Smooth PARAFAC Decomposition for Tensor Completion

Tatsuya Yokota, Member, IEEE, Qibin Zhao, Member, IEEE, Chao Li, and Andrzej Cichocki, Fellow, IEEE

Abstract—Low-rank based tensor completion, which is a higher-order extension of matrix completion, has received considerable attention in recent years. However, low-rank assumption is not sufficient for recovering visual data, such as color and 3D images, under extremely high missing ratio. In this paper, we consider ‘smoothness’ constraints as well as low-rank approximations, and propose an efficient algorithm to perform tensor completion, which is particularly powerful for visual data. The proposed method gains the significant advantages due to the integration of smooth PARAFAC decomposition (PD) for an incomplete tensor and efficient model selection for minimizing tensor rank, which is thus termed as “Smooth Parafac tensor Completion” (SPC). To impose the smoothness constraints, we employed two strategies including total variation (SPC-TV) and squared variation (SPC-SV), and provide the corresponding algorithms for model learning. The extensive experimental evaluations on both synthetic and real-world visual data illustrated the significant improvements, in terms of prediction performance and efficiency, as compared with many state-of-the-art tensor completion methods.

Index Terms—Tensor completion for images, smoothness, low-rank tensor approximation, CP model, PARAFAC model, total variation (TV), squared variation.

1 INTRODUCTION

Completion is a procedure which allows to estimate values of missing elements of array data by using only available elements and structural properties of data. Of course, if there is no relationship between missing elements and available elements, completion is impossible. However, real world data usually have some correlations, latent factors, symmetry, continuity or repetition, then the completion is possible in many cases. For example, when a vector consists of the sampled values of some continuous function that has several missing values, some interpolation methods can be used for completion such as a simple linear interpolation, a spline interpolation and a polynomial interpolation. When a given matrix has several missing entries and the low-rank matrix factorization exists, then a low-rank structure can be used for completion by approximating given matrix as the low-rank factorization model. Such completion techniques are closely related to computer vision, pattern recognition, and compressed sensing [13], [17], [26].

Techniques for vector/matrix completion are well researched and many sophisticated methods exist. Furthermore, the techniques of ‘tensor’ completion have attracted attention in recent years because of their potential applications and flexibility. Tensor is a multi-dimensional array, and a vector and a matrix can be considered as a 1st-order tensor and a 2nd-order tensor, respectively. For example, a color-image data is a 3rd-order tensor because it consists of three color shading images of red, green and blue. Similarly, a color video data is a 4th-order tensor because it consists of multiple frames of color images. There are several papers which tried to complete color images by using matrix completion [22], [20], [25]. In [20], the authors proposed to complete each color shading image separately, and concatenated them finally. However, such approach would ignore the multi-way natural structure of tensor and neglect some important information. In [22], [25], the authors applied the matrix completion for matrices which consist of set of similar patches created by some patch matching algorithms, however generally the patch matching is time consuming and it does not work well under extremely high missing ratio.

On the other hand, the tensor completion methods exhibit a large progress for the completion of color images by exploiting the structural information of 3rd-order tensor. There are two types of approaches as the state-of-the-art methods for the tensor completion. First, the nuclear norm minimization approach is proposed based on the low-rank property [30], [16], [31]. The nuclear norm of matrix is defined as a sum of all singular values and it is a convex envelope for the rank of matrix [33]. Then, the nuclear norm of the tensor was firstly introduced, and applied to the tensor completion in [30]. Moreover, some improved algorithms were proposed in [16], [31]. The minimization of tensor nuclear norm was also applied in various applications such as error correction [27], [28], medical imaging [32], compression [24], and saliency detection [44]. Second, as an alternative approach low-rank tensor decomposition techniques were proposed in [11], [39], [23], [9], [49]. Tensor factorization is a method to decompose an Nth-order tensor into an Nth-order
tensor with smaller size, termed as a ‘core tensor’, and \( N \) factor matrices. There are two factorization models: Tucker decomposition \([6, 21]\) and PARAFAC decomposition \([6, 21]\). A core tensor of PARAFAC decomposition is given as a super-diagonal tensor, and a core tensor of Tucker decomposition is given as a general tensor, thus the PARAFAC decomposition is a special case of the Tucker decomposition. Similar to a singular value decomposition (SVD) of matrix factorization, the minimum size of core tensor corresponds to the rank of tensor. In \([9]\), authors consider the low rank Tucker decomposition model for the tensor completion and minimizing nuclear norm of individual factor matrices. In \([49]\), authors consider the low rank PARAFAC decomposition (PD) model for the tensor completion by using Bayesian framework to solve tuning parameter problems. Moreover, both papers \([9, 49]\) introduced the smoothness constraint as a factor prior, and yielding improved results for visual data.

In this paper, we investigate the smoothness constraints for the PD model based tensor completion, and propose a new strategy for rank determination algorithm. Smoothness is an important property embedded in many real world data, e.g., natural images/videos, some spectral signals, and biomedical data. In fact, matrix/tensor factorizations with smoothness constraints have many applications which are robust to noisy signals such as for blind source separation \([47, 46, 48]\), video structuring \([15]\), visual parts extraction \([45]\), genomics data analysis \([43]\), and brain signal analysis \([10]\). For estimating missing values of incomplete smooth signals, the smoothness constraint can play a key role to achieve high accuracy. Total variation (TV) \([37]\) is often used to impose piece-wise smoothness constraint which is defined as the \( L_1 \)-norm of the difference of neighbor elements. In general, natural images and additive noise can be assumed as smooth and non-smooth signals, respectively. Thus, the smoothness constraint will save the smooth natural images and remove non-smooth noise. There are many methods which applied the TV approach in image restoration and denoising \([34, 19, 42, 4, 33, 7]\). Furthermore, the TV constraint has been already applied for the matrix completion in \([14, 20]\), but till now not investigated for tensors. Thus, we apply the TV constraint into the PD model based tensor completion. Furthermore, we investigate another smoothness constraint which is defined as the ‘\(l_2\)-norm’ of the difference of neighbor elements, which is a stronger constraint than the TV constraint.

A general strategy of above mentioned “state-of-the-arts” algorithms is setting the upper-bound of the rank of tensor and optimizes the rank of tensor by some procedures to remove redundant components such as the singular value shrinkage. However, if we impose the smoothness constraint number of components will increase and we can not able to determine its upper-bound. Therefore, we propose an opposite strategy for estimating the rank of tensor (i.e., the optimal number of components) by increasing the rank of tensor step by step, starting with a rank-one tensor, and our algorithm will stop when the smooth PD model fit the observed data sufficiently well. We conducted the experiments for tensor completion problems by using several difficult benchmarks of color images with different types of missing pixels (e.g., text masked, scratched images, and random pixels missing etc), incomplete MRI data, and multi-way structured facial data.

The remainder of this paper is organized as follows. In Section 1.1, we explain the notations in this paper. Section 2 reviews the several existing matrix/tensor completion methods. In Section 3, we propose novel algorithms for the smooth PD model based tensor completion. In Section 4, we investigate the performance and applications of our algorithms, and compare them with some state-of-the-art methods. In Section 5, we discuss several aspects of our works. Finally, we give our conclusions in Section 6.

