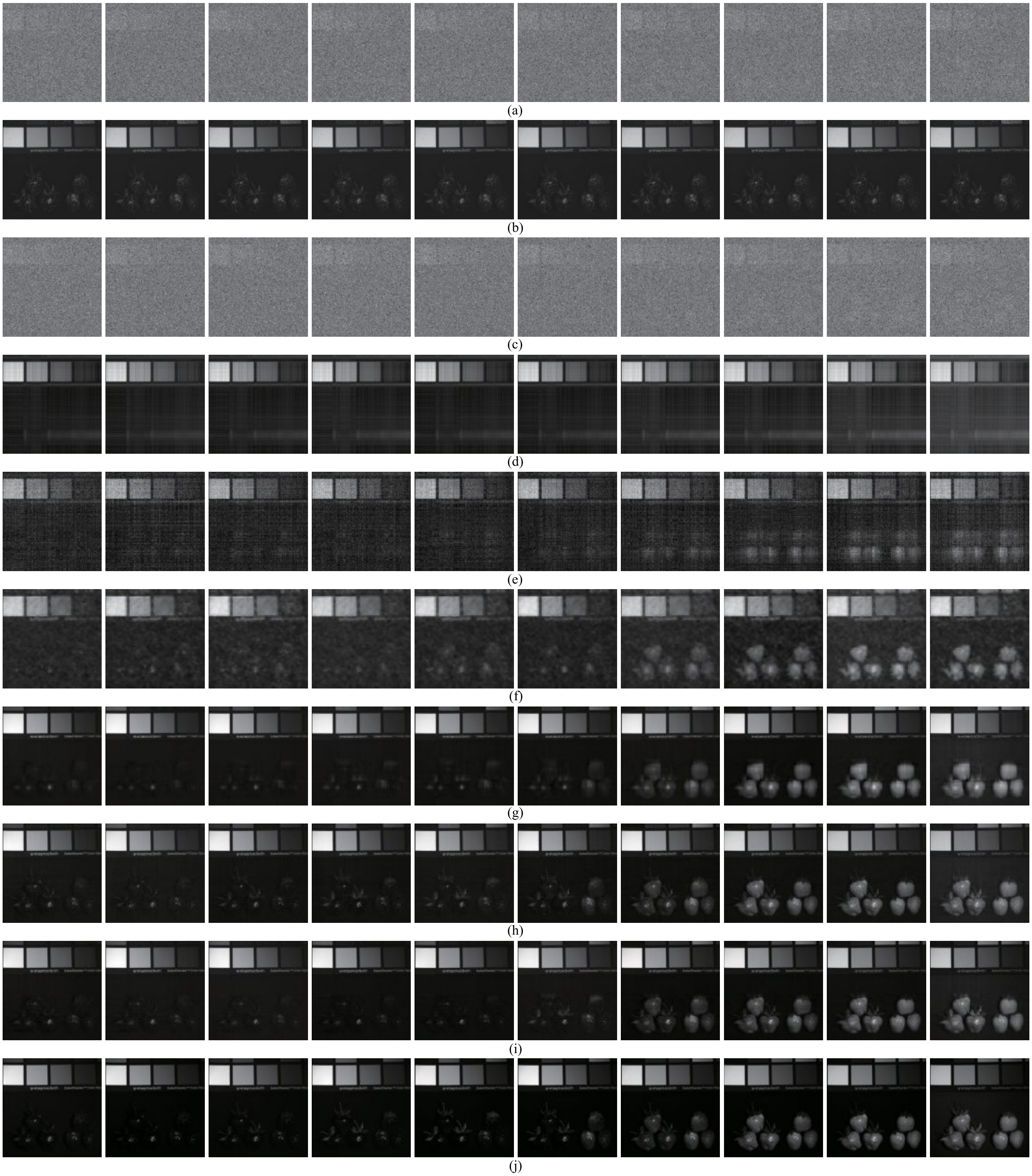
Fig. 4. Sampling images of the first people under nine illuminations.

Fig. 5. Face modeling results by different methods. (a) Original face images. (b) MoG LRMF. (c) HaLRTC. (d) LRTA. (e) PARAFAC. (f) MSI DL. (g) CWM LRTF. (h) MoG GWLRTF-CP. (i) MoG GWLRTF-Tucker.

other competing methods, which lose the image structure information.

