# Simultaneous Tensor Decomposition and Completion Using Factor Priors

Yi-Lei Chen, *Student Member, IEEE*, Chiou-Ting Hsu, *Senior Member, IEEE*, and Hong-Yuan Mark Liao, *Fellow, IEEE*

**Abstract**—The success of research on matrix completion is evident in a variety of real-world applications. Tensor completion, which is a high-order extension of matrix completion, has also generated a great deal of research interest in recent years. Given a tensor with incomplete entries, existing methods use either factorization or completion schemes to recover the missing parts. However, as the number of missing entries increases, factorization schemes may overfit the model because of incorrectly predefined ranks, while completion schemes may fail to interpret the model factors. In this paper, we introduce a novel concept: *complete the missing entries and simultaneously capture the underlying model structure*. To this end, we propose a method called simultaneous tensor decomposition and completion (STDC) that combines a rank minimization technique with Tucker model decomposition. Moreover, as the model structure is implicitly included in the Tucker model, we use *factor priors*, which are usually known a priori in real-world tensor objects, to characterize the underlying joint-manifold drawn from the model factors. By exploiting this auxiliary information, our method leverages two classic schemes and accurately estimates the model factors and missing entries. We conducted experiments to empirically verify the convergence of our algorithm on synthetic data and evaluate its effectiveness on various kinds of real-world data. The results demonstrate the efficacy of the proposed method and its potential usage in tensor-based applications. It also outperforms state-of-the-art methods on multilinear model analysis and visual data completion tasks.

**Index Terms**—Tensor completion, Tucker decomposition, factor priors, multilinear model analysis

✦

## 1 INTRODUCTION

THE increasing popularity of low-rank matrix approximation in recent years demonstrates the method's significance as a theoretic foundation for real-world problems, such as inpainting [1], [2], denoising [3], image batch alignment [4], key-point/saliency detection [5], [6], affinity/subspace learning [7], [8], and moving object analysis [9], [10], [11]. To infer a problem's statistics from limited information (e.g., noisy or incomplete data), the above methods estimate missing or noise-free values via matrix factorization or matrix completion. Matrix factorization techniques (e.g., singular value decomposition (SVD) [12] and nonnegative matrix factorization (NMF) [13]), under a fixed-rank representation, use a reconstruction step to maintain the principal variation and suppress the additive noise. In contrast, matrix completion techniques [14], [15], [16] exploit nuclear norm (or matrix trace norm) minimization to recover incomplete data effectively. It has been shown that the nuclear norm yields the tightest convex envelope of a matrix rank [17]. The major difference between the two techniques is the way they make decisions about the matrix rank. Completion-based methods automate the rank estimation step, while factorization-based methods assume a given rank, but tend to over/under-estimate the truth. A number of studies posit that matrix factorization techniques are superior for data analysis [18] and statistical modeling [19] because the factorized matrix structure is a natural fit for the problems.

As a tensor is a high-order extension of a matrix, low-rank tensor approximation has also generated increased interest in factorization techniques [20], [21], [22] and completion techniques [23], [24], [25], [26], [27], [28]. Like matrix approximation, low-rank tensor approximation can be solved by both techniques, but a tensor's rank is not usually as well defined as a matrix's rank. Theoretically, a tensor's rank is the minimum number of components $r$ required for rank-1 decomposition (also known as canonical polyadic (CP) decomposition). However, as it is difficult to estimate the minimum $r$ in practice, the mode-$k$ rank $(r_1, \ldots, r_K)$ estimated by Tucker decomposition [28] could be considered instead. Despite the difficulty in defining a tensor's rank, a great deal of research effort has been devoted to understanding low-rank tensor approximation because a tensor object generally exhibits the structure of real-world data.

In this paper, we focus on *tensor completion*, which is related to the missing data problem in many real-world applications. When only a fraction of the entries in a tensor can be observed, we try to approximate the unknown or missing entries and thereby improve the performance of the underlying applications. Two tensor factorization methods were recently proposed in [21], [22] to estimate the model

● *Y.-L. Chen and C.-T. Hsu are with the Multimedia Processing Laboratory, Department of Computer Science, National Tsing Hua University, 101, Sec. 2, Kuang Fu Rd, Hsinchu 300, Taiwan.*
*E-mail: fallcolor@gmail.com, cthsu@cs.nthu.edu.tw.*
● *H.-Y.M. Liao is with the Institute of Information Science, Academia Sinica, Taipei 115, Taiwan, and also with the Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan.*
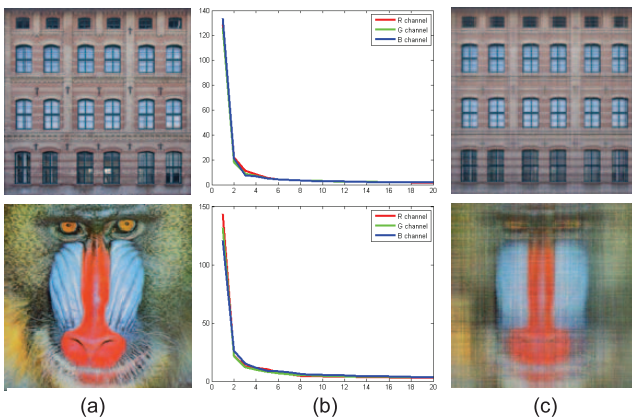*E-mail: liao@iis.sinica.edu.tw.*

Fig. 1. Two sample images "Façade" and "Baboon": (a) the original images, (b) the first 20 singular values, and (c) the completed results reported by [27] (the missing data rate = 90 percent).

factors in cases of incomplete data. Based on the CP model, the authors of [21] formulated the reconstruction of given entries as a weighted least-squares problem. Meanwhile, in [22], the authors proposed an expectation-maximization (EM)-like algorithm that uses high-order SVD [29] to characterize the multilinear subspaces as components of the Tucker model. Both methods were applied to real-world data analysis, for example, electroencephalogram (EEG), computer network traffic, and face image data, and they successfully approximated the model factors in terms of the factor matching score or classification rate. Even so, accurate estimates of missing entries are not guaranteed. Because the methods need to predefine the tensor rank, their models tend to overfit the given entries when only sparse observations are available. In contrast, completion-based methods do not make any assumption about the model and rely solely on rank minimization. Liu et al. [23] defined a tensor trace norm as the convex relaxation of a tensor's rank. They cast tensor completion as a convex optimization problem and reported a series of impressive results on visual data. Their success has motivated a number of methods based on the tensor trace norm [24], [25], [26]. In a subsequent work, Liu et al. [27] proposed two state-of-the-art algorithms to improve the performance of their approach. To sum up, completion-based methods, which benefit from the regularization of the tensor trace norm, facilitate reliable estimation of low-rank tensor objects. However, there is no theoretical guarantee that the tensor trace norm is the tightest convex envelope of a tensor rank. In addition, the above methods, which exploit the SVD shrinkage strategy (see Section 2.2), only work well if the tensor object is exactly low-rank. The example in Fig. 1 illustrates this point. The singular values in Fig. 1b show that both images are well represented by ranks lower than 20; however, the completion method [27] yields a poorer visual result on the Baboon image compared to the result of the Façade image. It seems that the tensor trace norm can characterize low-rank tensor objects, but it fails on general tensor objects. Therefore, further research is needed to solve general cases where the tensors do not have an explicit structure.

A number of works (e.g., [30], [31]) have raised similar concerns about probabilistic matrix factorization. Side information in different applications has been considered

as a latent variable in the Gaussian process [30] or as an adaptive prior for model parameters [31]. Furthermore, the regularized matrix factorization approach [18] incorporates a graph Laplacian regularizer to capture the relations between the data in a matrix's rows or columns. Motivated by [18], Narita et al. [32] proposed using the data relationship in CP and Tucker models for tensor factorization. They introduced two regularization terms to explore such auxiliary information and improved the estimates significantly. The approaches in [18] and [32] are based on factorization techniques, which make an assumption about a given rank. However, the assumption is impractical and it raises another challenge, namely, how to determine the unknown tensor's rank.

The above discussion shows that the main limitation of existing methods is that their models are either overstrict or oversimplified. To overcome this limitation, we introduce a novel concept: *complete the missing entries and simultaneously capture the underlying model structure*. Note that our objective is fundamentally different from that of factorization-based methods, which focus on the underlying factor estimates and then exploit the model factors to predict missing entries. By contrast, to achieve the best recovery rate, we argue that, *to obtain an accurate estimate, the missing data and latent factors should be considered simultaneously*. To deal with a tensor object that has incomplete entries, we propose a method called simultaneous tensor decomposition and completion (STDC). Instead of using predefined ranks, we formulate STDC as a constrained optimization problem that exploits rank minimization techniques and decomposition with the Tucker model. As the model structure is implicitly included in the Tucker model, we use the factor priors presented in our previous work [33] to characterize the underlying joint-manifold drawn from the model factors. Because the factor priors in real-world tensor objects are usually known, our method successfully incorporates this auxiliary information to link the factorization schemes and completion schemes. We discuss this aspect further in Section 3.1. Our experiment results demonstrate that STDC achieves significant improvements in tensor completion and outperforms existing works in a variety of tensor-based applications. Readers can find all the results and source codes in the supplementary material, which can be found in the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2013.164, and also on our project website.[1]

The remainder of this paper is organized as follows: In Section 2, we discuss some well-established techniques that form the basic components of our algorithm. In Section 3, we describe the STDC method in detail and also consider certain implementation issues and the method's advantages over existing works. Section 4 details the experiment results obtained on synthetic data and real-world data. Section 5 contains some concluding remarks.

## 2  PRELIMINARIES

### 2.1  Notations and Tensor Basics

First, we define the notations used in the following sections. Lower case letters $(x, y, \ldots)$ denote scalars, bold lowercase

---

1. http://mp.cs.nthu.edu.tw/project_STDC.

letters $(\mathbf{x}, \mathbf{y}, \ldots)$ denote vectors, bold uppercase letters $(\mathbf{X}, \mathbf{Y}, \ldots)$ represent matrices, and calligraphic uppercase letters $(\mathcal{X}, \mathcal{Y}, \ldots)$ represent high-order tensors. An $n$th-order tensor is represented by $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$ and its elements are denoted by $\mathcal{X}_{i_1, i_2, \ldots, i_n} \in \mathbb{R}$ $(1 \le i_k \le I_k)$. The Frobenius norm of $\mathcal{X}$ is defined by $\|\mathcal{X}\|_F^2 = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_n} \mathcal{X}_{i_1, \ldots, i_n}^2$.

In tensor operations, the mode-$k$ unfolding of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$ is defined by a matrix $\mathbf{X}^{(k)} \in \mathbb{R}^{I_k \times \Pi_{k' \neq k} I_{k'}}$, where the mode-$k$ folding is the process from $\mathbf{X}^{(k)}$ to $\mathcal{X}$. The operator $\times_k$ denotes the mode-$k$ product, and the mode-$k$ product of $\mathcal{X}$ with a matrix $\mathbf{U}$ is defined by $\mathcal{Y} = \mathcal{X} \times_k \mathbf{U}$, where $\mathbf{U} \in \mathbb{R}^{J_k \times I_k}$, $X \in \mathbb{R}^{I_1 \times \cdots \times I_k \times \cdots \times I_n}$, and $\mathcal{Y} \in \mathbb{R}^{I_1 \times \cdots \times J_k \times \cdots \times I_n}$. We can also rewrite $\mathcal{Y} = \mathcal{X} \times_k \mathbf{U}$ as $\mathbf{Y}^{(k)} = \mathbf{U}\mathbf{X}^{(k)}$ based on the mode-$k$ unfolding operation.

## 2.2 Low-Rank Matrix Approximation

The rank minimization problem has been widely studied and its robustness against noisy and missing data has been demonstrated. Given a corrupted matrix $\mathbf{M}$, the goal of low-rank matrix approximation is to solve

$$\hat{\mathbf{X}} = \operatorname{argmin} \operatorname{rank}(\mathbf{X}) \quad \text{s.t. } \|\mathbf{X} - \mathbf{M}\|_F^2 \le \varepsilon, \qquad (1)$$

or its Lagrangian version

$$\hat{\mathbf{X}} = \operatorname{argmin} \eta \operatorname{rank}(\mathbf{X}) + \frac{1}{2}\|\mathbf{X} - \mathbf{M}\|_F^2, \qquad (2)$$

where $\operatorname{rank}(\mathbf{X})$ denotes the rank of $\mathbf{X}$. However, rank minimization is generally NP-hard [34] because of its nonconvexity; hence, the global optimum cannot be guaranteed. According to Recht et al. [17], the nuclear norm is the tightest convex envelope of a matrix rank. Thus, we can represent the convex relaxation of (1) as follows:

$$\hat{\mathbf{X}} = \operatorname{argmin} \eta \|\mathbf{X}\|_* + \frac{1}{2}\|\mathbf{X} - \mathbf{M}\|_F^2, \qquad (3)$$

where $\mathbf{X}_* = \sum_i \sigma_i(\mathbf{X})$ denotes the nuclear norm, which is the sum of all the singular values, and $\sigma_i(\mathbf{X})$ denotes the $i$th largest singular value of $\mathbf{X}$.