### 1.1 Preliminaries and Notations

A vector is denoted by bold small letter \( \mathbf{a} \in \mathbb{R}^I \). A matrix is denoted by bold capital letter \( \mathbf{A} \in \mathbb{R}^{I \times J} \). A higher-order (\( N \geq 3 \)) tensor is denoted by bold calligraphic letter \( \mathbf{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \). The \( i \)th entry of vector \( \mathbf{a} \in \mathbb{R}^I \) is denoted by \( a(i) \), and \( (i,j) \)th entry of matrix \( \mathbf{A} \in \mathbb{R}^{I \times J} \) is denoted by \( A(i,j) \). The \((i_1, i_2, \ldots, i_N)\)th entry of \( N \)-th order tensor \( \mathbf{A} \) is denoted by \( \mathbf{A}_{i_1, i_2, \ldots, i_N} \), where \( i_n \in \{1, 2, \ldots, I_n\}, n \in \{1, 2, \ldots, N\} \).

Frobenius norm of \( N \)-th order tensor is defined by \( \| \mathbf{X} \|_F := \sqrt{\sum_{i_1, i_2, \ldots, i_N} \mathbf{X}^2_{i_1, i_2, \ldots, i_N}} \).

A mode-\( k \) unfolding of tensor \( \mathbf{X} \) is denoted as \( \mathbf{X}_{(k)} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}_{i_k \neq k} \). For example, a first mode unfolding of 3rd-order tensor \( \mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3} \) is given by \( \mathbf{X}_{(1)}(i_1, (i_3 - 1)I_2 + i_2) = \mathbf{X}_{i_1, i_3, i_2} \), where \( i_n \in \{1, 2, \ldots, I_n\}, n \in \{1, 2, 3\} \). A mode-\( k \) multiplication between a tensor \( \mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \) and a matrix/vector \( \mathbf{A} \in \mathbb{R}^{I_k \times R} \) is denoted by \( \mathbf{Y} = \mathbf{X} \times_k \mathbf{A}^T \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{k-1} \times R \times I_{k+1} \times \cdots \times I_N} \), with entries given by \( \mathbf{Y}_{i_1, \ldots, i_{k-1}, r, i_{k+1}, \ldots, i_N} = \sum_{i_k} \mathbf{X}_{i_1, \ldots, i_{k-1}, i_k, i_{k+1}, \ldots, i_N} A(i_k, r) \), and we have \( \mathbf{Y}_{(k)} = \mathbf{A}^T \mathbf{X}_{(k)} \).

### 2 Brief Review of Existing Methods

In this section, we explain the methodology for the matrix/tensor completion based on the low-rank assumption and review the several state-of-the-art methods.

For the matrix data, minimization of matrix rank subject to fit available elements is the most popular and basic approach for the matrix completion:

\[
\text{minimize} \ \text{rank}(\mathbf{X}), \ \text{s.t.} \ \mathbf{X}_\Omega = \mathbf{T}_\Omega, \quad (1)
\]
where $X$ is an output completed matrix, $T$ is an input incomplete matrix, $\Omega$ stands for indexes of available elements of $T$, an equation $X_\Omega = T_\Omega$ means that $X(i,j) = T(i,j)$, $\forall (i,j) \in \Omega$, rank($X$) denotes the rank of matrix $X$. Such a rank minimization approach was applied in various research fields such as machine learning [2], [3] and bioinformatics [40]. If an original matrix is structured as a low-rank matrix, this rank minimization approach can be used to obtain some estimation of the ground truth matrix. However, rank is not convex function with respect to $X$, and rank minimization is generally NP-hard [18]. Therefore, the nuclear-norm minimization is widely used in practice:

$$\min_X \|X\|_* \text{ s.t. } X_\Omega = T_\Omega,$$

(2)

where $\|X\|_* = \sum_i \sigma_i(X)$ denotes the nuclear norm, and $\sigma_i(X)$ denotes the $i$th largest singular value of matrix $X$. To solve the optimization problem (2), the following augmented Lagrange function can be applied:

$$L_A(X,Y,W,\beta) = \|X\|_* - \langle W, X - Y \rangle + \frac{\beta}{2} \|X - Y\|_F^2,$$

(3)

where $Y_\Omega = T_\Omega$ is an auxiliary matrix, $W$ is a Lagrangian multiplier matrix, $\beta$ is a penalty (trade-off) parameter for the augmented term $\|X - Y\|_F^2$, and $\beta$ is increased in each iteration step. The problem can be optimized by applying the alternating direction method of multipliers (ADMM) [5]. The update rules, which were proposed as inexact augmented Lagrangian multiplier (IALM) method [29], are given by

$$Y^{(k+1)} \leftarrow X^{(k)} - \frac{1}{\beta} W^{(k)};$$

(4)

$$Y^{(k+1)}_\Omega \leftarrow T_\Omega;$$

(5)

$$X^{(k+1)} \leftarrow D_{\frac{1}{\beta}} \left( Y^{(k+1)} + \frac{1}{\beta} W^{(k)} \right);$$

(6)

$$W^{(k+1)} \leftarrow W^{(k)} - \beta (X^{(k+1)} - Y^{(k+1)});$$

(7)

$$\beta \leftarrow c\beta;$$

(8)

where $c > 1$ is a parameter of increasing rate of $\beta$, and $D_{\frac{1}{\beta}}(\cdot)$ is an operator of singular value shrinkage which is defined by $D_{\frac{1}{\beta}}(x) := \sum_i \max(\sigma_i(x) - \tau, 0) u_i v_i^T$, where $u_i$ and $v_i$ are left and right singular vectors corresponding to $\sigma_i(X)$. There are several other papers which applied the ADMM scheme to matrix or tensor completion problems [16], [8], [31].

The smoothness property of data is often assumed for the image completion. If we consider $X$ is a gray-scale image and individual entries stand for values of individual pixels, the difference between neighbor pixels should be typically small. In [37], the minimization of total variation (TV) was employed for smoothness constraint which was defined by

$$\|X\|_{TV} := \sum_{i,j} \sqrt{X_v(i,j)^2 + X_h(i,j)^2},$$

(9)

where $X_v(i,j) := X(i+1,j) - X(i,j)$ and $X_h(i,j) := X(i,j+1) - X(i,j)$. The image completion via TV minimization was proposed in [14], and formulated as the following optimization problem:

$$\min_X \|X\|_{TV}, \text{ s.t. } X_\Omega = T_\Omega.$$

This problem is convex and can be solved by the gradient descent methods [14].

Furthermore, a combination of the low-rank approximation and smoothness constraints was proposed in [20]. Usually, natural images have both structural features and this combination is quite efficient. The most straightforward approach for the combination of low-rank approximation and smoothness constraints was implemented by the following optimization problem:

$$\min_X \|X\|_* + \gamma \|X\|_{TV}, \text{ s.t. } X_\Omega = T_\Omega,$$

(10)

where $\gamma$ is a trade-off parameter between a nuclear norm minimization and TV minimization. In [20], a modified linear total variation was defined as

$$\|X\|_{LTV} := \sum_{i,j} \{X_v(i,j)^2 + X_h(i,j)^2\},$$

(11)

and a smooth low-rank matrix completion method was proposed by

$$\min_X \|X\|_* + \gamma \|X\|_{LTV}, \text{ s.t. } X_\Omega = T_\Omega.$$  

(12)

The optimization problem (12) was referred to as the linear total variation approximate regularized nuclear norm (LTVNN) minimization problem, and this problem can be solved by ADMM like optimization scheme. This approach can be considered as the state-of-the-art method for low-rank image completion, however the LTVNN may be not useful for tensor completion. For a color image (3rd-order tensor), the authors separated a color-image into red, green, and blue frames, and applied the completion method to each color frame, separately. However, when individual color frames are quite similar, such a separated method may not be effective.