C. Face Modeling

In this section, we evaluate the face modeling performances of the proposed methods for different objects under different illuminations. This is different from the traditional methods for dealing with the face modeling problem, which always focus only on one kind of object under different illuminations. The data set used here is the Extended Yale B database [49], which contains 45 faces of 5 objects and 9 illuminations with a size of 192×168. The original tensor is thus generated with a size of $192\times168\times9\times5$. Considering that some tensor methods are designed to deal with only three-order tensors and that the matrix methods were originally designed to solve matrix data, in this case, the original four-order tensor is vectorized into a three-order tensor with a size of $192 \times 168 \times 45$ and matrix data with a size of $32\,256 \times 45$ before processing.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

(j)

Fig. 6. Randomly selected bands of strawberries recovered by competing methods. (a) Noisy bands. (b) MoG LRMF. (c) HaLRTC. (d) LRTA. (e) PARAFAC. (f) MSI DL. (g) CWM LRTF. (h) MoG GWLRTF-CP. (i) MoG GWLRTF-Tucker. (j) Original bands.

The sampling images from this data subset are plotted in Fig. 4. Typical face images and the corresponding modeling results obtained using all the competing methods are demonstrated in Fig. 5. The Tucker-based methods (MoG GWLRTF-Tucker and CWM LRTF) outperform MoG GWLRTF-CP and other competing methods in reconstructing the faces. Specifically, MoG GWLRTF-Tucker has a better performance than that of CWM LRTF when performing MoG