Although (3) is nondifferentiable, the optimum solution $\mathbf{X}^*$ is guaranteed if and only if the subdifferential at $\mathbf{X}^*$ contains 0. The closed-form solution of (3) has been proved by Liu et al. [23]. Here, we only give their derived results. Interested readers may refer to [23] for further details. Let $\mathbf{M} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ be the singular value decomposition of $\mathbf{M}$. SVD shrinkage is used to obtain the global optimum of (3) in the following elegant closed form:

$$\mathbf{X}^* = \mathbf{U}\boldsymbol{\Sigma}_\eta \mathbf{V}^T, \qquad (4)$$

where $\boldsymbol{\Sigma}_\eta$ denotes the diagonal matrix with all the singular values of $\mathbf{M}$ that have been shrunk, i.e., $\max(\sigma_i(\mathbf{M}) - \eta, 0)$.

## 2.3 Augmented Lagrange Multiplier (ALM) Method

The ALM method was introduced in the mid-1970s and was well studied throughout the 1980s, to solve constrained optimization problems [35]. Although second-order methods (e.g., the interior-point method) are usually more precise, the ALM method has good scalability on large-scale cases as well as the flexibility to deal with nonsmooth functions [5], [6], [7], [8], [11], [19], [27], [47].

Here, we briefly explain how the ALM method solves equality constrained optimization problems. Let $f : \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R}^n \to \mathbb{R}^m$. Our objective is to solve

$$\hat{\mathbf{x}} = \operatorname{argmin} f(\mathbf{x}) \quad s.t. \ h(\mathbf{x}) = \mathbf{0}, \qquad (5)$$

where $\mathbf{x} \in \mathbb{R}^n$, and $\mathbf{0} \in \mathbb{R}^m$ contains all zero elements. The ALM method relaxes the constrained problem into an unconstrained problem by introducing an augmented Lagrange function:

$$L(\mathbf{x}, \boldsymbol{\lambda}, \mu) = f(\mathbf{x}) + \langle \boldsymbol{\lambda}, h(\mathbf{x}) \rangle + \frac{\mu}{2}\|h(\mathbf{x})\|_2^2. \qquad (6)$$

According to [35], we can find a global or local optimum of the original problem by iterative optimization as follows:

$$\mathbf{x}^{t+1} = \operatorname{argmin} L(\mathbf{x}, \boldsymbol{\lambda}^t, \mu^t), \qquad (7)$$

$$\boldsymbol{\lambda}^{t+1} = \boldsymbol{\lambda}^t + \mu^t h(\mathbf{x}^{t+1}), \ \text{ and} \qquad (8)$$

$$\mu^{t+1} = \rho \mu^t, \qquad (9)$$

where $t$ is the iteration index and $\rho$ is a penalty parameter larger than one. Equations (7)-(9) are called exact ALM (EALM). The superiority of the EALM's convergence property was proved in [35]. However, because the prima variables $\mathbf{x}$ are usually optimized blockwise in $f$ and $h$, finding the global optimum in (7) is time-consuming and difficult. To address this problem, the inexact ALM (IALM) [36] only optimizes each prima variable once in (7). The IALM's convergence has been well studied in relation to convex problems [37], but only a few works have investigated its convergence in nonconvex problems [38], [39]. Although the IALM lacks a theoretical guarantee, recent works [4], [5], [6], [7], [8] have reported promising results and suggest using the method because of its efficiency. The results also indicate that the IALM would be effective in real-world applications.

## 3 SIMULTANEOUS TENSOR DECOMPOSITION AND COMPLETION

Tensor representation has been widely studied as a structural foundation for visual data and the multilinear model in CP and Tucker decomposition. Because the CP model is a special case of the Tucker model with a superdiagonal core tensor, we focus on the Tucker model and the scenario where only a subset of entries $\Omega$ can be observed and the remaining $\bar{\Omega}$ are missing. Given an $n$th-order tensor $\mathcal{X}_0 \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$, we try to find a tensor $\mathcal{X}$ with its components $\mathcal{Z}, \mathbf{V}_1, \ldots, \mathbf{V}_n$ so that $\mathcal{X}_0$ and $\mathcal{X}$ have the same observed entries:

$$\mathcal{X} = \mathcal{Z} \times_1 \mathbf{V}_1^T \cdots \times_n \mathbf{V}_n^T \text{ and } \Omega(X) = \Omega(\mathcal{X}_0), \qquad (10)$$

where $\mathcal{Z}$ is an $n$th-order tensor of the same size as $\mathcal{X}_0$, and each $\mathbf{V}_k$ denotes an $I_k \times I_k$ matrix. According to the definition of mode-$k$ rank, if $\mathcal{X}$ is a low-rank tensor, then the core tensor $\mathcal{Z}$ is of low-rank or $\mathbf{V}_1, \ldots, \mathbf{V}_n$ are a set of low-rank matrices. Such a low-rank property is usually regarded as a global prior in tensor completion.

Note that (10) has infinite solutions if we do not include any priors in the model components $\mathcal{Z}, \mathbf{V}_1, \ldots, \mathbf{V}_n$. It has

been shown that the trace norm is the tightest convex envelope of the rank function for a matrix, but not for a tensor; therefore, we impose the low-rank penalty on $\mathbf{V}_1, \ldots, \mathbf{V}_n$ instead of on $\mathcal{Z}$. To further leverage the Tucker decomposition and rank minimization process, we introduce the concept of using factor priors as auxiliary information for tensor analysis in Section 3.1. We formulate the problem in Section 3.2 and then describe the proposed algorithm for simultaneous tensor decomposition and completion in Section 3.3.

## 3.1 Factor Priors for Tensor Analysis

In an $n$th-order tensor, each order represents one factor. Although a tensor could be comprised of randomly arranged elements, it is usually assumed that the within-factor and joint-factor variations are known a priori and can be regarded as auxiliary information. For example, a video object is a third-order tensor with variations spanned by the rows, columns, and time axis. Even when the value of an element is unknown, we may reasonably infer that adjacent rows/columns/frames are highly correlated. This is because the local similarity of visual data usually exists in within-factor relations (e.g., between adjacent rows/columns/frames) or joint-factor relations (e.g., between spatially adjacent and temporally adjacent pixels). In the modeling of social network [32] or facial images with pose, illumination, and expression (PIE) variations [22], their implicit semantics also reveal valuable information. In both cases, we call the auxiliary information *factor priors* for tensor analysis.

Because the factor priors encode the auxiliary relations, the underlying within- or joint-factor changes should lie on multiple low-dimensional submanifolds (corresponding to the $n$ factors) with restricted degrees of freedom. To characterize the submanifolds, we use the multilinear graph embedding (MGE) framework proposed in our previous work [33]. Suppose the observed data are drawn from $n$ factors, and the $k$th factor changes are quantized into $I_k$ variations for $1 \leq k \leq n$. Let $\mathcal{X}^{(i_1, \ldots, i_n)}$ denote the $i$th data drawn from the $n$ quantized factors $i_1, \ldots, i_n$ $(1 \leq i_k \leq I_k)$. Under the MGE framework, we parameterize $\mathcal{X}^{(i_1, \ldots, i_n)}$ by $\mathbf{v}_1^{i_1}, \ldots, \mathbf{v}_n^{i_n}$ and a core tensor $\mathcal{Z}$ as

$$\mathcal{X}^{(i_1, \ldots, i_n)} = \mathcal{Z} \times_1 \mathbf{v}_1^{i_1 \mathrm{T}} \times_2 \cdots \times_n \mathbf{v}_n^{i_n \mathrm{T}},$$
$$1 \leq i_k \leq I_k \text{ and } 1 \leq k \leq n. \quad (11)$$

The low-dimensional representation of $\mathcal{X}^{(i_1, \ldots, i_n)}$ can be written as $\mathbf{v}^{(i_1, \ldots, i_n)} = \mathbf{v}_1^{i_1} \otimes \cdots \otimes \mathbf{v}_n^{i_n}$, where $\otimes$ denotes the Kronecker product. Therefore, MGE extends the graph embedding framework [40] into the multilinear space:

$$\widehat{\mathbf{V}}_1, \ldots, \widehat{\mathbf{V}}_n = \operatorname{argmin} \sum_{i,j} \left\| \mathbf{v}^{(i_1, \ldots, i_n)} - \mathbf{v}^{(j_1, \ldots, j_n)} \right\|_2^2 w_{ij}$$
$$= \operatorname{argmin} \operatorname{tr}\left( (\mathbf{V}_1 \otimes \cdots \otimes \mathbf{V}_n) \mathbf{L} (\mathbf{V}_1 \otimes \cdots \otimes \mathbf{V}_n)^{\mathrm{T}} \right),$$
$$\mathbf{V}_k = \left[ \mathbf{v}_k^1, \ldots, \mathbf{v}_k^{I_k} \right] \text{ and } 1 \leq k \leq n, \quad (12)$$

where $\mathbf{V}_1, \ldots \mathbf{V}_n$ are the $n$ unknown submanifolds, $\operatorname{tr}(\mathbf{X})$ denotes the trace of $\mathbf{X}$, and $\mathbf{L} \in \mathbb{R}^{(I_1 I_2 \ldots I_n) \times (I_1 I_2 \ldots I_n)}$ is the so-called Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$. The $(i, j)$th element of $\mathbf{W}$ is $w_{ij}$ and $\mathbf{D}$ is a diagonal matrix whose $(i, i)$th element is

equal to $\sum_j w_{ij}$. To solve (12) with the MGE framework, we decompose the positive semidefinite matrix $\mathbf{L}$ into $\mathbf{H}^{\mathrm{T}}\mathbf{H}$ and then perform mode-$(n+1)$ folding on $\mathbf{H}$ to obtain $\mathcal{H}$. Finally, we apply the alternating minimizing scheme and optimize one variable at a time by

$$\widehat{\mathbf{V}}_k = \operatorname{argmin} \operatorname{tr}\left( \mathbf{V}_k \mathbf{H}_k^{\mathrm{T}} \mathbf{H}_k \mathbf{V}_k^{\mathrm{T}} \right), \quad (13)$$

where $\mathbf{H}_k^{\mathrm{T}}$ denotes the mode-$k$ unfolding of $\mathcal{H} \times_1 \mathbf{V}_1 \cdots \times_{k-1} \mathbf{V}_{k-1} \times_{k+1} \mathbf{V}_{k+1} \cdots \times_n \mathbf{V}_n$. Readers may refer to [33] for the detailed derivation.

In (12), the most important step is determining the edge weight $w_{ij}$, which should preserve both within- and joint-factor variations in the low-dimensional space. Next, we discuss two examples, the CMU-PIE face database [41] and the visual data used in Sections 4.2 and 4.3 to demonstrate our multilinear graph design.

### 3.1.1 MGE for the CMU-PIE Face Database

We consider a subset of facial images in the CMU-PIE database captured from different subjects under various poses and illumination changes. Therefore, the images should lie on three joint submanifolds. Because our goal is to characterize variations across the three factors, in contrast to [40], we use a factor-dependent strategy to define the edge between the $i$th and the $j$th face images as follows:

$$w_{ij} = \begin{cases} 1, & \text{if } i_k \in \mathrm{N}(j_k) \text{ or } j_k \in \mathrm{N}(i_k), \text{and } i_{\tilde{k}} = j_{\tilde{k}}, \\ 0, & \text{otherwise}, \end{cases} \quad (14)$$

where $\mathrm{N}(\cdot)$ is the neighborhood function (we use the 2-nearest-neighbor function (2-NN) in experiments) associated with the $k$th factor, $k \in \{\text{subject, pose, illumination}\}$, and $\tilde{k}$ denotes the complement set of $k$. The neighborhood functions for pose and illumination are known a priori (e.g., given five poses $\{-90, -45, 0, 45, 90 \text{ degrees}\}$, the two neighbors of pose 0 degree are the poses $\pm 45$ degrees). With regard to the subject neighborhood, we simply measure the average euclidean distance between two subject groups and then determine if they are 2-NN.

### 3.1.2 MGE for Visual Data

When an image object is regarded as a second-order tensor, the factor dependence can be measured in terms of the numerical difference between factor changes. Therefore, the joint-factor graph is defined as follows:

$$w_{ij} = \exp(-|i_{\text{row}} - j_{\text{row}}|) \exp(-|i_{\text{column}} - j_{\text{column}}|). \quad (15)$$

Equation (15) characterizes the pixelwise local similarity of visual data as precise factor priors. However, this joint-factor formulation would yield a very large Laplacian matrix $\mathbf{L} \in \mathbb{R}^{(I_{\text{row}} I_{\text{column}}) \times (I_{\text{row}} I_{\text{column}})}$. In practice, we suggest using a single-factor (or within-factor) graph built on a smaller Laplacian matrix:

$$\operatorname{argmin} \sum_{i,j} \left\| \mathbf{v}^{(i_k)} - \mathbf{v}^{(j_k)} \right\|_2^2 w_{ij} = \operatorname{argmin} \operatorname{tr}\left( \mathbf{V}_k \mathbf{L}_k \mathbf{V}_k^{\mathrm{T}} \right), \text{ and}$$
$$w_{ij} = \exp(-|i_k - j_k|), k \in \{\text{row, column}\}, \quad (16)$$

where the size of $\mathbf{L}_k$ is either $I_{\text{row}} \times I_{\text{row}}$ or $I_{\text{column}} \times I_{\text{column}}$.