Tensor completion is a natural extension of the matrix completion with respect to data structure, and it can use such structural information more effectively than matrix completion. Basic method for tensor completion was proposed as simple low-rank tensor completion (SiLRTC) [31], and then it was formulated as the following constrained optimization problem:

$$\min_{X, Y^{(1)}, \ldots, Y^{(N)}} \sum_{i=1}^N \left\{ \alpha_i \|Y^{(i)}\|_* + \frac{\beta}{2} \|X^{(i)} - Y^{(i)}\|_F^2 \right\},$$

(13)

s.t. $X_\Omega = T_\Omega$. 

\[ \min_{X, Y^{(1)}, \ldots, Y^{(N)}} \sum_{i=1}^N \left\{ \alpha_i \|Y^{(i)}\|_* + \frac{\beta}{2} \|X^{(i)} - Y^{(i)}\|_F^2 \right\}, \]

\[ \text{s.t. } X_\Omega = T_\Omega, \]
where $\mathbf{X}$ is an output completed tensor, $\mathbf{T}$ is an input incomplete tensor, $Y^{(i)}$ is a low-rank matrix corresponding to mode-$i$ matricization form $X_{(i)}$, $\alpha_i$ and $\beta$ are weight parameters for individual cost functions. Essentially, this method attempts to minimize the tensor nuclear norm defined as $\sum_i \alpha_i \|X_{(i)}\|_*$, which is an generalization of matrix nuclear norm. Instead of the minimization of $\sum_i \alpha_i \|X_{(i)}\|_*$, we minimize the nuclear norm of alternate parameters $Y^{(i)}$ and mean squared error between alternate parameters $Y^{(i)}$ and mode-$i$ unfolding of $\mathbf{X}$.

Note that when we strictly consider to minimize $\sum_i \alpha_i \|X_{(i)}\|_*$, constraints $X_{(i)} = Y^{(i)}$ should be added. High accuracy low rank tensor completion (HaLRTC) [31] was proposed based on this concept by formulating the following optimization problem:

\[
\begin{align*}
\text{minimize} & \quad \mathcal{X}, Y^{(1)}, \ldots, Y^{(N)} \sum_{i=1}^{N} \alpha_i \|Y^{(i)}\|_*, \\
\text{s.t.} & \quad \mathcal{X} = Y^{(i)}, \quad \mathcal{T}_\Omega = \mathbf{T}_\Omega,
\end{align*}
\]

where $Y^{(i)}$ is a tensorized form of matrix $Y^{(i)}$. Furthermore, the authors considered an augmented Lagrange’s function as

\[
\begin{align*}
& L_B(\mathcal{X}, \mathcal{Y}^{(1)}, \ldots, \mathcal{Y}^{(N)}, Y^{(1)}, \ldots, Y^{(N)}, \beta) \\
= & \sum_{i=1}^{N} \left\{ \alpha_i \|Y^{(i)}\|_* + \langle \mathcal{W}^{(i)}, \mathcal{X} - Y^{(i)} \rangle \right\} \\
& + \frac{\beta}{2} \|\mathcal{X} - Y^{(i)}\|_F^2,
\end{align*}
\]

where $\mathcal{W}^{(i)}$ is a Lagrange’s multiplier tensor. Actually, [15] is an extension of [5], and it can be minimized by using ADMM optimization scheme. Then the authors developed new update rules for $\mathcal{X}$, $\mathcal{W}^{(i)}$, and $Y^{(i)}$ based on the ADMM with $L_\beta$, and proposed to update three kinds of parameters alternately and iteratively with increasing $\beta$ in [31].

Recently, alternative very efficient completion was proposed as the simultaneous tensor decomposition and completion (STDC) [9]. This method is based on the Tucker decomposition and minimizes the nuclear norm of individual factor matrices by solving the following optimization problem:

\[
\begin{align*}
\text{minimize} & \quad \mathbf{g}, U^{(1)}, \ldots, U^{(N)} \sum_{i=1}^{N} \alpha_i \|U^{(i)}\|_* + \delta \text{tr}(\Phi L \Phi^T) + \gamma \|\mathbf{G}\|_F^2, \\
\text{s.t.} & \quad \mathbf{X} = \mathbf{g} \times \{U^{(i)}\}_{i=1}^{N}, \quad \mathcal{T}_\Omega = \mathbf{T}_\Omega,
\end{align*}
\]

where $\mathbf{G}$ is a core tensor, $U^{(i)}$ is a factor matrix, $\mathbf{g} \times \{U^{(i)}\}_{i=1}^{N} := \mathbf{g} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 \ldots \times_N U^{(N)}$ is the Tucker decomposition model of tensor, $\Phi := (U^{(1)} \otimes \ldots \otimes U^{(N)})$ is a unified factor matrix which consists of all factor matrices, $\otimes$ is the Kronecker product, and $L$ is a matrix designed by some prior information. The STDC method is based on the ‘low-rank Tucker decomposition’ and ‘factor prior’. For example, smoothness of images can be considered as the factor prior, and the term $\delta \text{tr}(\Phi L \Phi^T)$ is corresponding to the smoothness constraints. Thus, the STDC can be considered as a tensor extension of low-rank and smooth matrix completion. To perform [16], the authors considered an augmented Lagrange’s function as

\[
L_C(\mathbf{X}, \mathbf{g}, U^{(1)}, \ldots, U^{(N)}, \mathbf{W}, \beta)
= \sum_{i=1}^{N} \alpha_i \|U^{(i)}\|_* + \delta \text{tr}(\Phi L \Phi^T) + \gamma \|\mathbf{G}\|_F^2, \\
+ \langle \mathbf{W}, \mathbf{X} - \mathbf{g} \times \{U^{(i)}\}_{i=1}^{N} \rangle, \\
+ \frac{\beta}{2} \|\mathbf{X} - \mathbf{g} \times \{U^{(i)}\}_{i=1}^{N}\|_F^2.
\]

Finally, another efficient and promising method for the tensor completion is the fully Bayesian CANDECOMP/PARAFAC tensor completion with mixture prior (FBCP-MP) [49]. The FBCP-MP method is based on the ‘probabilistic low-rank PD’ and ‘smoothness’. The FBCP-MP finds an PD with an appropriate tensor rank by using the Bayesian inference without need to tune or adjust any parameters.

Hence, we can summarize the state-of-the-art tensor completion methods as three types: low-rank-nuclear-norm, low-rank-Tucker-decomposition, and low-rank-PD with optional smoothness constraints. Above mentioned tensor completion papers did not discuss the smoothness constraints in details, however smoothness is a quite important factor for visual data completion which is obvious in the matrix completion problems [14, 35, 20]. In this paper, we investigate the smoothness constraints for the low-rank PD and tensor completion. The main objective of this paper is to improve performance of tensor completion, especially when number of missing entries is relatively large.