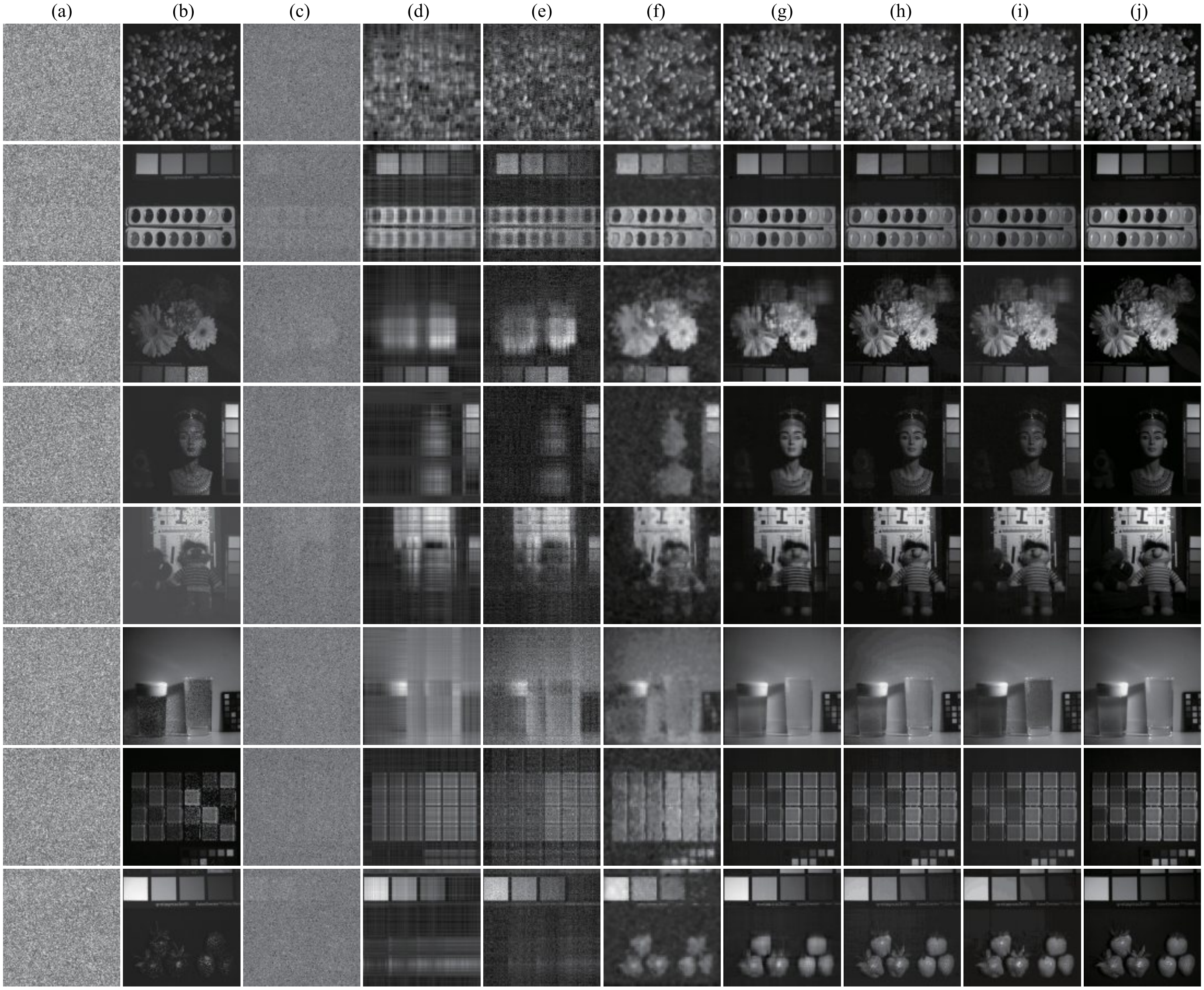


Fig. 7. 31st band of multispectral images. (a) Noisy band. (b) MoG LRMF. (c) HaLRTC. (d) LRTA. (e) PARAFAC. (f) MSI DL. (g) CWM LRTF. (h) MoG GWLRTF-CP. (i) MoG GWLRTF-Tucker. (j) Original band.

noise modeling to remove the cast shadows and saturations in the nose area on the face.

D. Multispectral Image Recovery

The Columbia Multispectral Image database [50][1] contains 32 scenes of a wide range of real-world objects. It contains 31 spectral bands, and the images are of size 512 × 512. We use 8 of them (Jelly beans, Paints, Flowers, Egyptian statue, Chart and Stuffed toy, Beers, Glass tiles, and Strawberries) to test the effectiveness of our methods. The images used are resized by half and rescaled to [0,1]. We add large-magnitude complex noise to these images.

For a better comparison of each method in recovering each band, we randomly choose ten bands of the strawberry images for this demonstration. From Fig. 6, we can see that our methods have a relatively better performance in obtaining the image structure and protecting the spectral information of each band compared with the other methods. Note that to remove the noise from the images, the ranks used in MoG LRMF are all set to 1. As we can see, the recovered bands obtained via MoG LRMF have fine structure information, but their spectral information is lost.

Furthermore, we provide the recovery results for the 31st band of the multispectral images in Fig. 7. Quantitative results are given in Table IV according to the PSNR, RSE, and FSIM quality measure indices. The proposed MoG GWLRTF method outperforms the other competing methods in both structure information recovery and spectral information protection. Specifically, the MoG GWLRTF-CP performs better than the MoG GWLRTF-Tucker in this application.

E. Real Hyperspectral Image Restoration

In this section, we further employ the proposed methods for the real hyperspectral image restoration application to test their effectiveness compared with that of other methods.

---

[1] http://www1.cs.columbia.edu/CAVE/databases/multispectral.

TABLE IV

MULTISPECTRAL IMAGE RESTORATION RESULTS OBTAINED BY COMPETING METHODS UNDER COMPLEX NOISE. THE BEST PERFORMANCE IS HIGHLIGHTED IN BOLD AND THE SECOND IS UNDERLINED

		MoG LRMF	HaLRTC	LRTA	PARAFAC	MSI DL	CWM LRTF	MoG GWLRTF-CP	MoG GWLRTF-Tucker
Jelly beans	PSNR	19.26	8.444	16.51	15.84	18.47	19.94	**22.17**	22.45
	RSE	0.5169	1.773	0.7003	0.7565	0.5588	0.4720	**0.3652**	0.3535
	FSIM	0.8101	0.5312	0.7025	0.6487	0.8213	0.8506	**0.8864**	0.9004
Paints	PSNR	18.90	8.725	18.76	16.26	19.20	21.75	**27.29**	24.48
	RSE	0.4022	1.336	0.4209	0.5607	0.3998	0.2983	**0.1576**	0.2177
	FSIM	0.8543	0.4750	0.7827	0.6367	0.8182	0.9165	**0.9514**	0.9427
Flowers	PSNR	16.45	7.480	18.40	16.54	19.15	22.72	**26.17**	24.24
	RSE	0.9073	2.522	0.7173	0.8888	0.6583	0.4364	**0.2933**	0.3511
	FSIM	0.8018	0.4172	0.8073	0.5271	0.8220	0.9126	**0.9153**	0.9139
Egyptian statue	PSNR	15.32	7.109	18.59	17.01	19.43	23.20	**25.22**	23.58
	RSE	1.563	3.847	1.026	1.231	0.9308	0.6032	**0.4779**	0.5271
	FSIM	0.7834	0.3696	0.8245	0.4243	0.8142	0.9187	**0.9439**	0.9233
Chart	PSNR	12.05	7.911	18.29	16.23	18.95	21.46	**23.95**	23.33
	RSE	0.8633	1.388	0.4201	0.5327	0.3895	0.2919	**0.2191**	0.2353
	FSIM	0.8069	0.4534	0.7729	0.5366	0.8092	0.8901	**0.9221**	0.9402
Beers	PSNR	22.22	11.56	19.92	17.14	21.33	21.92	**25.07**	20.98
	RSE	0.2389	0.8150	0.3114	0.4287	0.2646	0.2474	**0.1722**	0.2756
	FSIM	0.7659	0.4155	0.7535	0.4423	0.8214	**0.9430**	0.9200	0.8748
Glass	PSNR	19.02	8.380	18.74	16.49	18.89	21.66	**26.66**	20.79
	RSE	0.6314	2.025	0.6140	0.7961	0.6038	0.4389	**0.2467**	0.4541
	FSIM	0.6990	0.4690	0.7726	0.5738	0.7287	0.9075	**0.9475**	0.7946
Strawberries	PSNR	17.06	7.762	19.22	16.78	19.54	18.99	**24.80**	23.41
	RSE	0.7630	2.149	0.5745	0.7607	0.5534	0.5900	**0.3021**	0.3652
	FSIM	0.7906	0.3997	0.8059	0.4932	0.8128	0.9229	**0.9283**	0.9137