Note that (16) is a special case where MGE is used to encode one submanifold. Because of the generality of MGE, the factor priors can encode any subset of submanifolds $\Xi \subseteq \{\mathbf{V}_k \mid 1 \le k \le n\}$. Hence, we define the factor priors by

$$\mathrm{tr}\big(\mathbf{V}_\Xi \mathbf{L}_\Xi \mathbf{V}_\Xi^\mathrm{T}\big), \tag{17}$$

where $\mathbf{V}_\Xi$ denotes the Kronecker product of all elements in $\Xi$. In (17), the definition of $\mathbf{L}_\Xi$ only relies on priors about the within- or joint-factor relations, even with unknown entries. In tensor completion, we consider the proposed factor priors as regularization terms that can be used to capture a low-dimensional representation.

## 3.2 Maximum a Posteriori (MAP) Formulation

Based on the feasible set defined in (10), we use the *maximum a posteriori* strategy to find $\mathcal{X}, \mathcal{Z}, \mathbf{V}_1, \ldots, \mathbf{V}_n$:

$$\begin{aligned}
\hat{\mathcal{X}}, \hat{\mathcal{Z}}, \hat{\mathbf{V}}_1, \ldots, \hat{\mathbf{V}}_n &= \arg\max \mathrm{p}(\mathcal{Y}, \mathcal{Z}, \mathbf{V}_1, \ldots, \mathbf{V}_n \mid \mathcal{X}_0) \\
&= \arg\max \mathrm{p}(\mathcal{X}_0 \mid \mathcal{X})\mathrm{p}(\mathcal{X} \mid \mathcal{Z}, \mathbf{V}_1, \ldots, \mathbf{V}_n) \\
&\quad \times \mathrm{p}(\mathcal{Z})\mathrm{p}(\mathbf{V}_1, \ldots, \mathbf{V}_n).
\end{aligned} \tag{18}$$

Because we only consider nonnoisy observations and assume that $\mathcal{X}$ follows Tucker decomposition, the two likelihood terms $\mathrm{p}(\mathcal{X}_0 \mid \mathcal{X})$ and $\mathrm{p}(\mathcal{X} \mid \mathcal{Z}, \mathbf{V}_1, \ldots, \mathbf{V}_n)$ are always equal to one and can, therefore, be ignored. Note that our MAP formulation can be easily extended to noisy cases by including $\mathrm{p}(\mathcal{X}_0 \mid \mathcal{X}) = \exp(-\|\Omega(\mathcal{X}) - \Omega(\mathcal{X}_0)\|_\mathrm{F}^2)$.

To define the prior probability $\mathrm{p}(\mathcal{Z})$, we follow [31] and assume that every element is drawn from a zero-mean spherical Gaussian prior:

$$\begin{aligned}
\mathrm{p}(\mathcal{Z}) &= \prod_{i_1} \cdots \prod_{i_n} \exp(-\gamma \mathcal{Z}_{i_1,\ldots,i_n}^2) \\
&= \exp\left(-\gamma \sum_{i_1} \cdots \sum_{i_n} \mathcal{Z}_{i_1,\ldots,i_n}^2\right) = \exp\big(-\gamma\|\mathcal{Z}\|_\mathrm{F}^2\big).
\end{aligned} \tag{19}$$

The role of $\mathrm{p}(\mathcal{Z})$ here is to prevent overfitting of the Tucker model instead of to capture the underlying low-rank structure. This simple but useful prior also circumvents the complex low-rank approximation of high-order tensor objects.

The modeling of $\mathbf{V}_1, \ldots, \mathbf{V}_n$, which characterizes the implicit complexity of both low-rank and low-dimensional manifolds in tensor objects, provides the foundation for our method. With the low-rank matrix assumption and the proposed factor priors in (17), we define the prior probability $\mathrm{p}(\mathbf{V}_1, \ldots, \mathbf{V}_n)$ as

$$\begin{aligned}
\mathrm{p}(\mathbf{V}_1, \ldots, \mathbf{V}_n) &= \exp(-\beta\mathrm{tr}(\mathbf{V}\mathbf{L}\mathbf{V}^\mathrm{T}))\prod_{k=1}^n \exp(-\alpha_k\|\mathbf{V}_k\|_*) \\
&= \exp\left(-\left(\beta\mathrm{tr}(\mathbf{V}\mathbf{L}\mathbf{V}^\mathrm{T}) + \sum_{k=1}^n \alpha_k\|\mathbf{V}_k\|_*\right)\right) \\
&\quad \text{and } \mathbf{V} = \mathbf{V}_1 \otimes \cdots \otimes \mathbf{V}_n.
\end{aligned} \tag{20}$$

To simplify the notations in our derivation, here we only consider the factor priors defined by all submanifolds; i.e.,

$\Xi = \{\mathbf{V}_1, \ldots, \mathbf{V}_n\}$. The formulation could be modified by including more factor priors.

By taking the negative log probability of (19) and (20) under the feasible set defined in (10), we can convert our MAP formulation into a constrained minimization problem as follows:

$$\begin{aligned}
\hat{\mathcal{X}}, \hat{\mathcal{Z}}, \hat{\mathbf{V}}_1, \ldots, \hat{\mathbf{V}}_n &= \arg\min \sum_{k=1}^n \alpha_k\|\mathbf{V}_k\|_* \\
&\quad + \beta\mathrm{tr}\big((\mathbf{V}_1 \otimes \cdots \otimes \mathbf{V}_n)\mathbf{L}(\mathbf{V}_1 \otimes \cdots \otimes \mathbf{V}_n)^\mathrm{T}\big) \\
&\quad + \gamma\|\mathcal{Z}\|_\mathrm{F}^2 \\
&\text{s.t. } \mathcal{X} = \mathcal{Z} \times_1 \mathbf{V}_1^\mathrm{T} \cdots \times_n \mathbf{V}_n^\mathrm{T} \text{ and } \Omega(\mathcal{X}) = \Omega(\mathcal{X}_0).
\end{aligned} \tag{21}$$

## 3.3 Iterative Convex Optimization via IALM

To solve (21) with a large number of unknowns and equality constraints, we use the ALM method to define the augmented Lagrange function as follows:

$$\begin{aligned}
&\mathrm{L}(\mathcal{X}, \mathcal{Z}, \mathbf{V}_1, \ldots, \mathbf{V}_n, \mathcal{Y}, \mu) \\
&= \sum_{k=1}^n \alpha_k\|\mathbf{V}_k\|_* + \beta\mathrm{tr}\big((\mathbf{V}_1 \otimes \cdots \otimes \mathbf{V}_n)\mathbf{L}(\mathbf{V}_1 \otimes \cdots \otimes \mathbf{V}_n)^\mathrm{T}\big) \\
&\quad + \gamma\|\mathcal{Z}\|_\mathrm{F}^2 + \langle \mathcal{Y}, \mathcal{X} - \mathcal{Z} \times_1 \mathbf{V}_1^\mathrm{T} \cdots \times_n \mathbf{V}_n^\mathrm{T}\rangle \\
&\quad + \frac{\mu}{2}\big\|\mathcal{X} - \mathcal{Z} \times_1 \mathbf{V}_1^\mathrm{T} \cdots \times_n \mathbf{V}_n^\mathrm{T}\big\|_\mathrm{F}^2.
\end{aligned} \tag{22}$$

Because of the operations of the Kronecker product and the mode-$k$ product, (22) is a nonconvex function that contains many local minimizers. However, if we optimize one variable at a time with the others fixed, every subproblem can be simplified to a convex optimization problem with a global optimum. As noted in [19], at best we can find one local minimizer. Given the IALM algorithm's efficiency, we use it to optimize the prima variables $\mathbf{V}_1, \ldots, \mathbf{V}_n, \mathcal{Z}, \mathcal{X}$ as described below.

### 3.3.1 Optimization of $\mathbf{V}_k$

By performing mode-$k$ unfolding and substituting the result derived in (13), we have the following sub-Lagrange function with respect to $\mathbf{V}_k$:

$$\begin{aligned}
\mathrm{L}_{\mathbf{V}_k}(\mathbf{V}_k) &= \alpha_k\|\mathbf{V}_k\|_* + \beta\mathrm{tr}\big(\mathbf{V}_k\mathbf{H}_k^\mathrm{T}\mathbf{H}_k\mathbf{V}_k^\mathrm{T}\big) \\
&\quad + \big\langle \mathbf{Y}^{(k)}, -\mathbf{V}_k^\mathrm{T}\mathbf{Z}^{(k)}\mathbf{U}_k\big\rangle \\
&\quad + \frac{\mu}{2}\big\|\mathbf{X}^{(k)} - \mathbf{V}_k^\mathrm{T}\mathbf{Z}^{(k)}\mathbf{U}_k\big\|_\mathrm{F}^2 \\
&\text{and } \mathbf{U}_k = (\mathbf{V}_1 \otimes \cdots \otimes \mathbf{V}_{k-1} \otimes \mathbf{V}_{k+1} \otimes \cdots \otimes \mathbf{V}_n).
\end{aligned} \tag{23}$$

If we disregard the constant terms and split the nondifferential terms, we obtain the subproblem:

$$\widehat{\mathbf{V}_k} = \arg\min \mathrm{f}_1(\mathbf{V}_k) + \mathrm{f}_2(\mathbf{V}_k),$$

$$\text{where} \quad \mathrm{f}_1(\mathbf{V}_k) = \alpha_k\|\mathbf{V}_k\|_* \text{ and}$$

$$\begin{aligned}
\mathrm{f}_2(\mathbf{V}_k) = \mathrm{tr}\Big(&\beta\mathbf{V}_k\mathbf{H}_k^\mathrm{T}\mathbf{H}_k\mathbf{V}_k^\mathrm{T} + \frac{\mu}{2}\mathbf{V}_k^\mathrm{T}\mathbf{Z}^{(k)}\mathbf{U}_k\mathbf{U}_k^\mathrm{T}\mathbf{Z}^{(k)\mathrm{T}}\mathbf{V}_k \\
&- \mathbf{V}_k^\mathrm{T}\mathbf{Z}^{(k)}\mathbf{U}_k(\mu\mathbf{X}^{(k)} + \mathbf{Y}^{(k)})^\mathrm{T}\Big).
\end{aligned} \tag{24}$$

Equation (24) is a typical nuclear norm minimization problem, but there is no closed-form solution because $f_2(\mathbf{V}_k)$ does not fit the least-squares form in (3). Some algorithms (e.g., [14], [42]) try to resolve such problems in the general form of a continuous function. In our case, because $f_2(\mathbf{V}_k)$ is quadratic, its function behavior can be well approximated by a local linearization technique. We use the technique proposed in [42] and obtain

$$
\begin{aligned}
\widehat{\mathbf{V}}_k &\approx \arg\min f_1(\mathbf{V}_k) + f_2\big(\mathbf{V}_k^t\big) + \big\langle \nabla f_2\big(\mathbf{V}_k^t\big), \mathbf{V}_k - \mathbf{V}_k^t \big\rangle \\
&\quad + \frac{\tau}{2}\big\|\mathbf{V}_k - \mathbf{V}_k^t\big\|_{\mathrm{F}}^2 \\
&= \arg\min \frac{\alpha_k}{\tau}\|\mathbf{V}_k\|_* + \frac{1}{2}\left\|\mathbf{V}_k - \left(\mathbf{V}_k^t - \frac{1}{\tau}\nabla f_2\big(\mathbf{V}_k^t\big)\right)\right\|_{\mathrm{F}}^2, \text{ and} \\
\nabla f_2\big(\mathbf{V}_k^t\big) &= 2\beta \mathbf{V}_k^t \mathbf{H}_k^{\mathrm{T}} \mathbf{H}_k + \mu \mathbf{Z}^{(k)} \mathbf{U}_k \mathbf{U}_k^{\mathrm{T}} \mathbf{Z}^{(k)\mathrm{T}} \mathbf{V}_k^t \\
&\quad - \mathbf{Z}^{(k)} \mathbf{U}_k\big(\mu \mathbf{X}^{(k)} + \mathbf{Y}^{(k)}\big)^{\mathrm{T}},
\end{aligned}
$$
(25)

where $\mathbf{V}_k^t$ is the approximation from the previous iteration, and $\tau$ is the Lipschitz constant defined by

$$
\tau = \sigma_1\big(2\beta \mathbf{H}_k^{\mathrm{T}} \mathbf{H}_k\big) + \sigma_1\big(\mu \mathbf{Z}^{(k)} \mathbf{U}_k \mathbf{U}_k^{\mathrm{T}} \mathbf{Z}^{(k)\mathrm{T}}\big). \tag{26}
$$

Finally, we solve (25) by low-rank matrix approximation.