3 NEW PROPOSED METHOD

In this section, we propose a new algorithm for the PARAFAC (called also polyadic decomposition) tensor completion with smoothness constraint. First, the unconstrained canonical PD model can be formulated as

\[
\mathbf{Z} = \sum_{r=1}^{R} g_r u^{(1)}_r \circ u^{(2)}_r \circ \ldots \circ u^{(N)}_r,
\]

where $u^{(n)}_r$ are feature vectors called ‘components’, $g_r$ are scaling multiplier. Put $U^{(n)} = [u^{(n)}_1, u^{(n)}_2, \ldots, u^{(n)}_R]$ and $\mathbf{G}$ is a super-diagonal core tensor that each super-diagonal element is $\mathbf{G}_{rr+\ldots} = g_r$, then the PD model
can be also denoted by $\mathbf{Z} = [\mathbf{G}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]$. The minimum number of components $R$, which satisfy the equation (18), is called as the ‘tensor rank’ of $\mathbf{Z}$. Generally, when we impose the smoothness constraint for the feature vectors $\mathbf{u}^{(n)}_i$, minimum number of components $R$ will increase because smoothness constraint decrease the flexibility of the decompostion model. For this reason, it is difficult to determine the upper bound of tensor rank of original tensor when the smoothness constraint is imposed. In the approach of the STDC [9] and the FBCP-MP [49], upper bound of the Tucker/tensor rank was determined at first as some large value and next the Tucker/tensor rank was estimated by decreasing the number of components in each iteration. In this paper, we propose a completely different approach which does not need to determine the upper bound of tensor rank because in the proposed method we increase the number of components $R$ gradually from 1 to its optimal value. We call the new method as the Smooth Parafac tensor Completion (SPC).

3.1 Fundamental Problem for the Fixed Rank SPC (FR-SPC)

In this section, we consider the fixed rank version of the SPC (FR-SPC). The optimization problem for the FR-SPC is formulated as

$$
\begin{align*}
\text{minimize} & \quad \frac{1}{2} ||\mathbf{X} - \mathbf{Z}||_F^2 \\
& + \sum_{r=1}^{R} \frac{g_r}{2} \sum_{n=1}^{N} \rho^{(n)} ||L^{(n)} \mathbf{u}_r^{(n)}||_p^p, \\
\text{subject to} & \quad \mathbf{Z} = \sum_{r=1}^{R} g_r \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \cdots \circ \mathbf{u}_r^{(N)}, \\
& \quad \mathbf{X}_\Omega = \mathbf{T}_\Omega, \quad \mathbf{X}_\bar{\Omega} = \mathbf{Z}_{\bar{\Omega}}, \quad ||\mathbf{u}_r^{(n)}||_2 = 1, \\
& \quad \forall r \in \{1, \ldots, R\}, \forall n \in \{1, \ldots, N\},
\end{align*}
$$

where $\mathbf{X}$ is a completed output tensor where the missing entries are filled by the smooth complete approximation $\mathbf{Z}$. Thus the constraints $\mathbf{X}_\Omega = \mathbf{T}_\Omega, \mathbf{X}_\bar{\Omega} = \mathbf{Z}_{\bar{\Omega}}$ stand for

$$
\mathbf{X}_{i_1i_2 \ldots i_N} = \begin{cases} 
\mathbf{T}_{i_1i_2 \ldots i_N} & (i_1, i_2, \ldots, i_N) \in \Omega, \\
\mathbf{Z}_{i_1i_2 \ldots i_N} & \text{otherwise}
\end{cases}
$$

Next, $\rho = [\rho^{(1)}, \rho^{(2)}, \ldots, \rho^{(N)}]^T$ is a smoothness parameter vector, $p \in \{1, 2\}$ is a parameter to select the types of smooth constraints, and matrix $L^{(n)} \in \mathbb{R}^{(I_n - 1) \times I_n}$ is a smoothness constraint matrix typically defined as

$$
L^{(n)} := \begin{pmatrix}
1 & -1 & 1 & -1 & \cdots & 1 & -1 \\
\end{pmatrix}.
$$

The first term $||\mathbf{X} - \mathbf{Z}||_F^2$ of objective function (19) stands for the mean squared error (MSE) between the values of observed entries $\mathbf{T}_\Omega$ and the PD model $\mathbf{Z}_\bar{\Omega}$ because $\mathbf{X}_\Omega = \mathbf{T}_\Omega$ and $\mathbf{X}_\bar{\Omega} = \mathbf{Z}_\bar{\Omega}$. Thus the minimization of the first term of objective function in (19) provides a PD of given tensor $\mathbf{T}$.

The second term of objective function in (19) is a penalty term to assure smooth component vectors $\mathbf{u}_r^{(n)}$. Note that we have $||L^{(n)} \mathbf{u}_r^{(n)}||_p = \sum_{i=1}^{I_n-1} |u_r^{(n)}(i) - u_r^{(n)}(i+1)|^p$, and minimization of this non-smoothness measure provides the smoothness of individual feature vectors. When $p = 1$, the constraint term becomes the total variation (TV), and when $p = 2$, it becomes the squared variation (SV). There are two existing approaches imposing smoothness constraints in completion methods. In [14], [20], authors imposed the smoothness constraints into the output matrix/tensor, directly. On the other hand, the smoothness constraints were imposed into factor matrices (or latent components) in [9], [49]. In other words, the former and the latter approaches assume surface and causal smoothness, respectively. Our method assumes the causal smoothness of the data, and impose the TV and SV constraints into the individual component vectors.

3.2 Derivation of the FR-SPC Algorithm

We solve the optimization problem (19) by using hierarchical alternating least squares (HALS) approach [11], [12]. According to HALS approach, we consider...
the minimization of local cost functions:

\[
\begin{aligned}
&\text{minimize}_{g_r, u_r^{(1)}, \ldots, u_r^{(N)}} \frac{1}{2} ||\mathbf{y}_r - g_r u_r^{(1)} \circ u_r^{(2)} \circ \cdots \circ u_r^{(N)}||_F^2 \\
&\quad + \frac{g_r^2}{2} \sum_{n=1}^{N} \rho(n) ||L(n) u_r^{(n)}||^p_p,
\end{aligned}
\]  
(22)

s.t. \( ||u_r^{(n)}||_2 = 1, \)

\( \forall n \in \{1, \ldots, N\}, \forall r \in \{1, \ldots, R\}, \) where \( \mathbf{y}_r = \mathbf{X} - \sum_{i \neq r} g_i u_i^{(1)} \circ u_i^{(2)} \circ \cdots \circ u_i^{(N)} \). The local-problem (22) is only for \text{rth} components of \text{PD} with fixed \( \mathbf{X} \). We update \( u_r^{(1)}, u_r^{(2)}, \ldots, u_r^{(N)} \) and \( g_r \) for each \( r \) by solving the local optimization problem (22).