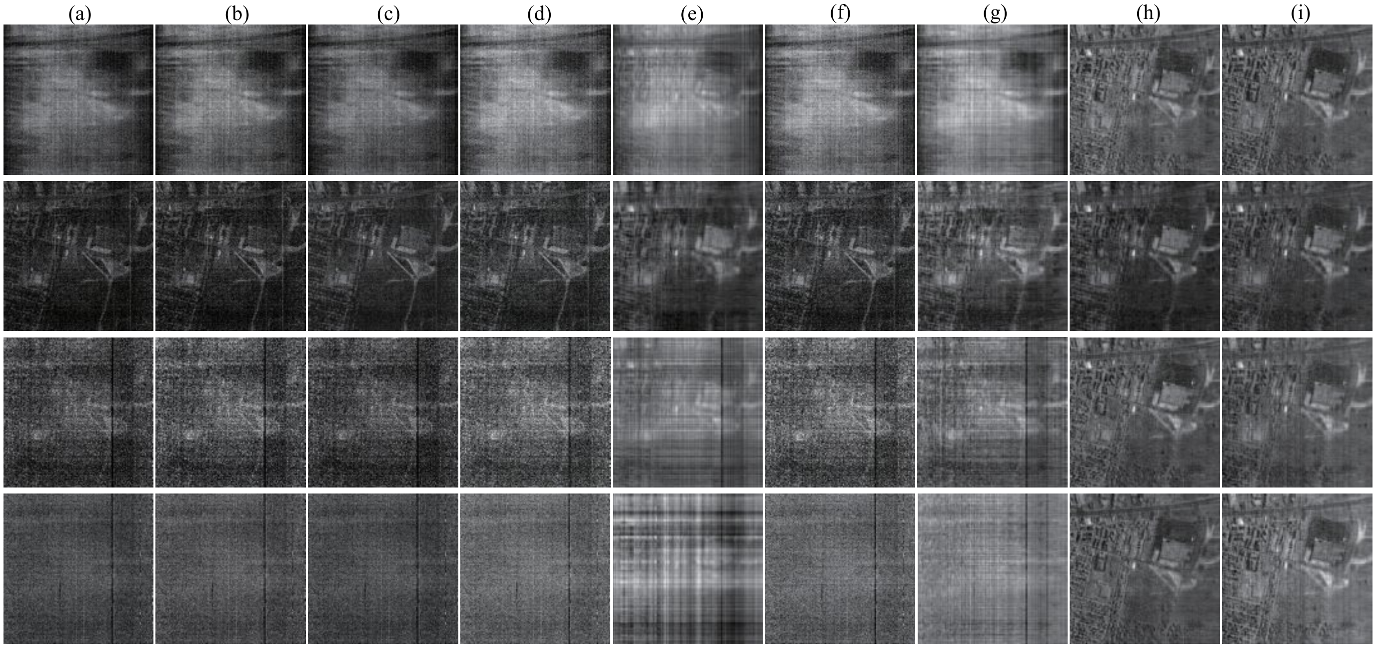


Fig. 8. Real hyperspectral image restoration. (a) Original polluted bands. (b) MoG LRMF. (c) HaLRTC. (d) LRTA. (e) PARAFAC. (f) MSI DL. (g) CWM LRTF. (h) MoG GWLRTF-CP. (i) MoG GWLRTF-Tucker.

The data used here are the HYDICE urban image,[2] which contains 210 bands with an image size of 307 × 307. Some of these bands are seriously corrupted by the atmosphere and water absorption. The existence of these contaminated bands poses steep challenges regarding the effectiveness of the current methods. These bands are sometimes discarded before applying the current methods for further vision tasks [36]. In contrast, we directly apply our methods to all these bands and try to restore all of them.