### 3.3.2 Optimization of $\mathcal{Z}$

To minimize the sub-Lagrange function with respect to $\mathcal{Z}$, we have

$$
\hat{\mathcal{Z}} = \arg\min \gamma \|\mathcal{Z}\|_{\mathrm{F}}^2 + \frac{\mu}{2}\left\|\left(\mathcal{X} + \frac{1}{\mu}\mathcal{Y}\right) - \mathcal{Z} \times_1 \mathbf{V}_1^{\mathrm{T}} \cdots \times_n \mathbf{V}_n^{\mathrm{T}}\right\|_{\mathrm{F}}^2. \tag{27}
$$

If we vectorize $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ as $\mathbf{x}, \mathbf{y}, \mathbf{z}$, we obtain a typical least-squares problem with ridge regression:

$$
\hat{\mathbf{z}} = \arg\min \gamma \|\mathbf{z}\|_2^2 + \frac{\mu}{2}\left\|\left(\mathbf{x} + \frac{1}{\mu}\mathbf{y}\right) - \mathbf{V}^{\mathrm{T}}\mathbf{z}\right\|_2^2 \tag{28}
$$

$$
\text{and } \mathbf{V} = \mathbf{V}_1 \otimes \cdots \otimes \mathbf{V}_n.
$$

Although (28) has the closed-form solution $\hat{\mathbf{z}} = (\mu \mathbf{V}\mathbf{V}^{\mathrm{T}} + 2\gamma \mathbf{I})^{-1}\mathbf{V}(\mathbf{x} + \frac{1}{\mu}\mathbf{y})$, this linear system could be very difficult to solve when the size of $\mathbf{V}$ is extremely large. In this paper, we use the conjugate gradient (CG) method to approximate the global minimizer.

### 3.3.3 Optimization of $\mathcal{X}$

By minimizing the sub-Lagrange function with respect to $\mathcal{X}$, we obtain

$$
\hat{\mathcal{X}} = \arg\min \left\|\mathcal{X} - \left(\mathcal{Z} \times_1 \mathbf{V}_1^{\mathrm{T}} \cdots \times_n \mathbf{V}_n^{\mathrm{T}} - \frac{1}{\mu}\mathcal{Y}\right)\right\|_{\mathrm{F}}^2. \tag{29}
$$

Then, with the second equality constraint in (10), we derive

$$
\Omega(\mathcal{X}) = \Omega(\mathcal{X}_0) \text{ and } \bar{\Omega}(\mathcal{X}) = \bar{\Omega}\left(\mathcal{Z} \times_1 \mathbf{V}_1^{\mathrm{T}} \ldots \times_n \mathbf{V}_n^{\mathrm{T}} - \frac{1}{\mu}\mathcal{Y}\right). \tag{30}
$$

Algorithm 1 summarizes the steps of the *simultaneous tensor decomposition and completion* method. Although our

formulation cannot guarantee the global minimizer, the proposed factor priors and the components of the Tucker model characterize the tensor structure and improve the search space, thereby ensuring the best possible recovery rate.

**Algorithm 1.** Simultaneous Tensor Decomposition and
Completion.
**Input:** an incomplete tensor $\mathcal{X}_0 \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$, a positive
　　　 semi-definite matrix $\mathbf{L}$, and the parameters
　　　 $\alpha_1, \ldots, \alpha_n, \beta, \gamma, \mu$.
1) Initialize $\mathbf{V}_1, \ldots, \mathbf{V}_n, \mathcal{Z}, \mathcal{X}, \mathcal{Y}$ by
　　 $\mathbf{V}_k$: identity matrix $(1 \le k \le n)$,
　　 $\mathcal{Z}, \mathcal{X}$: $\mathcal{X}_0$, and
　　 $\mathcal{Y}$: a tensor with all zero elements.
2) Let $t = 1$ and $\mu^1 = \mu$.
3) **Do**
　　 For $k = 1$ to $n$
　　 Optimize $\mathbf{V}_k$ with other variables fixed, as per Sec. 3.3.1.
　　 End
　　 Optimize $\mathcal{Z}$ with other variables fixed, as per Sec. 3.3.2.
　　 Optimize $\mathcal{X}$ with other variables fixed, as per Sec. 3.3.3.
　　 Update $\mathcal{Y}$ by $\mathcal{Y} + \mu^t(\mathcal{X} - \mathcal{Z} \times_1 \mathbf{V}_1^{\mathrm{T}} \cdots \times_n \mathbf{V}_n^{\mathrm{T}})$.
　　 $\mu^{t+1} = \rho\mu^t, \rho \in [1.1, 1.2]$.
　　 $t = t + 1$.
　　 **While** $t < 100$ and $\|\mathcal{X} - (\mathcal{Z} \times_1 \mathbf{V}_1^{\mathrm{T}} \cdots \times_n \mathbf{V}_n^{\mathrm{T}})\|_{\mathrm{F}}^2 >$
　　 $10^{-4}\|\mathcal{X}_0\|_{\mathrm{F}}^2$
**Output:** the submanifolds $\mathbf{V}_1, \ldots, \mathbf{V}_n$, the core tensor $\mathcal{Z}$,
　　　　 and the complete tensor $\mathcal{X}$.

## 3.4 Convergence of the STDC Algorithm

Algorithm 1 exploits two key techniques, IALM and linearization, to solve (21). The convergence of IALM has been proven for separable convex problems [37], convex problems using linearization [42], and nonconvex problems (e.g., low-rank matrix factorization [38], [39]). The most similar work to our approach is [39], which combines matrix factorization and rank minimization to reduce the computational complexity. However, its elegant convergence property may not hold when the linearization technique is used. Since our model is much more complicated than that in [39] (because of the high-order tensor structure and factor priors), it is difficult to derive a theoretical guarantee for Algorithm 1. Instead, we provide empirical evidence to demonstrate the feasibility and applicability of our method. To alleviate the model's sensitivity to the algorithm's parameters, in Section 3.5.2, we simplify the parameters $\alpha_1, \ldots, \alpha_N, \beta, \gamma, \mu$ by exploring a tradeoff between the different objectives. This strategy yields the physical meaning of the parameters related to the factor priors and the Tucker model. It also reduces the number of parameters so that only two need to be determined. In Section 4.1, we verify the convergence of Algorithm 1 on synthetic data and demonstrate that it converges within a wide range of parameter values.

## 3.5 Implementation Issues

### 3.5.1 Efficient Variants of STDC

In real-world applications, a tensor may be extremely large such that construction of the Laplacian matrix in (12) becomes infeasible. Although the factor priors are usually well defined by a small subset $\Xi$, our method is affected by

the above limitation and an efficient solution is therefore needed. Here, we propose a simple method based on downsampling the submanifolds. Given the factor priors in (21), we define the small-scale version of the Laplacian matrix as follows:

$$\mathrm{tr}\big((\mathbf{V}_1\mathbf{S}_1 \otimes \cdots \otimes \mathbf{V}_n\mathbf{S}_n)\mathbf{L}_{\mathrm{s}}(\mathbf{V}_1\mathbf{S}_1 \otimes \cdots \otimes \mathbf{V}_n\mathbf{S}_n)^{\mathrm{T}}\big), \quad (31)$$

where $\mathbf{S}_k \in \mathbb{R}^{I_k \times R_k}$ is a bilinear downsampling matrix with $R_k \ll I_k$; and $\mathbf{L}_{\mathrm{s}} \in \mathbb{R}^{\Pi_{k=1}^n R_k \times \Pi_{k=1}^n R_k}$ denotes the small-scale Laplacian matrix. When $\prod_{k=1}^n R_k < 5{,}000$, the MGE method can efficiently decomposes $\mathbf{L}_{\mathrm{s}}$. However, this strategy only works when the factor variation is also characterized in its small-scale factor (e.g., the local similarity of an image is also preserved in its low resolution). In addition, when we optimize $\mathbf{V}_k$, the Hessian matrix $\mathbf{S}_k\mathbf{H}_k^{\mathrm{T}}\mathbf{H}_k\mathbf{S}_k^{\mathrm{T}}$ is singular and may lead to an ill-condition problem in $\mathrm{f}_2(\mathbf{V}_k)$. Therefore, we have to split (31) into $n$ cases by maintaining the original scale of $\mathbf{V}_k$ as follows:

$$\mathrm{tr}\big(\mathbf{P}_k\mathbf{L}_{\mathrm{s}}^k\mathbf{P}_k^{\mathrm{T}}\big) \text{ and } \mathbf{P}_k = (\mathbf{V}_1\mathbf{S}_1 \otimes \cdots \otimes \mathbf{V}_{k-1}\mathbf{S}_{k-1} \otimes \mathbf{V}_k \\ \otimes \mathbf{V}_{k+1}\mathbf{S}_{k+1} \otimes \cdots \otimes \mathbf{V}_n\mathbf{S}_n). \quad (32)$$

### 3.5.2 Parameter Settings

The drawback of the model in (21) is its sensitivity to the parameters. To tune the parameters, we try to balance different objectives in the augmented Lagrange function. Recall that in Algorithm 1, $\mathbf{V}_1, \ldots, \mathbf{V}_n$ are initialized as identity matrices and $\mathcal{Z}$ is initialized as $\mathcal{X}_0$. In the first iteration, we control the tradeoff in (22) in terms of $\kappa, \omega, \delta$ as follows:

$$\alpha_1 = \cdots = \alpha_n = 1, \quad (33)$$

$$\beta = \kappa^2 \frac{\mu\sigma_1(\mathbf{X}^{(1)}\mathbf{X}^{(1)^{\mathrm{T}}})/2}{\sigma_1(\mathbf{H}^{(1)}\mathbf{H}^{(1)^{\mathrm{T}}})}, \quad (34)$$

$$\mu = \frac{1}{\delta\sigma_1(\mathbf{X}^{(1)}\mathbf{X}^{(1)^{\mathrm{T}}})}, \text{ and} \quad (35)$$

$$\gamma = \frac{\omega}{\delta\|\mathcal{X}\|_{\mathrm{F}}^2}. \quad (36)$$

Note that the values of $\alpha_1, \ldots, \alpha_n$ are usually application-dependent and related to their dimensions $I_1, \ldots, I_n$. For example, in color images/videos, we usually want to preserve all the chromatic information and thus set the value of $\alpha$ that corresponds to the color factor as zero. Here, to better explain the physical meaning of the other parameters related to the factor priors and the Tucker model, we simply set $\alpha = 1$. If a low-rank penalty that has an equal impact on different tensor orders is required, we suggest using $\alpha_i = \sqrt{I_{\max}/I_i}$ ($I_{\max}$ is the maximum of $I_1, \ldots, I_n$).

The parameter $\kappa$ in (34) controls the tradeoff between the factor priors and the Tucker model in (26); that is, the Lipschitz constant $\tau$ in (26) would be multiples of $\mu\sigma_1(\mathbf{X}^{(1)}\mathbf{X}^{(1)^{\mathrm{T}}})$. For the parameter $\delta$, if we set $\alpha_1 = 1$ as in (33) and substitute (35) into $\mu$ in (26), then $\delta$ would directly control the first threshold for SVD shrinkage (because $\frac{\alpha_1}{\tau}$ also indicates the first threshold). Recall that the initial $\mathbf{V}_1$ is

an identity matrix, i.e., $\sigma_1(\mathbf{V}_1) = 1$. Therefore, to avoid severe truncation, we set $\delta = 0.1$ in all the experiments. Finally, we substitute $\delta$ in (35) into (36) so that the parameter $\omega$ controls the tradeoff between the $\mathcal{Z}$-priors and Tucker model.

Based on the above discussion, if we set $\kappa = \omega = 1$, we can balance the impacts of the factor priors, $\mathcal{Z}$-priors, and Tucker model. Equations (33)-(36) explain the properties of $\kappa, \omega$ and also reduce the number of parameters. The strategy is also independent of the actual range of $\mathcal{X}$ and $\mathbf{L}$, so it is applicable in real-world cases.