### 3.2.1 Update Rule for \( g_r \)

To solve the problem (22) with respect to \( g_r \), objective function can be transformed as

\[
F_p^{(r)}(g_r) := \frac{1}{2} g_r^2 - g_r (\mathbf{y}_r, u_r^{(1)} \circ u_r^{(2)} \circ \cdots \circ u_r^{(N)})
\]

\[
+ \frac{1}{2} g_r^2 \sum_{n=1}^{N} \rho(n) ||L(n) u_r^{(n)}||_p^p.
\]  
(23)

The gradient of the cost function is given by

\[
\nabla F_p^{(r)} = g_r \left( \frac{1}{2} \sum_{n=1}^{N} \rho(n) ||L(n) u_r^{(n)}||_p^p \right)
\]

\[ - (\mathbf{y}_r, u_r^{(1)} \circ u_r^{(2)} \circ \cdots \circ u_r^{(N)}). \]  
(24)

Hence, by equaling it to zero, we obtain an update rule for \( g_r \) as follow:

\[
g_r \leftarrow \frac{\mathbf{y}_r (\mathbf{y}_r, u_r^{(1)} \circ u_r^{(2)} \circ \cdots \circ u_r^{(N)})}{(1 + \sum_{n=1}^{N} \rho(n) ||L(n) u_r^{(n)}||_p^p)}.
\]  
(25)

### 3.2.2 Update Rule for \( u_r^{(n)} \)

Next, we solve the problem (22) with respect to \( u_r^{(n)} \), for this purpose the objective function can be transformed as

\[
F_p^{(r,n)}(u_r^{(n)}) := \frac{g_r^2}{2} \rho(n) ||L(n) u_r^{(n)}||^p_p - g_r u_r^{(n)T} y_r^{(n)}
\]

\[
+ \frac{1}{2} g_r^2 ||u_r^{(n)}||^p_p.
\]  
(26)

where \( y_r^{(n)} := \text{vec}(\mathbf{y}_r \times_1 u_r^{(1)T} \times_2 \cdots \times_n-1 u_r^{(n-1)T} \times_{n+1} u_r^{(n+1)T} \times_{n+2} \cdots \times_N u_r^{(N)T}) \). When \( p = 1 \), the sub-gradient of objective function is given by

\[
\nabla F_1^{(r,n)}(u_r^{(n)}) := \frac{g_r^2}{2} \rho(n) L(n) u_r^{(n)} \text{sgn}(L(n) u_r^{(n)})
\]

\[ - g_r y_r^{(n)} + g_r^2 u_r^{(n)}, \]  
(27)

where \( \text{sgn}(L(n) u_r^{(n)}) \) is a vector function which outputs the signs of individual elements of an input vector. Then, we update the \( u_r^{(n)} \) by

\[
\begin{aligned}
&u_r^{(n)} \leftarrow \text{argmin}_u F_1^{(r,n)}(u), \quad (28) \\
u_r^{(n)} \leftarrow u_r^{(n)} / ||u_r^{(n)}||_2, \quad (29)
\end{aligned}
\]

### 3.3 Model Selection for the Number of Components \( R \)

The key problem of the FR-SPC is how to choose the optimal number of components \( R \). The PD model with too small \( R \) is not able to fit the data and the PD model with too large \( R \) may cause the over-fitting problem. To estimate an optimal value of \( R \), we gradually increase the number of \( R \) till we achieved desired fit by formulating the following optimization problem:

\[
\begin{aligned}
&\text{minimize}_{R} \quad R, \\
&\text{s.t.} \quad ||\mathbf{Z}_\Omega - \mathbf{T}_\Omega||_F^2 \leq \varepsilon, \\
&\quad \mathbf{Z} \in \mathbb{S}(R, p, \rho),
\end{aligned}
\]  
(34)

where minimization of \( F_1^{(r,n)} \) is a convex optimization problem and it can be solved by any gradient based optimizations such as steepest descent method. When \( p = 2 \), the gradient of objective function is given by

\[
\nabla F_2^{(r,n)}(u_r^{(n)}) := g_r^{2 \rho(n)} L(n)^T L(n) u_r^{(n)}
\]

\[ - g_r y_r^{(n)} + g_r^2 u_r^{(n)}. \]  
(30)

By setting \( \nabla F_2^{(r,n)} = 0 \), we have

\[
\text{argmin}_u F_2^{(r,n)}(u) = \frac{1}{g_r} (I + \rho(n) L(n)^T L(n))^{-1} y_r^{(n)}. \]  
(31)

Hence we obtain finally the following update rule:

\[
\begin{aligned}
&u_r^{(n)} \leftarrow (I + \rho(n) L(n)^T L(n))^{-1} y_r^{(n)}, \quad (32) \\
u_r^{(n)} \leftarrow u_r^{(n)} / ||u_r^{(n)}||_2. \quad (33)
\end{aligned}
\]

The FR-SPC optimization scheme can be summarized as Algorithm 2.

### 3.4 Algorithm 2: Algorithm for Estimation of Optimal Number of Components \( R \)

1. \textbf{input:} \( \mathbf{T}, \Omega, p, \rho, \) and SDR.
2. \( \varepsilon \leftarrow 10^{-10} \frac{\text{SDR}}{10} ||\mathbf{T}_\Omega||_F^2; \)
3. \( R \leftarrow 1; \)
4. \textbf{repeat}
5. \( [\mathbf{Z}, \mathbf{X}] \leftarrow \text{FR-SPC}((\mathbf{T}, \Omega, R, p, \rho)); \)
6. \( R \leftarrow R + 1; \)
7. \textbf{until} \( ||\mathbf{Z}_\Omega - \mathbf{T}_\Omega||_F^2 \leq \varepsilon \)
8. \textbf{output:} \( \mathbf{Z}, \mathbf{X}. \)

The criterion (34) allows us to finds a tensor \( \mathbf{Z} \) based on the smooth PD model with the minimum number of components that guarantee sufficient accuracy to fit the input tensor \( \mathbf{T} \). In order to guarantee that signal to distortion ratio (SDR) is bounded to specific threshold, we can define the error bound as \( \varepsilon = 10^{-\frac{\text{SDR}}{10}} ||\mathbf{T}_\Omega||_F^2 \). Note that \( \min_{\mathbf{Z} \in \mathbb{S}(R, p, \rho)} \{||\mathbf{Z}_\Omega - \mathbf{T}_\Omega||_F^2\} \) is a monotonically non-increasing function
Algorithm 3 SPC Algorithm (accelerated version of Algorithm 2)

1: input: $T$, $\Omega$, $p$, $p$, SDR, and $\nu$.
2: $\varepsilon \leftarrow 10^{-\text{SDR}/10} ||T_{\Omega}||^2_F$.
3: $X_0 \leftarrow T_{\Omega}$; $X_0 \leftarrow$ average of $T_{\Omega}$;
4: Construct matrix $L^{(n)}$ by (21); $\forall n \in \{1, ..., N\}$
5: $R \leftarrow 1$.
6: Initialize $\{u^{(n)}_{R}\}_{n=1}^{N}$, randomly, where $||u^{(n)}_{R}||_2 = 1$;
7: $g_{R} \leftarrow \{X_{n}, u^{(1)}_{R} \circ u^{(2)}_{R} \circ \cdots \circ u^{(N)}_{R} \}$;
8: $E = X - \sum_{r=1}^{R} g_{R} u^{(1)}_{R} \circ u^{(2)}_{R} \circ \cdots \circ u^{(N)}_{R}$;
9: $E_{0} = 0$;
10: $t \leftarrow 0$;
11: $\mu_{t} \leftarrow ||E||^2_F$;
12: repeat
13: Update $\{u^{(n)}_{R}\}_{n=1}^{N}$ by the FR-SPC algorithm (9th-18th lines);
14: $\mu_{t+1} \leftarrow ||E||^2_F$;
15: if $\frac{\mu_{t+1} - \mu_{t}}{\mu_{t+1} - \varepsilon} \leq \nu$ then
16: $R \leftarrow R + 1$;
17: Initialize $\{u^{(n)}_{R}\}_{n=1}^{N}$, randomly, where $||u^{(n)}_{R}||_2 = 1$;
18: $g_{R} \leftarrow \{E_{n}, u^{(1)}_{R} \circ u^{(2)}_{R} \circ \cdots \circ u^{(N)}_{R} \}$;
19: $E = E - g_{R} u^{(1)}_{R} \circ u^{(2)}_{R} \circ \cdots \circ u^{(N)}_{R}$;
20: $E_{0} = 0$;
21: end if
22: $t \leftarrow t + 1$;
23: until $\mu_{t} \leq \varepsilon$
24: $Z \leftarrow \sum_{r=1}^{R} g_{R} u^{(1)}_{R} \circ u^{(2)}_{R} \circ \cdots \circ u^{(N)}_{R}$;
25: $X_{\Omega} \leftarrow Z_{\Omega}$;
26: output: $X$, $Z$.