Four typical polluted bands are demonstrated in the first column of Fig. 8. The corresponding restoration results obtained using the MoG GWLRTF-CP, MoG GWLRTF-Tucker, and other competing methods are given in the following columns.

Similar to the case of multispectral image recovery, the MoG GWLRTF-CP is continued to obtain relatively better results compared with those of the MoG GWLRTF and the

---

[2] http://www.tec.army.mil/hypercube.

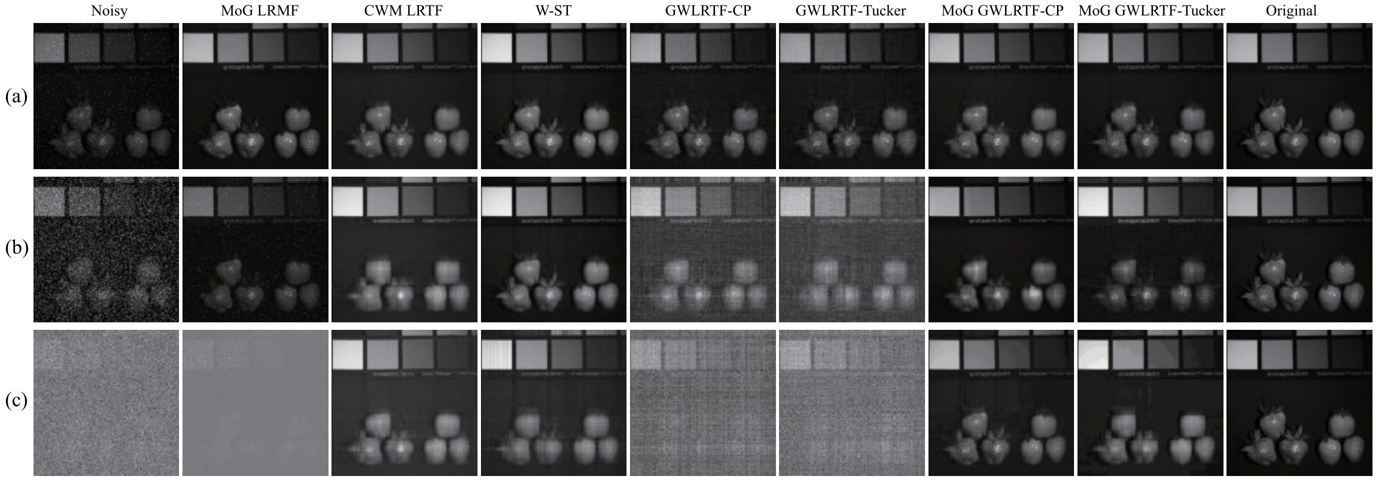


Fig. 9. Heavy-tailed noise removal by competing methods. (a) Small-scale Cauchy noise. (b) Large-scale Cauchy noise. (c) Complex noise.

other competing methods in this real hyperspectral image restoration experiment.

F. Heavy-Tailed Noise Removal

In radar and sonar systems, the observed data are susceptible to the atmosphere and water absorption [51]. Different from the Gaussian density curve, which has very light tails, the distribution curve of the noise in these observed data often has heavy tails, though it becomes smoother near zero compared with the Laplacian curve [52]. Some specific algorithms have been designed to deal with this kind of noise, such as the tensor approaches with regularized redescending M-estimators [53], Cauchy noise removal via the nonconvex alternating direction method of multiplier [54], and the variational approach for restoring images with Cauchy noise [55].

In this section, we pay attention to the noise that arises in these applications and further test the robustness of our proposed methods on this kind of data. Instead of continuing to use the original competing methods, we compare our proposed methods with the methods that correspond to heavier tails:

1) matrix-based MoG denoising method (MoG LRMF);
2) tensor-based L_1-norm denoising method (CWM LRTF);
3) the typical Cauchy denoising method [Welsch loss soft thresholding (W-ST)] [53];
4) our CP-based LRTF model (GWLRTF-CP);
5) our Tucker-based LRTF model (GWLRTF-Tucker).

Three noise situations are considered.

1) *Small-Scale Cauchy Noise:* Random outliers drawn from the chi-square distribution with a small scale of 0.1.
2) *Large-Scale Cauchy Noise:* Random outliers drawn from the chi-square distribution with a large scale of 4.
3) *Complex Noise:* 20% missing entries, nonmissing pixels with a 20% small-scale chi-square distribution, 20% uniform distribution over $[-5, 5]$, 20% Gaussian distribution $\mathcal{N}(0, 0.2)$, and 20% Gaussian distribution $\mathcal{N}(0, 0.01)$.