### 3.6 Advantages over Existing Work

Next, we consider the advantages of STDC over existing tensor completion approaches. Instead of using a predefined small size of $\mathcal{Z}$ like *factorization schemes*, we exploit nuclear norm minimization to capture the low-rank structure. The advantage is twofold. First, the exact rank is usually over- or underestimated in real-world applications. Some works suggest that an overestimated rank can help approximate missing entries effectively. However, when the tensor's rank is extremely low, the performance of tensor completion may still deteriorate on general tensor objects, especially those with a high missing data rate (as shown by our results in Section 4.3). Second, as most methods rely on (30) to update the missing entries, the low-rank penalty determines how much information about the current update is preserved in each iteration. Using (30), *factorization schemes* always discard a fixed amount of information, but STDC can capture the significant information adaptively. Therefore, *completion schemes* generally outperform *factorization schemes* in terms of the accurate recovery of missing data. The *completion scheme* proposed in [23] solves the tensor completion problem as follows:

$$\hat{\mathcal{X}} = \arg\min \sum_{k=1}^n \|\mathbf{M}_k\|_* \quad \text{s.t. } \mathbf{X}^{(k)} = \mathbf{M}_k \\ \text{and } \Omega(\mathcal{X}) = \Omega(\mathcal{X}_0). \quad (37)$$

Although the above *completion scheme* can approximate the exact rank without any model assumption, the intrinsic structure is only considered in the constraints $\mathbf{X}^{(k)} = \mathbf{M}_k$. However, the constraints may not be satisfied and may be oversimplified for factor estimates (as shown by our results in Section 4.2).

Finally, we consider the difference between our factor priors, which fully utilize the auxiliary information about a tensor object, and a similar approach recently proposed in [32]. To improve the *factorization scheme*, the authors also use graph embedding [40] to define within-mode regularization $\sum_{k=1}^n \mathrm{tr}(\mathbf{V}_k(\mathbf{D}_k - \mathbf{W}_k)\mathbf{V}_k^{\mathrm{T}})$ and cross-mode regularization $\prod_{k=1}^n \mathrm{tr}(\mathbf{V}_k\mathbf{D}_k\mathbf{V}_k^{\mathrm{T}}) - \prod_{k=1}^n \mathrm{tr}(\mathbf{V}_k\mathbf{W}_k\mathbf{V}_k^{\mathrm{T}})$. The two regularization terms are constructed on single-factor Laplacian matrices $\mathbf{L}_k = \mathbf{D}_k - \mathbf{W}_k (1 \le k \le n)$, whereas our factor priors are derived from the joint-factor Laplacian matrix. Therefore, our method theoretically outperforms [32] because it captures the underlying submanifolds simultaneously.

## 4 EXPERIMENT RESULTS

In this section, we compare the proposed STDC algorithm with the following state-of-the-art methods: LRTC [23],

TABLE 1
The Factor Priors of the STDC Variants
Discussed n Section 4.1

| | Factor priors |
|---|---|
| STDC-L1 | $\sum_{k=1}^{3} \mathrm{tr}(\mathbf{V}_k \mathbf{L}_k \mathbf{V}_k^{\mathrm{T}})$ |
| STDC-L2 | $\sum_{(k_1,k_2)\in\{(1,2),(2,3),(1,3)\}} \mathrm{tr}((\mathbf{V}_{k_1}\otimes\mathbf{V}_{k_2})\mathbf{L}_{k_1 k_2}(\mathbf{V}_{k_1}\otimes\mathbf{V}_{k_2})^{\mathrm{T}})$ |
| STDC-L3 | $\mathrm{tr}((\mathbf{V}_1\otimes\mathbf{V}_2\otimes\mathbf{V}_3)\mathbf{L}(\mathbf{V}_1\otimes\mathbf{V}_2\otimes\mathbf{V}_3)^{\mathrm{T}})$ |
| STDC-Ls | $\sum_{k=1}^{3} \mathrm{tr}(\mathbf{P}_k \mathbf{L}_s^k \mathbf{P}_k^{\mathrm{T}})$ |

HaLRTC [27], M$^2$SA [22], and M$^2$SA-G [32]. LRTC was the first work to define the tensor trace norm and it is considered the baseline of completion-based methods. HaLRTC is an extension of LRTC and its effectiveness over existing works has been demonstrated. M$^2$SA is an EM-like algorithm that is the baseline of factorization-based methods, and M$^2$SA-G extends M$^2$SA by including auxiliary graph regularization.

### 4.1 Validation of STDC on Synthetic Data

First, we validate STDC on a synthetic third-order tensor $\mathcal{X} \in \mathbb{R}^{20 \times 14 \times 10}$ constructed by the Tucker model using a low-rank core tensor $\mathcal{Z} \in \mathbb{R}^{14 \times 8 \times 6}$ and an underlying joint-manifold. The entries of $\mathcal{Z}$ are randomly drawn from uniform distribution. For the joint-manifold, we construct a sixth-order weight tensor $\mathcal{W} \in \mathbb{R}^{20 \times 14 \times 10 \times 20 \times 14 \times 10}$ and define its elements by $\mathcal{W}_{i_1,j_1,k_1,i_2,j_2,k_2} = e^{-|i_1-i_2|} + e^{-|j_1-j_2|} + e^{-|k_1-k_2|}$. Then we unfold $\mathcal{W}$ into $\mathbf{W} \in \mathbb{R}^{2800 \times 2800}$ to obtain its Laplacian matrix $\mathbf{L}$. Using MGE [33], we obtain the low-dimensional matrices $\mathbf{V}_1 \in \mathbb{R}^{14 \times 20}, \mathbf{V}_2 \in \mathbb{R}^{8 \times 14}, \mathbf{V}_3 \in \mathbb{R}^{6 \times 10}$ and determine that $\mathcal{X} = \mathcal{Z} \times_1 \mathbf{V}_1^{\mathrm{T}} \times_2 \mathbf{V}_2^{\mathrm{T}} \times_3 \mathbf{V}_3^{\mathrm{T}}$ accordingly.

Here, we compare HaLRTC with five variants of STDC: STDC-Lx, STDC-L1, STDC-L2, STDC-L3, and STDC-Ls. Only STDC-Lx does not use auxiliary information. The factor priors of the other four variants are listed in Table 1. $\mathbf{L}_k$ and $\mathbf{L}_{k_1 k_2}$ are constructed in a similar way to $\mathbf{L}$ by considering second-order and fourth-order weight tensors, respectively, and $\mathbf{L}_s^k$ denotes the small-scale version of $\mathbf{L}$, whose downsampling rate is set at two (i.e., $2R_k = I_k$). Next, we perform tensor completion under four missing data rates (60, 70, 80, 90 percent), using $\omega \in [10^{-1}, 10^1]$ for STDC-Lx and $\omega \in [10^{-3}, 10^{-1}], \kappa \in [10^0, 10^2]$ for the others. The relative square error $\mathrm{RSE} = \|\mathcal{X}\text{-}\hat{\mathcal{X}}\|_{\mathrm{F}}/\|\mathcal{X}\|_{\mathrm{F}}$ is used to evaluate the performance. Because we cannot prove theoretically that Algorithm 1 converges to a local optimum, we investigate whether STDC can recover the ground truth in a stable manner. If the RSE fluctuates severely with a frequency of $(\mathrm{RSE}^t\text{-}\mathrm{RSE}^{t-1} > 0.001)^2$ larger than 0.3, we consider it a case of nonconvergence and set the RSE $= 1$; otherwise, we determine that the algorithm converges and report the final RSE. All the experiments were repeated ten times with randomly missing entries and the average results are reported.

Fig. 2 shows the convergence rate and RSE with different settings of $\omega$ for STDC-Lx (no factor prior). Larger $\omega$ settings

2. We find that when STDC imposes strong factor priors, it tends to achieve the lowest RSE initially, and then gradually increases the RSE in a very small range (usually less than 0.01) to converge to a local optimum. Therefore, we only consider cases where the reversal of RSE is larger than 0.001 in consecutive iterations to avoid such miscalculations.
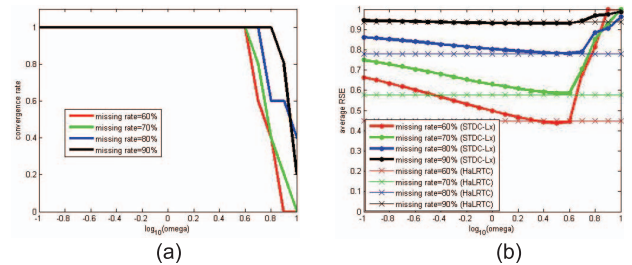


Fig. 2. The performance of STDC-Lx: (a) the convergence rate, and (b) the average RSE; the $x$-axis denotes $\log_{10}(\omega)$.

generally avoid overfitting the Tucker model and yield lower RSEs (see Fig. 2b), but if $\omega$ is too large, the nonconvergence of Algorithm 1 severely degrades the performance (see Fig. 2a). However, by carefully tuning $\omega$, STDC-Lx achieves a comparable performance to HaLRTC, even though it is still very sensitive to the parameters. Next, we consider the convergence rates of the other STDC variants to verify the significance of the factor priors, which leverage the Tucker decomposition and rank minimization method. Fig. 3 shows the convergence rates of the other STDC variants with respect to the parameters $\omega$ and $\kappa$. Our algorithm achieves a higher convergence rate with smaller $\kappa$ (i.e., there is less overfitting of the factor priors), especially when the missing data rate increases or the factor priors are not very accurate. In the cases with the highest missing data rate, i.e., 90 percent, STDC-L1 converges within a smaller range of the parameters; however, in most cases, STDC-L2, STDC-L3, and STDC-Ls still converge successfully. In other words, if we can capture the joint-manifold, a high convergence rate can be expected even with very sparse observations. The corresponding RSEs are shown in Fig. 4. The red rectangles indicate the parameter settings where STDC outperforms HaLRTC. Generally, when we use more accurate factor priors, we obtain a better performance with smaller $\omega$ and larger $\kappa$ settings. This is reasonable because larger $\kappa$ settings impose strong factor priors, while smaller
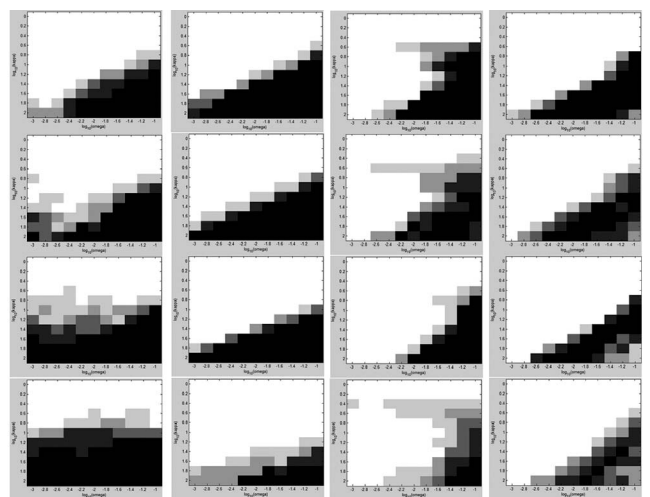


Fig. 3. The convergence rates of STDC-L1, STDC-L2, STDC-L3, and STDC-Ls (from left to right) under missing data rates of 60, 70, 80, and 90 percent (from top to bottom). The $x$-axis and $y$-axis denote $\log_{10}(\omega)$ and $\log_{10}(k)$, respectively, and the changes in intensity from black to white indicate the value range from 0 to 1.
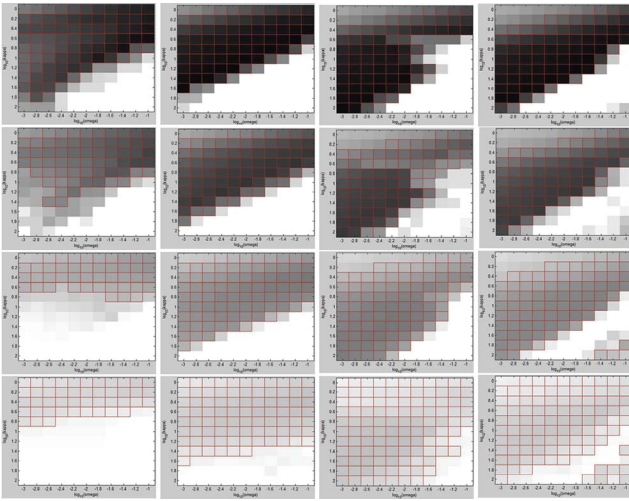
Fig. 4. The average RSEs of STDC-L1, STDC-L2, STDC-L3, and STDC-Ls (from left to right) under different missing rates 60, 70, 80, and 90 percent (from top to bottom). The $x$-axis and $y$-axis denote $\log_{10}(\omega)$ and $\log_{10}(k)$, respectively, and the changes in intensity from black to white indicate the value range from 0 to 1. The red rectangles indicate that STDC outperforms HaLRTC on those parameter settings.

$\omega$ settings simultaneously alleviate the underfitting problem. Hence, we can easily tune the two parameters to ensure that STDC achieves a good performance.