with respect to $R$. Algorithm 2 allows us to find the best $R$, however the algorithm is quite time-consuming. If current value of $R$ is too small to fit the PD model sufficiently well to the given data tensor during the iteration process, then we can stop the algorithm for current $R$ and increase $R \leftarrow R + 1$, and run the algorithm again for a new increased $R$. In this procedure, we propose to switch in early stage of iterations to increased $R$ if the following condition is met:

$$\frac{\mu_{t} - \mu_{t+1}}{\mu_{t+1} - \varepsilon} < \nu, \quad \text{(35)}$$

where $\mu_{t} = ||Z_{\Omega} - T_{\Omega}||^2_F$, $Z_{t}$ stands for the PD model at iteration step $t$, and $\nu > 0$ is a stopping threshold (typically, $\nu = 0.01$). The left part of (35) is a measure of convergence speed, so this condition means that when the convergence speed becomes substantially slow, we stop the iteration procedure for current $R$. By incorporating this simple stopping criterion, we finally developed an improved and considerably accelerated algorithm for the automatic determination of number of components $R$ of the FR-SPC. We call this method simply “Smooth Parafac tensor Comple-

![Fig. 1. Concept of optimization process in the SPC algorithm. An area of $R$ between arcs shows the space of tensor which spanned by $R$’s components in the smooth PD model. An area which is filled with waves shows the space of tensor which satisfy the condition $||Z_{\Omega} - T_{\Omega}||^2_F \leq \varepsilon$. Concatenated arrows show the optimization process of the SPC algorithm which increases gradually the number of components $R$ from a rank-1 tensor to its end point when criterion is satisfied.](Image)

![Fig. 2. Iso-surface visualization of synthetic data and completion results by SPC-TV and SPC-SV.](Image)

4 EXPERIMENTAL RESULTS

4.1 Convergence Properties using Synthetic 3rd-order Tensor

Fig. 2 shows the iso-surface visualization of synthetic data and completion results by SPC-TV and SPC-SV, and it is summarized in Algorithm 3. Fig. 1 shows the concept of optimization process of the SPC algorithm, and its details are provided in Section 3. Our algorithms have been implemented in MATLAB and are available for readers at https://sites.google.com/site/yokotatsuya/home/software.

![Fig. 2. Iso-surface visualization of synthetic data and completion results by SPC-TV and SPC-SV.](Image)
R with respect to number of iterations of the SPC algorithm with \( \rho = [0.01, 0.01, 0.01] \) for TV smoothing, \( \rho = [1.0, 1.0, 1.0] \) for SV smoothing, with SDR= 30 dB, and various switching threshold values of \( \nu \in \{0.1, 0.01, 0.001, 0.0001\} \). Other parameters were set as SDR= 30 dB, \( \rho = [0.01, 0.01, 0.01] \) for TV smoothing, and \( \rho = [1.0, 1.0, 1.0] \) for SV smoothing. We prepared a text masked image of ‘Barbara’, a scratched image of ‘Peppers’, two incomplete images with random voxels missing (dead pixels) of ‘Giant’, ‘Wasabi’, all color elements of individual pixels (so called dead pixels) were deleted for all missing rates. We applied the SPC-TV and SPC-SV algorithms with SDR = 25 dB, \( \nu = 0.01 \), and various values of \( \rho = [\rho, \rho, 0] \). We defined \( \rho : = \tau/(1 - \tau) \) and tested for wide range of \( \tau \in [0.05, 0.85] \). We found that the larger value of \( \rho \) increase considerably PSNR and SSIM in SV/TV smoothing, however too large \( \rho \) substantially decreases the performances in TV smoothing. Furthermore, larger \( \rho \) requires the larger number of components \( R \) to more precisely fit and the larger computational cost at the same time. The SPC algorithm with TV smoothing required larger number of components \( R \) than SV smoothing, and usually the performances expressed by PSNR and SSIM were inferior in comparison to SV smoothing. According to our results, the SV smoothing is, in general, better than TV smoothing for tensor completion. We discuss the interpretation of these results in Section 5.

### 4.2 Color Image Completion

Fig. 4 shows the images used in our experiments as benchmarks. There are ten color images and each image is a represented as a third order tensor, the size of which is 256 × 256 × 3. We generated incomplete data by deleting the elements of these images randomly with several different missing rates \( \in \{60, 70, 80, 90, 95\%\} \). For the images of ‘Giant’ and ‘Wasabi’, all color elements of individual pixels (so called dead pixels) were deleted for all missing rates.

#### 4.2.1 Performances for Various Values of \( \rho \)

Fig. 3 shows the peak signal-to-noise ratio (PSNR), structural similarity (SSIM) index, and number of components \( R \) for different values of smoothing parameter \( \rho \). A benchmark image ‘Lena’ with random 80% missing pixels was used in this experiment. We applied the SPC-TV and SPC-SV algorithms with SDR = 25 dB, \( \nu = 0.01 \), and various values of \( \rho = [\rho, \rho, 0] \). We defined \( \rho : = \tau/(1 - \tau) \) and tested for wide range of \( \tau \in [0.05, 0.85] \). We found that the larger value of \( \rho \) increase considerably PSNR and SSIM in SV/TV smoothing, however too large \( \rho \) substantially decreases the performances in TV smoothing. Furthermore, larger \( \rho \) requires the larger number of components \( R \) to more precisely fit and the larger computational cost at the same time. The SPC algorithm with TV smoothing required larger number of components \( R \) than SV smoothing, and usually the performances expressed by PSNR and SSIM were inferior in comparison to SV smoothing. According to our results, the SV smoothing is, in general, better than TV smoothing for tensor completion. We discuss the interpretation of these results in Section 5.

#### 4.2.2 Comparison with the State-of-the-art Methods

Fig. 6 shows the results of tensor completions using all 10 benchmark images with various random pixel missing rates by using the proposed method with TV and SV smoothing and compares performance with the state of the arts methods: the LTVNN [20], the HaLRTC [31], the STDC [9] and the FBCP-MP [19]. The parameters for the SPC algorithms were set as \( \rho = [0.1, 0.1, 0.0] \) for TV smoothing, \( \rho = [0.5, 0.5, 0.0] \) for SV smoothing, SDR= 25 dB, \( \nu = 0.01 \). The hyper-parameters for the other methods were optimally tuned manually. Interestingly, the LTVNN outperformed the HaLRTC for several images in respect to SSIM, however it is difficult to say whether the LTVNN outperformed the HaLRTC in respect to PSNR. This implies that the smoothness constraint improves the structural similarity of images. Thus we can conclude that the smoothed methods (i.e., LTVNN, STDC, FBCP-MP and the proposed method) are evaluated better by SSIM than PSNR. The proposed method with TV smoothing was inferior to STDC and FBCP-MP in most cases (e.g., ‘Peppers’ and ‘House’), however the proposed method with SV smoothing outperformed considerably the all existing methods for all our benchmarks.