A visual comparison based on multispectral strawberry images is given in Fig. 9. The specific Cauchy noise removal method W-ST obtains slightly better results than those of our methods (MoG GWLRTF-CP and MoG GWLRTF-Tucker) in removing the small-scale Cauchy noise. In addition, CWM LRTF has results comparable to those of ours, whereas GWLRTF-CP and GWLRTF-Tucker perform worse. MoG LRMF can preserve the structure information very well but loses the spectral information, as shown in Fig. 6(b). In the large-scale Cauchy noise case, W-ST demonstrates a much better recovery performance than that of the others. The denoising performances of the GWLRTF-CP and GWLRTF-Tucker decrease sharply compared with those of the MoG GWLRTF-CP and MoG GWLRTF-Tucker. However, when dealing with complex noise, our methods (MoG GWLRTF-CP and MoG GWLRTF-Tucker) outperform W-ST and CWM LRTF. In addition, the GWLRTF-CP and GWLRTF-Tucker begin to lose their recovery effectiveness. Specifically, the MoG GWLRTF-CP still performs better than the MoG GWLRTF, as demonstrated in the spectral image recovery Sections D and E.

VI. CONCLUSION AND DISCUSSION

As an extension of the shorter version of our work [37], this paper proposes a generalized LRTF model integrated with MoG noise modeling, which is shown to be robust to complex noise and to have wider applicability. To better estimate the low-rank subspaces from high-dimensional data, we employ two typical tensor factorizations, i.e., CP factorization and Tucker factorization, in the low-rank tensor recovery procedure. The MoG parameters are calculated using the maximum log-likelihood function, and the low-rank subspace parameters are estimated by solving the GWLRTF model. Our extensive experiments demonstrate the effectiveness of MoG GWLRTF compared with that of other competing methods.

Specifically, the Tucker-based MoG GWLRTF is achieved better results in single RGB image reconstruction and face modeling than those of the CP-based MoG GWLRTF. As a multilinear version of PCA, the Tucker-based method is good at leveraging its mode matrices. Therefore, it has an advantage in applications in which the observed data have completely different physical meanings in different modes.

The CP-based MoG GWLRTF is performed relatively well in all spectral image recovery applications. In these spectral applications, the observed data along each mode are similar enough, especially along the spectral mode. As a higher order generalization of matrix SVD, the CP-based method does well in low-rank subspace learning and data compression, as expected. The different properties between these two factorization versions indicate their complementary advantages in various applications.

Inspired by this observation, we will further investigate the difference between MoG GWLRTF-CP and MoG GWLRTF-Tucker by applying them to additional real applications and by integrating them with the Markov random field to further evaluate their video processing performances.

REFERENCES

[1] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 2080–2088.

[2] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *J. Opt. Soc. Amer. A, Opt. Image Sci. Vis.*, vol. 4, no. 3, pp. 519–524, 1987.

[3] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 1991, pp. 586–591.

[4] P. N. Belhumeur, J. P. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[5] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," *Int. J. Comput. Vis.*, vol. 9, no. 2, pp. 137–154, Nov. 1992.

[6] H. Murase and S. K. Nayar, "Learning and recognition of 3D objects from appearance," in *Proc. IEEE Workshop Qualitative Vis.*, Jun. 1993, pp. 39–50.

[7] R. Vidal, R. Tron, and R. Hartley, "Multiframe motion segmentation with missing data using powerfactorization and GPCA," *Int. J. Comput. Vis.*, vol. 79, no. 1, pp. 85–105, 2008.

[8] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, p. 11, 2011.

[9] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, Jan. 2013.

[10] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an N-way generalization of 'Eckart-Young' decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[11] R. A. Harshman, "Foundations of the parafac procedure: Models and conditions for an 'explanatory' multimodal factor analysis," *UCLA Working Papers Phonetics*, vol. 16, no. 1, pp. 1–84, 1970.

[12] H. A. L. Kiers, "Towards a standardized notation and terminology in multiway analysis," *J. Chemometrics*, vol. 14, no. 3, pp. 105–122, 2000.

[13] Q. Zhao, L. Zhang, and A. Cichocki, "Bayesian CP factorization of incomplete tensors with automatic rank determination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1751–1763, Sep. 2015.

[14] P. Rai, Y. Wang, S. Guo, G. Chen, D. Dunson, and L. Carin, "Scalable Bayesian low-rank decomposition of incomplete multiway tensors," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1800–1808.