Table 2 shows the best parameter settings of every STDC-variant for different missing data rates and compares their average RSEs with those of HaLRTC. Overall, STDC outperforms HaLRTC by a significant margin. For example, under the 60 percent missing data rate, the average RSE of HaLRTC is slightly less than 0.46; by contrast, all the STDC variants that use factor priors achieve average RSEs of less than 0.1. The improvement is maintained as the missing data rate increases. Note that all the joint-manifold methods achieve a similar performance, but they have different convergence ranges (STDC-L3 $\approx$ STDC-Ls > STDC-L2), as shown in Fig. 3. These results also demonstrate the advantage of using accurate factor priors and the downsampling techniques. However, there is a tradeoff between the performance and the complexity of our method. Table 3 shows the average number of iterations and the runtimes of STDC and HaLRTC under different missing data rates. Our method is much more time-consuming due to the additional matrix folding/unfolding operations and the mode-$k$ product operations. In the experiments, we found that the optimization of $\mathcal{Z}$ is the most time-consuming step because we use the conjugate

### TABLE 2
### The Average RSEs on Synthetic Data

| missing rate | 60% | 70% | 80% | 90% |
|---|---|---|---|---|
| HaLRTC | 0.4591 | 0.5889 | 0.7922 | 0.9304 |
| STDC-Lx | 0.4377 | 0.5936 | 0.7844 | 0.9260 |
| STDC-L1 | 0.0938 | 0.2881 | 0.6197 | 0.8224 |
| STDC-L2 | **0.0504** | **0.2355** | 0.5135 | 0.7733 |
| STDC-L3 | 0.0597 | 0.2432 | **0.4914** | **0.7513** |
| STDC-Ls | 0.0587 | 0.2405 | 0.5268 | 0.7821 |

### TABLE 3
### The Average Number of Convergence Iterations/Runtimes (in Seconds) on Synthetic Data

| missing rate | 60% | 70% | 80% | 90% |
|---|---|---|---|---|
| HaLRTC | 26.67/0.05 | 27.50/0.05 | 27.33/0.05 | 30.83/0.06 |
| STDC-Lx | 65.83/1.04 | 48.33/0.73 | 57.17/0.94 | 38.00/0.49 |
| STDC-L1 | 100/1.79 | 100/1.86 | 52.83/0.96 | 44.00/0.84 |
| STDC-L2 | 100/1.78 | 98.67/2.02 | 94.00/1.90 | 55.33/1.13 |
| STDC-L3 | 100/89.68 | 100/89.62 | 96.50/86.23 | 84.50/75.67 |
| STDC-Ls | 100/16.58 | 100/16.76 | 89.00/14.98 | 58.17/9.80 |

gradient method. It is well known that CG converges slowly when the number of unknowns (the elements of $\mathcal{Z}$) is very large. To improve the efficiency, we will apply more advanced numerical methods in our future work.

### 4.2 Performance Evaluation on Multilinear Model Analysis

We use the CMU-PIE face database [41] for multilinear model analysis. All the facial images are first aligned by their eye positions and then cropped to size $32 \times 32$. Next, we use a subset selected from the first 30 subjects with 11 poses and 21 illumination changes to construct a fourth-order tensor $\mathcal{X} \in \mathbb{R}^{30 \times 11 \times 21 \times 1024}$. Usually, a complete training set is not available in real applications, as noted in [22]. To handle the missing data problem in face recognition, we assume the observed tensor $\mathcal{X}_0$ is an incomplete training set in which

$$p\% \left( p = \left\{ \begin{array}{l} 40, 50, 60, 70, 80, \\ 90, 92, 94, 96, 98 \end{array} \right\} \right)$$

of the facial images are missing. We use the missing images as a test set for evaluation. Each experiment was performed ten times and the average classification rate was recorded.

We compare STDC with $\mathrm{M}^2\mathrm{SA}$, LRTC, and HaLRTC, as well as with classic methods, including unsupervised methods (PCA [43] and LPP [44]) and supervised methods (LDA [45] and MFA [40]). All of these methods use the nearest neighbor classifier. Note that the classic methods only use incomplete data sets and are not affected by the missing data problem. For the multilinear-based methods, we first complete the tensors to enlarge our data set and then use unsupervised LPP to reduce the dimensionality of features. In addition, as the CMU-PIE database provides the environmental 3D coordinates, we can approximate the relation of two poses or the illumination based on the angle between two camera views. Using the graph construction introduced in Section 3.1.1, we obtain a three-factor Laplacian matrix $\mathbf{L} \in \mathbb{R}^{6930 \times 6930}$ to define our factor priors.

Figs. 5, 6, and 7 show the qualitative and quantitative evaluation results. As shown in Fig. 5, the completion-based methods (LRTC and HaLRTC) and STDC outperform the factorization-based method ($\mathrm{M}^2\mathrm{SA}$) in terms of visual quality. This is because $\mathrm{M}^2\mathrm{SA}$ is designed to estimate the underlying factors instead of the exact values. Although LRTC and HaLRTC can capture the facial structure, they fail to interpret illumination changes. By contrast, STDC successfully characterizes the facial appearance under different poses and lighting variations. In Fig. 6, only
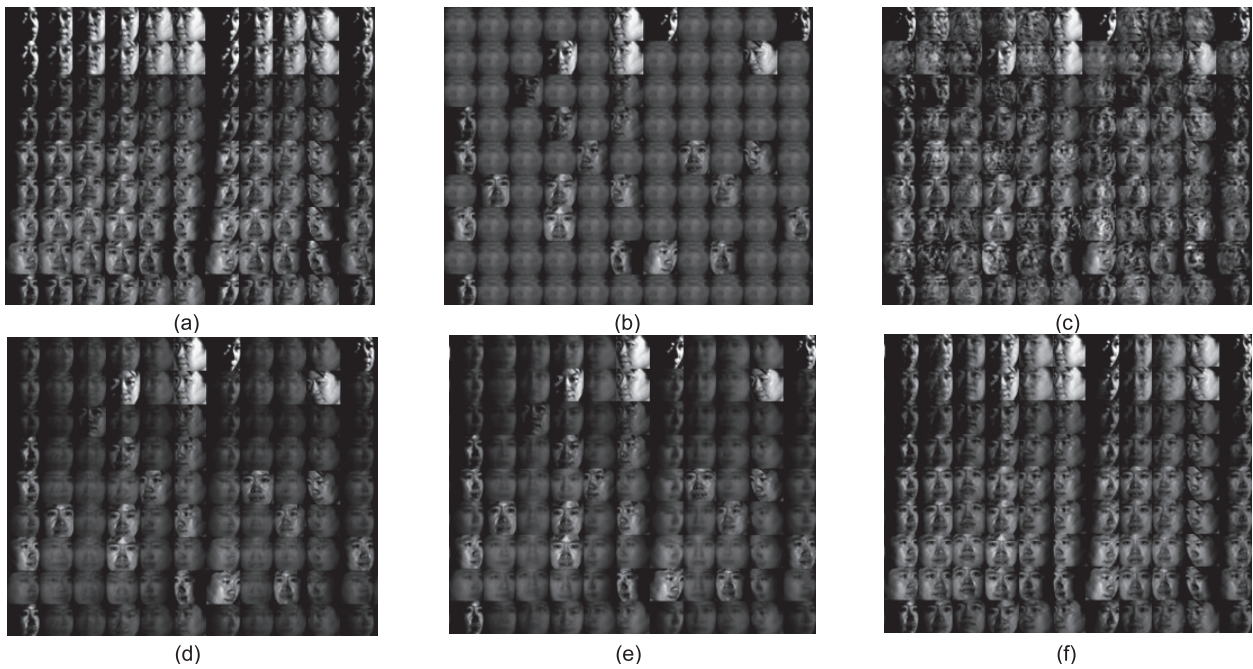
Fig. 5. An example from the CMU-PIE face database (the missing rate = 80 percent): (a) some of the ground truth facial images; (b) the incomplete data in (a), where the missing images are replaced by the mean face of the given images; (c), (d), (e), and (f) show the tensor completion results obtained by M²SA, LRTC, HaLRTC and STDC, respectively.
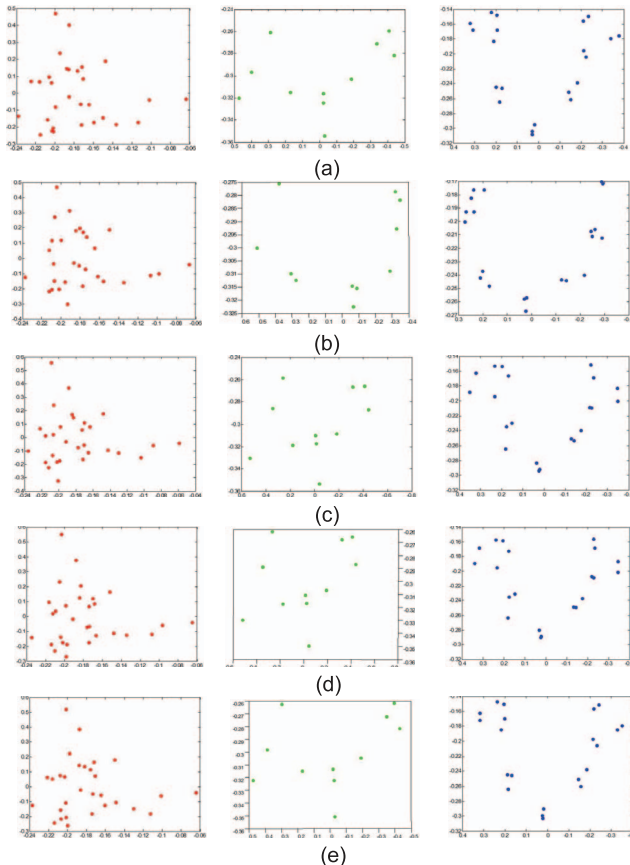


Fig. 6. An example from the CMU-PIE face database (the missing data rate = 80 percent). The submanifolds are obtained by HOSVD with respect to the subject (first column), pose (second column), and illumination (third column): (a) the ground truth; (b)-(e) show the results obtained by M²SA, LRTC, HaLRTC, and STDC, respectively.

$\text{M}^2\text{SA}$ and STDC accurately recover the first two principal dimensions of all the submanifolds (especially for poses) because they consider the intrinsic tensor structure. These results demonstrate the superiority of STDC in terms of tensor recovery and factor estimates. Figs. 7a and 7b show the average RSEs and classification rates, respectively. Among the classic methods, only MFA is comparable to the multilinear-based methods, but it is still inferior to them when the missing data rate is extremely high or extremely low. It is noteworthy that HaLRTC always outperforms LRTC in terms of the RSE, but its classification rate decreases as the missing data rate increases; that is, in the missing data problem, good reconstruction results do not always imply good factor estimates for classification. In contrast, because STDC combines the completion and factorization schemes simultaneously, the corresponding RSEs and classification rates are significantly better than those of the compared methods.

## 4.3 Performance Evaluation on Visual Data

We use the eight images in Fig. 8 as our benchmark data set for image completion. As most works focus on structural images (e.g., Fig. 8a), there is a dearth of research on natural images (e.g., Figs. 8b, 8c, 8d, 8e, 8f, 8g, 8h). Hence, we examine the feasibility of using related techniques for general visual data. In our experiments, we randomly select $p\%$ pixels $(p = \{60, 70, 80, 90, 95\})$ as missing entries and set their values at zero. Then, we apply HaLRTC, $\text{M}^2\text{SA}$, $\text{M}^2\text{SA-G}$, and STDC to recover the missing entries. For factorization-based methods, we try every possible rank to determine the best one. Because of the enormous number of image pixels, we use single-factor priors, also called within-mode regularization terms in [32], as the auxiliary information for $\text{M}^2\text{SA-G}$ and STDC (see the graph construction in Section 3.1.2). To find the best performance of $\text{M}^2\text{SA-G}$, we
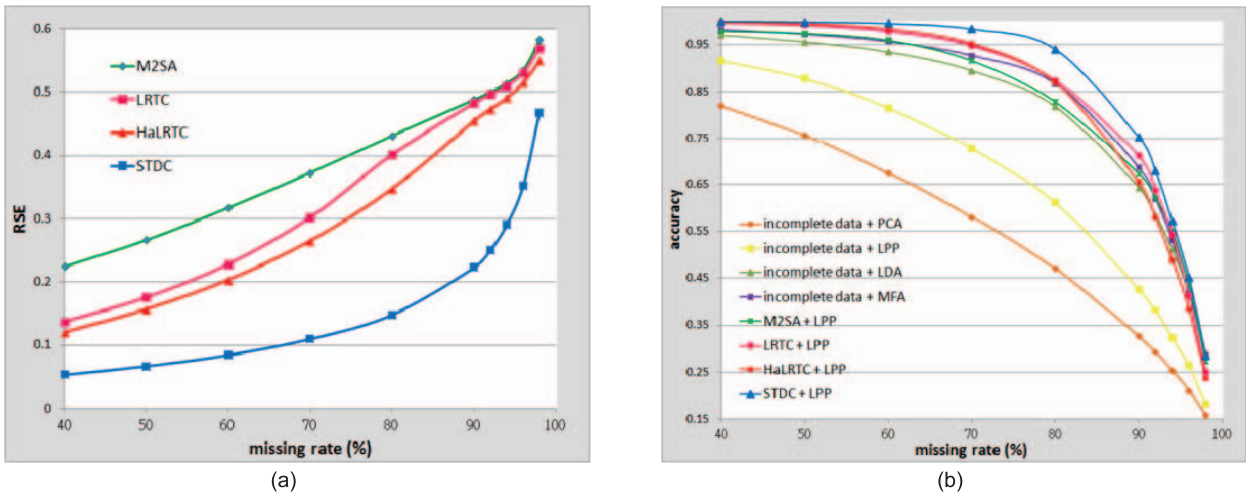
Fig. 7. (a) The average RSE and (b) the average classification rate for images in the CMU-PIE face database.

test the regularization parameters in the range 1-16. Finally, we use two well-known criteria, the PSNR and SSIM [46], to evaluate the performance.