Fig. 7 shows several results of completed images by using the existing methods and the proposed method. We prepared a text masked image of ‘Barbara’, a scratched image of ‘Peppers’, two incomplete images with random pixels missing (dead pixels) of ‘Giant’, and two incomplete image with random voxels missing of ‘Lena’. We also tested for the first time an extreme case where the ‘99%’ pixels are missing,
Fig. 4. Test images are $256 \times 256$ pixels color images which consist of three layers of red, green, and blue color images.

Fig. 5. Results of PSNR, SSIM, and number of components of the SPC algorithms using the image ‘Lena’ with missing rate 80%. We performed various values of $\rho = [\rho, \rho, 0]^T$, where $\rho$ is calculated by $\rho = \tau / (1 - \tau)$. Other parameters were set as $\text{SDR} = 25 \text{ dB}$ and $\nu = 0.01$. The algorithms were relatively insensitive to wide variation of parameters.

Fig. 6. Comparison of performance (PSNR and SSIM subtracted by baseline (LTVNN)) for all benchmark images and for various missing rates (from 60% to 95%) obtained by the proposed methods (SPC-TV and SPC-SV), and state-of-the-art algorithms: LTVNN, HaLRTC, STDC, and FBCP-MP.
and such a large missing rate may be an interesting challenge in this research field. There were rather small difference in performance between the most of methods in the first and second rows, however, we observed big differences for random pixels/voxels missing images, especially when the ratio of missing pixels were high. In the reconstructed images of last three rows by the non-smooth methods (i.e., ADMM and HaLRTC), it was difficult to recognize the shapes of objects. In the reconstructed images of second last row, the smooth matrix completion method (i.e., LTVNN) was not able to reconstruct the woman’s facial parts, however it was not difficult to recognize the woman’s facial parts of the second last row by applying STDC, FBCP-MP, and SPC. For the last row with 99% missing ratio, the shape of ‘Lena’ could be recognized by only the FBCP-MP and the proposed SPC-SV, but the performance of the SPC-SV was better than the FBCP-MP.

4.3 MRI Image Completion

Fig. 8 shows the results for several slices of MRI 3D-images (109 × 91 × 91) with 60%-95% missing voxels obtained by the HaLRTC, the STDC, and the SPC with TV and SV smoothing. Since the MRI image is smooth in all three dimensions (modes), the hyper-parameters of the SPC algorithm were set as $\rho = [0.1, 0.1, 0.1]$ for TV smoothing and $\rho = [0.5, 0.5, 0.5]$ for SV smoothing. Other parameters were set as SDR=25 dB and $\nu = 0.01$ in both methods. The hyper-parameters of other methods were tuned optimally and manually. Table 1 shows the performance (SDR) of the individual algorithms. From Fig. 8 and Table 1 we observed that the SPC algorithm succeeded to complete the incomplete MRI 3D-images even with 90%-95% missing voxels and also outperformed significantly the other methods in respect to the SDR.
This reason, in this experiment we used the smooth PARAFAC decomposition for tensor completion.

Fig. 8. Results of tensor completion for MRI data by using the HaLRTC, the STDC, and the proposed SPC-TV and SPC-SV for various missing rates.

Fig. 9. Results of tensor completion for CMU face dataset by using HaLRTC, STDC, and the proposed SPC-TV and SPC-SV. Values of SDR [dB] are described in each figure. Considerable improvements of performance were achieved.

4.4 4th-order Tensor Completion using a CMU Faces Dataset

Next, we applied the three completion methods to a facial image database provided by Carnegie Mellon University (CMU) called as ‘CMU faces’ [38]. The CMU faces is a 4D-tensor the size of which is $(30 \times 11 \times 21 \times 1024)$. In this data, a face image is described as a 1024-dimensional vector which is a vectorized from a $(32 \times 32)$-gray-scaled-image. For this reason, in this experiment we used the smooth constraint matrix as a combination of the vertical and horizontal smoothness:

$$L^{(n)} = \begin{pmatrix} L_v^{(n)} \\ L_h^{(n)} \end{pmatrix},$$

where $L_v^{(n)}$ and $L_h^{(n)}$ are vertical and horizontal smoothness matrices, respectively. The smoothness parameters of the SPC were set as $\rho = [0, 0, 0, 0.05]$ for TV smoothing and $\rho = [0, 0, 0, 0.1]$ for SV smoothing. Other parameters were set as SDR=$25$ dB and $\nu = 0.01$ in both methods. Fig. 9 shows the original and incompeleted CMU faces with $80\%$ missing faces, and its completed results obtained by the HaLRTC, the STDC, and the SPC algorithms. In this case the HaLRTC failed to complete faces, while the STDC provided several broken faces, however SPC provided excellent results.

| Table 1 |
| --- |
| Signal to distortion ratio of MRI completion |
| methods | 60% | 70% | 80% | 90% | 95% |
| HaLRTC | 14.59 | 12.40 | 9.95 | 6.93 | 4.99 |
| STDC | 14.97 | 13.78 | 12.86 | 11.46 | 11.94 |
| Proposed (TV) | 19.56 | 17.88 | 15.93 | 13.19 | 10.81 |
| Proposed (SV) | 20.62 | 19.06 | 17.25 | 14.77 | 12.86 |

5 DISCUSSIONS

5.1 Advantages of Our Approach

In this section, we discuss the advantages of the SPC algorithm. First, we do not need to determine the upper-bound of the tensor rank in the proposed method. If there is no smoothness constraint, the weak upper-bound of tensor rank is $\min_k((\prod_{n \neq k} I_n) \times 1)$ for general $N$-th order tensor $X \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$.

because any $X$ can be factorized by only one dense matrix $U^{(k)} = X^{(k)}$ and a unit matrix constructed as $I = U^{(1)} \odot U^{(2)} \odot \cdots \odot U^{(k-1)} \odot U^{(k+1)} \odot \cdots \odot U^{(N)} \in \mathbb{R}^{\prod_{n \neq k} I_n \times \prod_{n \neq k} I_n}$. However, for smooth constraint such a unit matrix is not suitable, and it is difficult to estimate the upper-bound of tensor rank because it depends on the imposed level of smoothness. Since the proposed method does not need to determine the upper-bound, it can be easily applied to the practical smooth PARAFAC decomposition for tensor completion problems.

Second, the proposed model selection scheme is useful for the initialization of the algorithm. Generally, even if we know the exact minimum (canonical) tensor rank of the smooth PARAFAC/Tucker decomposition for completion problem, the solution may not be unique. If we consider several local optimal solutions of the problem, the result of optimization methods will depend on the initialization. Fig. 1 shows the concept of optimization process of the SPC algorithm. The algorithm starts from the rank-1 tensor ($R = 1$) the initialization of which has no critical
meaning at this moment. The initialization of rank-$R$ tensor factorization is given by rank-$(R - 1)$ tensor factorization which would give better initialization for completion problem because this initialization come from the lower-rank tensor space. When algorithm is stopped finally, the SPC algorithm is able to find a good solution which is close to the lower-rank tensor space.

5.2 TV Smoothing and SV Smoothing

In this paper, we consider the two types of smooth constraints: TV and SV smoothing. The TV smoothing was often used for the image denoising and restoration. If we consider the minimization of TV/SV term of a signal $z = [0, \text{NaN}, 2, \text{NaN}, 0]^T$, the solutions are respectively $z_{TV} = [0, a, b, 0]^T$ where $a, b \in [0, 2]$, and $z_{SV} = [0, 1, 2, 1, 0]^T$. This means that the enforcement of SV term is stronger than TV term.