[15] L. Xiong, X. Chen, T.-K. Huang, J. G. Schneider, and J. G. Carbonell, "Temporal collaborative filtering with Bayesian probabilistic tensor factorization," in *Proc. SIAM Int. Conf. Data Mining*, 2010, pp. 211–222.

[16] B. Savas, "Analyses and tests of handwritten digit recognition algorithms," M.S. thesis, Dept. Math., Linköping Univ., Linköping, Sweden, Jan. 2003.

[17] J. B. Kruskal, "Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear Algebra Appl.*, vol. 18, no. 2, pp. 95–138, 1975.

[18] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *Stud. Appl. Math.*, vol. 6, nos. 1–4, pp. 164–189, 1927.

[19] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[20] L. De Lathauwer and J. Vandewalle, "DDimensionality reduction in higher-order signal processing and rank-(R1,R2,…,RN) reduction in multilinear algebra," *Linear Algebra Appl.*, vol. 391, pp. 31–55, Nov. 2004.

[21] D. Muti and S. Bourennane, "Multidimensional filtering based on a tensor approach," *Signal Process.*, vol. 85, no. 12, pp. 2338–2353, 2005.

[22] B. Savas and L. Eldén, "Handwritten digit classification using higher order singular value decomposition," *Pattern Recognit.*, vol. 40, no. 3, pp. 993–1003, 2007.

[23] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear analysis of image ensembles: Tensorfaces," in *Proc. Eur. Conf. Comput. Vis.*, 2002, pp. 447–460.

[24] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear subspace analysis of image ensembles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2003, pp. II-93–II-99.

[25] D. Vlasic, M. Brand, H. Pfister, and J. Popović, "Face transfer with multilinear models," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 426–433, 2005.

[26] H. Wang and N. Ahuja, "Facial expression decomposition," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 958–965.

[27] K. J. Friston, S. Williams, R. Howard, R. S. Frackowiak, and R. Turner, "Movement-related effects in fMRI time-series," *Magn. Reson. Med.*, vol. 35, no. 3, pp. 346–355, 1996.

[28] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1459–1472, Nov. 2004.

[29] H. Huang and C. Ding, "Robust tensor factorization using R1 norm," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.

[30] E. C. Chi and T. G. Kolda. (Oct. 2010). "Making tensor factorizations robust to non-Gaussian noise." [Online]. Available: https://arxiv.org/abs/1010.3043

[31] Z. Zhou, X. Li, J. Wright, E. Candès, and Y. Ma, "Stable principal component pursuit," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2010, pp. 1518–1522.

[32] Q. Gu, H. Gui, and J. Han, "Robust tensor decomposition with gross corruption," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1422–1430.

[33] X. Chen, Z. Han, Y. Wang, Y. Tang, and H. Yu, "Nonconvex plus quadratic penalized low-rank and sparse decomposition for noisy image alignment," *Sci. China Inf. Sci.*, vol. 59, p. 052107, May 2016.

[34] D. Meng and F. De La Torre, "Robust matrix factorization with unknown noise," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 1337–1344.

[35] Q. Zhao, D. Meng, Z. Xu, W. Zuo, and L. Zhang, "Robust principal component analysis with complex noise," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1–9.

[36] Q. Zhao *et al.*, "A novel sparsity measure for tensor recovery," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 271–279.

[37] X. Chen, Z. Han, Y. Wang, Q. Zhao, D. Meng, and Y. Tang, "Robust tensor factorization with unknown noise," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5213–5221.

[38] P. Scheunders and S. D. Backer, "Wavelet denoising of multicomponent images using Gaussian scale mixture models and a noise-free image as priors," *IEEE Trans. Image Process.*, vol. 16, no. 7, pp. 1865–1872, Jul. 2007.

[39] M. J. Wainwright and E. P. Simoncelli, "Scale mixtures of Gaussians and the statistics of natural images," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 855–861.

[40] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. B, Methodol.*, vol. 39, no. 1, pp. 1–38, 1977.

[41] D. Meng, B. Zhang, Z. Xu, L. Zhang, and C. Gao, "Robust low-rank tensor factorization by cyclic weighted median," *Sci. China Inf. Sci.*, vol. 58, no. 5, pp. 1–11, 2015.

[42] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, "Pathwise coordinate optimization," *Ann. Appl. Stat.*, vol. 1, no. 2, pp. 302–332, 2007.

[43] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *J. Statist. Softw.*, vol. 33, no. 1, pp. 1–22, 2010.