Figs. 9 and 10 show the qualitative and quantitative results. As shown in Fig. 9, the visual quality derived by $M^2SA$ is poor, even though the ranks are fine-tuned. HaLRTC only recovers missing data accurately when the rate is less than 70 percent; otherwise, its performance is worse than that of $M^2SA$-G. Because of the advantages of graph regularization, $M^2SA$-G improves the performance of $M^2SA$ significantly if the best parameters are chosen. Meanwhile, STDC always achieves the best recovery in terms of visual quality, especially in cases with a high missing data rate. In addition, because the proposed factor priors can be regarded as regularization terms for the local similarity, we also compare STDC with the total variation (TV) technique, which is widely used in image processing. To ensure a fair comparison, we combine the fast TV method [47] (also solved by ALM) with HaLRTC. As shown in Fig. 9, our results are highly comparable to those of HaLRTC+TV. In fact, at high missing data rates, the images recovered by STDC are less blurry and contain more details. The accuracy is due to the precise factor estimates, which, instead of only using neighboring similarity on missing entries, closely relate to the global latent structure of the Tucker model. In Fig. 10, the average PSNR and SSIM results demonstrate the superiority of STDC over existing approaches. To sum up, STDC simultaneously characterizes 1) the factor relation defined in *factor priors*, 2) the underlying structure of the *factorization scheme*, and 3) the low-rank property in the *completion scheme*. Hence, we reduce the performance gap significantly in tensor completion under existing algorithms.

To further investigate the parameters' sensitivity on real-world data, we evaluated their performance under different settings of $\omega$, as shown in Fig. 11. From Figs. 3 and 4, it is clear that smaller $\kappa$ values yield strong convergence rates and good RSEs when we only use single-factor priors. Therefore, we fix $\kappa = 10^{0.2}$ and examine $\omega$ in the range of $[10^{-1}, 10^1]$. As shown in Fig. 11, our algorithm is insensitive to $\omega$ when the missing data rate is less than 90 percent. If the observations are too sparse, larger $\omega$ can prevent overfitting of the Tucker model and yield good PSNR/SSIM results. However, the parameter-setting problem is not a critical concern in tensor completion. As the missing data rate is usually given, we can easily fine-tune our parameters to maximize the advantages of STDC.

Finally we use video inpainting to demonstrate the potential of STDC. Although the low-rank property can characterize random missing entries, the structured missing entry problem severely degrades the performance of tensor completion [21], [32]. In our experiments, we use a dynamic wave sequence as a fourth-order tensor and compare HaLRTC with two variants of STDC. STDC-1 considers three single-factor priors in the X (row), Y (column), and T (time) domains; while STDC-2 considers two joint-factor priors in the X-T and Y-T domains. Fig. 12 shows the results of dynamic background inpainting. All the methods have similar RSEs and converge at around 0.1. However, STDC yields better visual results, while HaLRTC produces obvious artifacts. In addition, because STDC-2 considers the joint-factor changes, it outperforms STDC-1 in terms of visually consistent textures. Fig. 13 shows another example of repairing an old film.
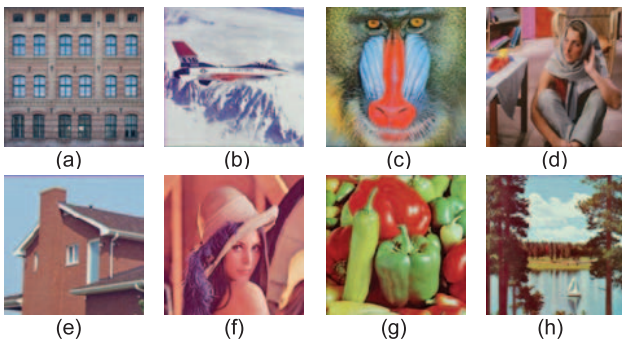
## 5 CONCLUDING REMARKS

In this paper, we incorporate factor priors into a novel tensor completion method called simultaneous tensor



Fig. 8. Eight benchmark images. (a) Façade, (b) airplane, (c) baboon, (d) barbara, (e) house, (f) lena, (g) peppers, and (h) sailboat.

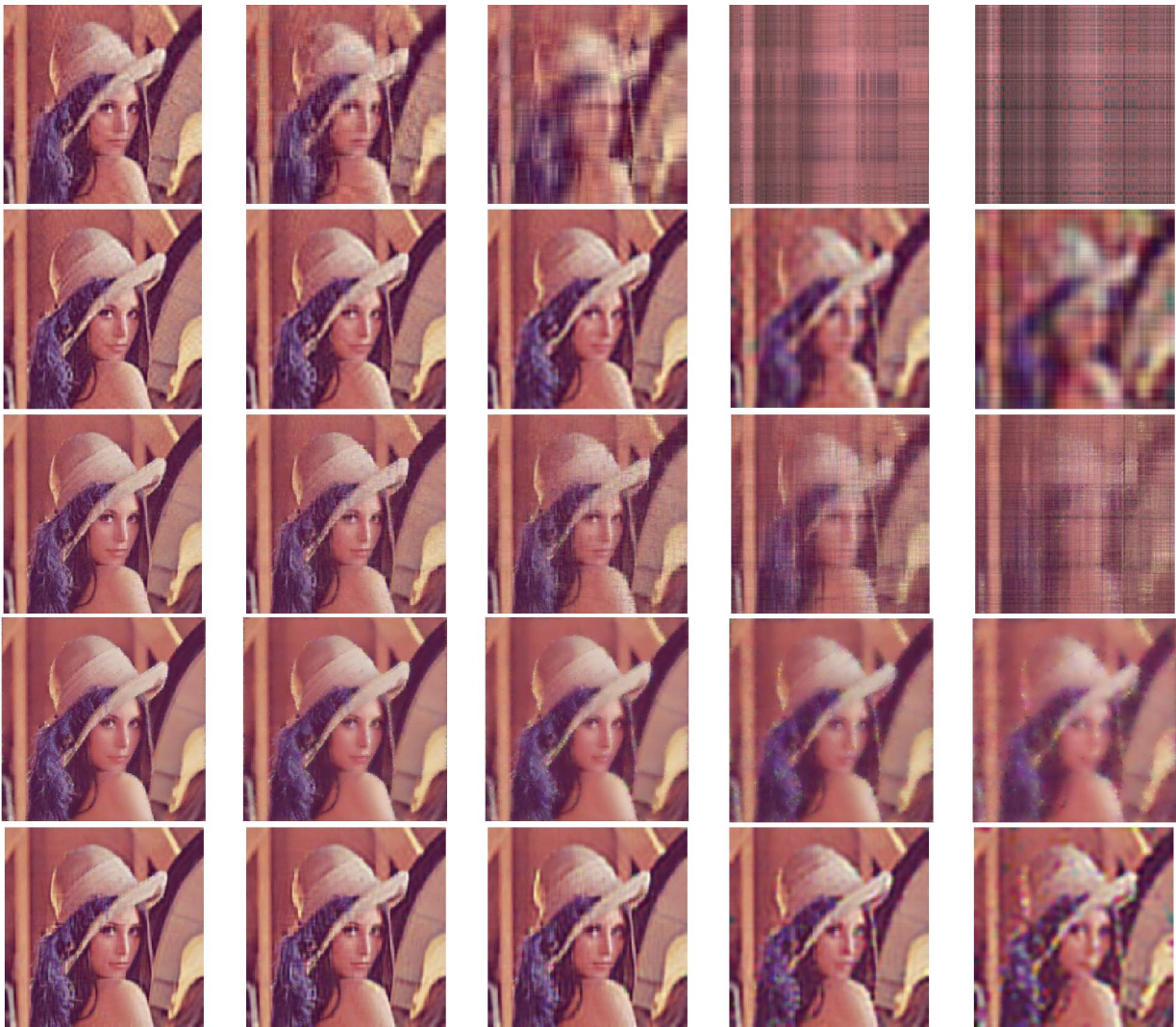Fig. 9. The tensor completion results for the Lena image obtained by M²SA, M²SA-G, HaLRTC, HaLRTC+TV, and STDC (from top to bottom) under missing data rates of 60, 70, 80, 90, and 95 percent (from left to right).

decomposition and completion, which completes missing data entries while exploiting the factorization structure. The contribution of this work is twofold. First, STDC is the first approach that bridges the gap between factorization schemes and completion schemes and, thereby, simultaneously guarantees accurate estimates of missing data and latent factors. Second, the proposed factor priors, which benefit from the generality of MGE, unify and characterize the underlying joint-manifold in a tensor object, even without knowing the exact entries. Because of these characteristics, STDC does not suffer from the limitations of existing tensor completion algorithms, e.g., the fixed-rank assumption and an oversimplified model.

The results of experiments on synthetic and real-world data show that the proposed method outperforms state-of-the-art methods. The low RSEs in general cases verify the superiority of STDC over existing approaches, while the high recognition rate demonstrates the precision of the factor estimates.
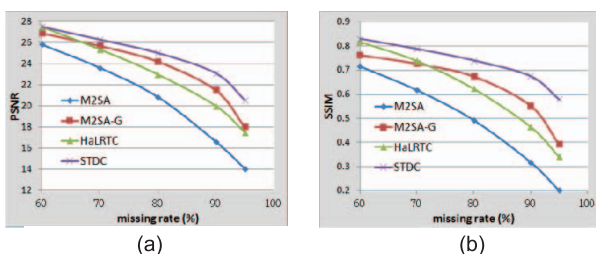


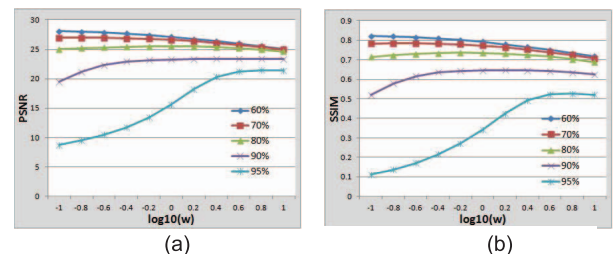Fig. 10. (a) The average PSNRs and (b) the average SSIMs of the eight benchmark images.



Fig. 11. (a) The average PSNRs and (b) the average SSIMs versus $\log_{10}(\omega)$ under different missing rates.
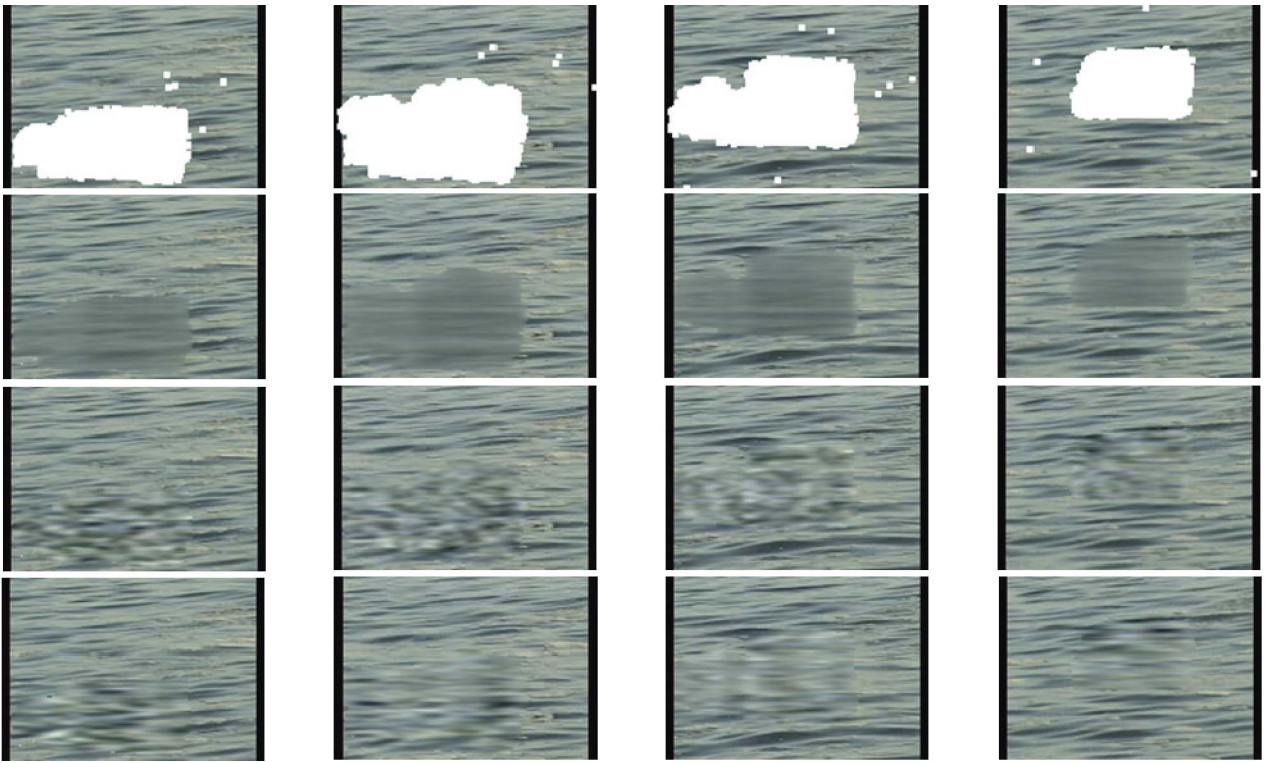
Fig. 12. The 1st, 9th, 17th, and 25th frames (from left to right) in the "Wave" sequence, and the tensor completion results obtained by HaLRTC, STDC-1, and STDC-2 (from top to bottom).