In this paper, the smooth constraint matrix $L(n)$ is defined as \(^{[21]}\), however there are other options to construct a such matrix, for example \(^{[12]}\)

$$L_2(n) := \begin{pmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ \vdots & \vdots & \vdots \\ -1 & 2 & -1 \end{pmatrix} \in \mathbb{R}^{(I_n-2) \times I_n}.$$  \quad (37)

The linear slope is not penalized in this smoothness constraint. In fact, we can design various types of smoothness constraint matrices easily, and such matrices can be applied to the SPC algorithm by only replacing the smooth constraint matrix $L(n)$.

6 CONCLUSIONS

In this paper, we proposed a new low-rank smooth PARAFAC decomposition for tensor completion problems. Our approach and algorithms are quite different from the existing methods, instead of setting the upper bound of the expected rank of tensor, our algorithm increases the number of components gradually till reaching the optimal one. We considered two types of smoothness constraint: total variation (TV) and squared variation (SV) in our SPC method, which outperforms the state-of-the-art algorithms, especially, HaLRTC, STDC, and FBCP-MP.

ACKNOWLEDGMENT

This work was supported by Japan Society for the Promotion of Science (JSPS) KAKENHI Grant Number 15K16067.

REFERENCES

[1] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup. Scalable tensor factorizations for incomplete data. Chemometrics and Intelligent Laboratory Systems, 106(1):41–56, 2011.

[2] Y. Amit, M. Fink, N. Srebro, and S. Ullman. Uncovering shared structures in multiclass classification. In Proceedings of the 24th international conference on Machine learning, pages 17–24. ACM, 2007.

[3] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In Advances in Neural Information Processing Systems 19, pages 41–48. MIT Press, 2007.

[4] A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. IEEE Transactions on Image Processing, 18(11):2419–2434, 2009.

[5] D. P. Bertsekas and J. N. Tsitsiklis. Parallel and Distributed Computation: Numerical Methods. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[6] J. Carroll and J.-J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of “Eckart-Young” decomposition. Psychometrika, 35:283–319, 1970.

[7] 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 Transactions on Image Processing, 20(11):3097–3111, 2011.

[8] C. Chen, B. He, and X. Yuan. Matrix completion via an alternating direction method. IMA Journal of Numerical Analysis, 32(1):227–245, 2012.

[9] Y.-L. Chen, C.-T. Hsu, and H.-Y. Liao. Simultaneous tensor decomposition and completion using factor priors. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(3):577–591, 2014.

[10] Z. Chen, A. Cichocki, and T. M. Ruttowski. Constrained non-negative matrix factorization method for EEG analysis in early detection of Alzheimer disease. In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2006.

[11] A. Cichocki, R. Zdunek, and S.-i. Amari. Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization. In Independent Component Analysis and Signal Separation, volume 4666 of LNCS, pages 169–176. Springer, 2007.

[12] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari. Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation. John Wiley & Sons, 2009.

[13] A. Criminisi, P. Pérez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13(9):1200–1212, 2004.

[14] Q. Dai and W. Sha. The physics of compressive sensing and the gradient-based recovery algorithms. arXiv preprint arXiv:0906.1487, 2009.

[15] S. Essid and C. Fevotte. Smooth nonnegative matrix factorization for unsupervised audiovisual document structuring. *IEEE Transactions on Multimedia*, 15(2):415–425, 2013.

[16] S. Gandy, B. Recht, and I. Yamada. Tensor completion and optimization for matrix recovery. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1149–1165, 2011.

[17] X. Geng, K. Smith-Miles, Z.-H. Zhou, and L. Wang. Face image modeling by multilinear subspace analysis with missing values. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 41(3):881–892, 2011.

[18] N. Gillis and F. Glineur. Low-rank matrix approximation with weighted or missing data. *SIAM Journal on Matrix Analysis and Applications*, 32(4):591, 2014.

[19] D. Goldfarb and W. Yin. Second-order cone programming methods for total variation-based image restoration. *SIAM Journal on Scientific Computing*, 27(2):622–645, 2005.

[20] A. Han, J. Wu, L. Wang, Y. Chen, L. Senhadji, and H. Shu. Linear total variation approximate regularized nuclear norm optimization for matrix completion. *Abstract and Applied Analysis*, ID 765782, 2014.

[21] R. Harshman. Foundations of the PARAFAC procedure: Model and conditions for an ‘explanatory’ multi-mode factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84, 1970.
[22] H. Ji, C. Liu, Z. Shen, and Y. Xu. Robust video denoising using low rank matrix completion. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1791–1798. IEEE, 2010.

[23] D. Kressner, M. Steinlechner, and B. Vandereycken. Low-rank tensor completion by Riemannian optimization. *BIT Numerical Mathematics*, 54(2):447–468, 2014.

[24] N. Li and B. Li. Tensor completion for on-board compression of hyperspectral images. In Proceedings of IEEE International Conference on Image Processing (ICIP), pages 517–520. IEEE, 2010.

[25] W. Li, L. Zhao, Z. Lin, D. Xu, and D. Lu. Non-local image inpainting using low-rank matrix completion. *Computer Graphics Forum*, 2014.

[26] X. Li. Compressed sensing and matrix completion with constant proportion of corruptions. *Constructive Approximation*, 37(1):73–99, 2013.

[27] Y. Li, J. Yan, Y. Zhou, and J. Yang. Optimum subspace learning and error correction for tensors. In *Computer Vision–ECCV 2010*, volume 6313 of LNCS, pages 790–803. Springer, 2010.

[28] Y. Li, Y. Zhou, J. Yan, J. Yang, and X. He. Tensor error correction for corrupted values in visual data. In Proceedings of IEEE International Conference on Image Processing (ICIP), pages 2321–2324. IEEE, 2010.

[29] Z. Lin, M. Chen, and Y. Ma. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

[30] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. In Proceedings of International Conference on Computer Vision (ICCV), pages 2114–2121. IEEE, 2009.

[31] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220, 2013.

[32] G. Lv, L. Yin, Y. Jie, and L. Li. Hole-filling by rank sparsity tensor decomposition for medical imaging. *IEICE Transactions on Information and Systems*, 94(2):396–399, 2011.

[33] M. K. Ng, P. Weiss, and X. Yuan. Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods. *SIAM journal on Scientific Computing*, 32(5):2710–2736, 2010.

[34] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4(2):460–489, 2005.

[35] I. Ram, M. Elad, and I. Cohen. Image processing using smooth ordering of its patches. *IEEE Transactions on Image Processing*, 22(7):2764–2774, 2013.

[36] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

[37] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

[38] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1615–1618, 2003.

[39] L. Sorber, M. Van Barel, and L. De Lathauwer. Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank-(l_1,l_2,l_3) terms, and a new generalization. *SIAM Journal on Optimization*, 23(2):695–720, 2013.

[40] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[41] L. R. Tucker. Implications of factor analysis of three-way matrices for measurement of change. In *Problems in measuring change*, pages 122–137. University of Wisconsin Press, 1963.

[42] Y. Wang, J. Yang, W. Yin, and Y. Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008.

[43] D. M. Witten, R. Tibshirani, and T. Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 2009.