[44] B. W. Bader and T. G. Kolda. (Feb. 2015). *MATLAB Tensor Toolbox Version 2.6*. [Online]. Available: http://www.sandia.gov/~tgkolda/TensorToolbox/

[45] X. Liu, S. Bourennane, and C. Fossati, "Denoising of hyperspectral images using the PARAFAC model and statistical performance analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 10, pp. 3717–3724, Oct. 2012.

[46] N. Renard, S. Bourennane, and J. Blanc-Talon, "Denoising and dimensionality reduction using multilinear tools for hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 2, pp. 138–142, Apr. 2008.

[47] Y. Peng, D. Meng, Z. Xu, C. Gao, Y. Yang, and B. Zhang, "Decomposable nonlocal tensor dictionary learning for multispectral image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 2949–2956.

[48] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, Aug. 2011.

[49] A. S. Georghiades, P. N. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, Jun. 2001.

[50] F. Yasuma, T. Mitsunaga, D. Iso, and S. K. 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.

[51] J. Fernandez, J. L. Speyer, and M. Idan, "A stochastic controller for vector linear systems with additive cauchy noise," in *Proc. IEEE 52nd Annu. Conf. Decision Control (CDC)*, Dec. 2013, pp. 1872–1879.

[52] Y. Wu, H. Tan, Y. Li, F. Li, and H. He, "Robust tensor decomposition based on Cauchy distribution and its applications," *Neurocomputing*, vol. 223, pp. 107–117, Feb. 2017.

[53] Y. Yang, Y. Feng, and J. A. K. Suykens, "Robust low-rank tensor recovery with regularized redescending M-estimator," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 9, pp. 1933–1946, Sep. 2016.

[54] J.-J. Mei, Y. Dong, T.-Z. Huang, and W. Yin, "Cauchy noise removal by nonconvex ADMM with convergence guarantees," *J. Sci. Comput.*, pp. 1–24, 2017.

[55] F. Sciacchitano, Y. Dong, and T. Zeng, "Variational approach for restoring blurred images with cauchy noise," *SIAM J. Imag. Sci.*, vol. 8, no. 3, pp. 1894–1922, 2015.

Xi'ai Chen received the B.Sc. degree from the Department of Automation, Qufu Normal University, Jining, China, in 2012. She is currently pursuing the Ph.D. degree with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, University of Chinese Academy of Sciences, Beijing, China.

Her current research interests include low-rank matrix/tensor recovery and image/video modeling.

Zhi Han (M'16) received the B.Sc., M.Sc., and Ph.D. degrees in applied mathematics from Xi'an Jiaotong University, Xi'an, China, in 2005, 2007, and 2012, respectively, and the joint Ph.D. degree in statistics from the University of California at Los Angeles, Los Angeles, CA, USA, in 2011.

He is currently an Associate Professor with the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Beijing, China. His current research interests include image/video modeling, low-rank matrix recovery, and illumination processing.

Yao Wang received the Ph.D. degree in applied mathematics from Xi'an Jiaotong University, Xi'an, China, in 2014.

He was a Visiting Student with the Georgia Institute of Technology, Atlanta, GA, USA, from 2010 to 2011. He is currently an Associate Professor with the Department of Statistics, Xi'an Jiaotong University. His current research interests include statistical signal processing, high-dimensional statistical inference, large-scale video analysis, and computational biology.

Qian Zhao received the B.Sc. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 2009 and 2015, respectively.

He was a Visiting Scholar with Carnegie Mellon University, Pittsburgh, PA, USA, from 2013 to 2014. He is currently a Lecturer with the School of Mathematics and Statistics, Xi'an Jiaotong University. His current research interests include low-matrix/tensor analysis, Bayesian modeling, and self-paced learning.

Deyu Meng (M'14) received the B.Sc., M.Sc., and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 2001, 2004, and 2008, respectively.

He is currently an Associate Professor with the School of Mathematics and Statistics, Institute for Information and System Sciences, Xi'an Jiaotong University. From 2012 to 2014, he took his two-year sabbatical leave at Carnegie Mellon University, Pittsburgh, PA, USA. His current research interests include self-paced learning, noise modeling, and tensor sparsity.

Lin Lin received the B.Sc. and M.Sc. degrees from the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, China, in 2013 and 2016, respectively.

His current research interests include low-rank modeling, compressed sensing, tensor sparsity, and hyperspectral image processing.

Yandong Tang (M'11) received the B.Sc. and M.Sc. degrees in mathematics from Shandong University, Jinan, China, in 1984 and 1987, respectively, and the Ph.D. degree in applied mathematics from the University of Bremen, Bremen, Germany, in 2002.

He is currently a Professor with the Shenyang Institute of Automation, Chinese Academy of Sciences, Beijing, China. His current research interests include numerical computation, image processing, and computer vision.