Fig. 13. The 7th, 25th, 31st, and 33rd frames (from left to right) in an old film, and the tensor completion results derived by HaLRTC and STDC (from top to bottom).

## REFERENCES

[1] T. Ding, M. Sznaier, and O.I. Camps, "A Rank Minimization Approach to Video Inpainting," *Proc. 11th IEEE Int'l Conf. Computer Vision (ICCV)*, pp. 1-8, 2007.

[2] Y.X. Wang and Y.J. Zhang, "Image Inpainting via Weighted Sparse Non-Negative Matrix Factorization," *Proc. 18th IEEE Int'l Conf. Image Processing (ICIP)*, pp. 3409-3412, 2011.

[3] H. Ji, C. Liu, Z. Shen, and Y. Xu, "Robust Video Denoising Using Low Rank Matrix Completion," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 1791-1798, 2010.

[4] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust Alignment by Sparse and Low-Rank Decomposition for Linearly Correlated Images," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 763-770, 2010.

[5] Z. Zhang, A. Ganesh, X. Liang, and Y. Ma, "TILT: Transform Invariant Low-Rank Textures," *Int'l J. Computer Vision,* vol. 99, no. 1, pp. 1-24, 2012.

[6] C. Lang, G. Liu, J. Yu, and S. Yan, "Saliency Detection by Multitask Sparsity Pursuit," *IEEE Trans. Image Processing,* vol. 21, no. 3, pp. 1327-1338, Mar. 2012.

[7] L. Zhuang, H. Gao, J. Huang, and N. Yu, "Semi-Supervised Classification via Low-Rank Graph," *Proc. Sixth Int'l Conf. Image and Graphics,* pp. 511-516, 2011.

[8] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust Recovery of Subspace Structures by Low-Rank Representation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 35, no. 1, pp. 171-184, Jan. 2013.

[9] K. Li, Q. Dai, W. Xu, and J. Jiang, "Three-Dimensional Motion Estimation via Matrix Completion," *IEEE Trans. Systems, Man, and Cybernetics—Part B: Cybernetics,* vol. 42, no. 2, pp. 539-551, Apr. 2012.

[10] X. Zhou, C. Yang, and W. Yu, "Moving Object Detection by Detecting Contiguous Outliers in the Low-Rank Representation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 35, no. 3, pp. 597-610, Mar. 2013.

[11] O. Oreifej, X. Lin, and M. Shah, "Simultaneous Video Stabilization and Moving Object Detection in Turbulence," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 35, no. 2, pp. 450-462, Feb. 2013.

[12] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification.* Wiley Interscience, 2000.

[13] D.D. Lee and H.S. Seung, "Learning the Parts of Objects by Nonnegative Matrix Factorization," *Nature,* vol. 401, pp. 788-791, 1999.

[14] S. Ma, D. Goldfarb, and L. Chen, "Fixed Point and Bregman Iterative Methods for Matrix Rank Minimization," *Math. Programming,* vol. 128, no. 1, pp. 321-353, 2009.

[15] J.F. Cai, E.J. Candes, and Z. Shen, "A Singular Value Algorithm for Matrix Completion," *SIAM J. Optimization,* vol. 20, no. 4, pp. 1956-1982, 2010.

[16] X. Lu, T. Gong, P. Yan, Y. Yuan, and X. Li, "Robust Alternative Minimization for Matrix Completion," *IEEE Trans. Systems, Man, and Cybernetics—Part B: Cybernetics,* vol. 42, no. 3, pp. 939-949, June 2012.

[17] B. Recht, M. Fazel, and P. Parrilo, "Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization," *SIAM Rev.,* vol. 52, no. 3, pp. 471-501, 2010.

[18] W.J. Li and D.Y. Yeung, "Relation Regularized Matrix Factorization," *Proc. 21st Int'l Joint Conf. Artificial Intelligence,* pp. 1126-1131, 2009.

[19] A.D. Bue, J. Xavier, L. Agapito, and M. Paladini, "Bilinear Modeling via Augmented Lagrange Multipliers (BALM)," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 34, no. 8, pp. 1496-1508, Aug. 2012.

[20] J. Chen and Y. Saad, "On the Tensor SVD and the Optimal Low Rank Orthogonal Approximation of Tensors," *SIAM J. Matrix Analysis and Applications,* vol. 30, no. 4, pp. 1709-1734, 2009.

[21] E. Acar, D.M. Dunlavy, T.G. Kolda, and M. Morup, "Scalable Tensor Factorization for Incomplete Data," *Chemometrics and Intelligent Laboratory Systems,* vol. 106, pp. 41-56, 2011.

[22] X. Geng, K. Smith-Miles, Z.H. Zhou, and L. Wang, "Face Image Modeling by Multilinear Subspace Analysis with Missing Values," *IEEE Trans. Systems, Man, and Cybernetics—Part B: Cybernetics,* vol. 41, no. 3, pp. 881-892, June 2011.

[23] J. Liu, P. Wonka, and J. Ye, "Tensor Completion for Estimating Missing Values in Visual Data," *Proc. 12th IEEE Int'l Conf. Computer Vision (ICCV),* 2009.

[24] Y. Li, J. Yan, Y. Zhou, and J. Yang, "Optimum Subspace Learning and Error Correction for Tensors," *Proc. 11th European Conf. Computer Vision (ECCV),* 2010.

[25] S. Gandy, B. Recht, and I. Yamada, "Tensor Completion and Low-N-Rank Tensor Recovery via Convex Optimization," *Inverse Problems,* vol. 27, no. 2, 2011.

[26] R. Tomioka, K. Hayashi, and H. Kashima, "Estimation of Low-Rank Tensors via Convex Optimization," *Arxiv Preprint ar-Xiv:1010.0789, 2011.

[27] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor Completion for Estimating Missing Values in Visual Data," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 35, no.1, pp. 208-220, Jan. 2013.

[28] M. Signoretto, Q.T. Dinh, L.D. Lathauwer, and J.A.K. Suykens, "Learning with Tensors: A Framework Based on Convex Optimization and Spectral Regularization," *Machine Learning,* May 2013.

[29] L.D. Lathauwer, B.D. Moor, and J. Vandewalle, "A Multilinear Singular Value Decomposition," *SIAM J. Matrix Analysis and Applications,* vol. 21, no. 4, pp. 1254-1278, 2000.

[30] R.P. Adams, G.E. Dahl, and I. Murray, "Incorporating Side Information in Probabilistic Matrix Factorization with Gaussian Process," *Proc. 26th Conf. Uncertainty in Artificial Intelligence,* pp. 1-9, 2010.

[31] R. Salakhutdinov and A. Mnih, "Probabilistic Matrix Factorization," *Proc. Advances in Neural Information Processing Systems (NIPS),* pp. 1257-1264, 2008.

[32] A. Narita, K. Hayashi, R. Tomioka, and H. Kashima, "Tensor Factorization Using Auxiliary Information," *Proc. European Conf. Machine Learning and Knowledge Discovery,* pp. 501-516, 2011.

[33] Y.L. Chen and C.T. Hsu, "Multilinear Graph Embedding: Representation and Regularization for Images," *IEEE Trans. Image Processing,* to appear.

[34] N. Gillis and F. Glineur, "Low-Rank Matrix Approximation with Weights or Missing Data Is NP Hard," *SIAM J. Matrix Analysis Applications,* vol. 32, no. 4, pp. 1149-1165, 2011.

[35] D. Bertsekas, *Constrained Optimization and Lagrange Multiplier Method.* Academic Press, 1982.

[36] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices," Technical Report UILU-ENG-09-2215, Univ. of Illinois at Urbana-Champaign, 2009.

[37] B. He, M. Tao, and X. Yuan, "Alternating Direction Method with Gaussian Back Substitution for Separable Convex Programming," *SIAM J. Optimization,* vol. 22, no. 2, pp. 313-340, 2012.

[38] Y. Shen, Z. Wen, and Y. Zhang, "Augmented Lagrangian Alternating Direction Method for Matrix Separation Based on Low-Rank Factorization," technical report, Rice Univ., 2011.

[39] G. Liu and S. Yan, "Active Subspace: Towards Scalable Low-Rank Learning," *Neural Computing,* vol. 24, no. 12, 2012.

[40] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin, "Graph Embedding and Extensions: A General Framework for Dimensionality Reduction," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 1, pp. 40-51, Jan. 2007.

[41] T. Sim, S. Baker, and M. Bsat, "The CMU Pose, Illumination, and Expression Database," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 12, pp. 1615-1618, Dec. 2003.

[42] J. Yang and X. Yuan, "Linearized Augmented Lagrangian and Alternating Direction Methods for Nuclear Norm Minimization," *Math. Computation,* vol. 82, pp. 301-329, 2013.

[43] M. Turk and A. Pentland, "Face Recognition Using Eigenfaces," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* pp. 586-591, 1991.

[44] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang, "Face Recognition Using Laplacianfaces," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 3 , pp. 328-340, Mar. 2005.

[45] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, pp. 711-720, July 1997.

[46] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Trans. Image Processing,* vol. 13, no. 4, pp. 600-612, Apr. 2004.

[47] S.H. Chan, R. Khoshabeh, K.B. Gibson, P.E. Gill, and T.Q. Nguyen, "An Augmented Lagrangian Method for Total Variation Video Restoration," *IEEE Trans. Image Processing,* vol. 20, no. 11, pp. 3097-3111, Nov. 2011.

**Yi-Lei Chen** received the BS degree in computer science in 2007 and the MS degree in computer science in 2009, both from National Tsing Hua University, Hsinchu, Taiwan. He is currently working toward the PhD degree in the Department of Computer Science, National Tsing Hua University. His research interests include machine learning, computer vision, and multilinear representation. He is a student member of the IEEE.

**Chiou-Ting Hsu** received the PhD degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 1997. From 1998 to 1999, she was with Philips Innovation Center Taipei, Philips Research, as a senior research engineer. She joined the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan, as an assistant professor in 1999 and is currently a professor. Her research interests include multimedia signal processing, video analysis, and content-based retrieval. She received the Citation Classic Award from Thomson ISI in 2001 for her paper "Hidden Digital Watermarks in Images." She was an associate editor of *Advances in Multimedia* during 2006-2012, and is currently an associate editor of the *IEEE Transactions on Information Forensics and Security*. She is a senior member of the IEEE.

**Hong-Yuan Mark Liao** received the PhD degree in electrical engineering from Northwestern University in 1990. In July 1991, he joined the Institute of Information Science, Academia Sinica, Taiwan, where he is currently a distinguished research fellow. He is jointly appointed as a professor in the Computer Science and Information Engineering Department of National Chiao-Tung University. During 2009-2012, he was jointly appointed as the Multimedia Information Chair Professor at National Chung Hsing University. Since August 2010, he has been an adjunct chair professor of Chung Yuan Christian University. He received the Young Investigators' Award from Academia Sinica in 1998, the Distinguished Research Award from the National Science Council of Taiwan in 2003 and 2010, the National Invention Award of Taiwan in 2004, the Distinguished Scholar Research Project Award from National Science Council of Taiwan in 2008, and the Academia Sinica Investigator Award in 2010. His professional activities include: cochair, 2004 International Conference on Multimedia and Exposition (ICME), technical cochair, 2007 ICME, Editorial Board member, *IEEE Signal Processing Magazine* (2010-present), associate editor of the *IEEE Transactions on Image Processing* (2009-present), *IEEE Transactions on Information Forensics and Security* (2009-2012), and *IEEE Transactions on Multimedia* (1998-2001). He has been a fellow of the IEEE since 2013.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.