2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition

Missing Slice Recovery for Tensors Using a Low-rank Model in Embedded Space

Tatsuya Yokota[1] Burak Erem[2] Seyhmus Guler[2] Simon K. Warfield[2] Hidekata Hontani[1]
[1]Nagoya Institute of Technology, Gokiso, Showa, Nagoya, Japan
[2]Boston Children's Hospital, 300 Longwood Ave., Boston, MA, United States
{t.yokota, hontani}@nitech.ac.jp, {burak.erem, seyhmus.guler, simon.warfield}@childrens.harvard.edu

Abstract

Let us consider a case where all of the elements in some continuous slices are missing in tensor data. In this case, the nuclear-norm and total variation regularization methods usually fail to recover the missing elements. The key problem is capturing some delay/shift-invariant structure. In this study, we consider a low-rank model in an embedded space of a tensor. For this purpose, we extend a delay embedding for a time series to a "multi-way delay-embedding transform" for a tensor, which takes a given incomplete tensor as the input and outputs a higher-order incomplete Hankel tensor. The higher-order tensor is then recovered by Tucker-based low-rank tensor factorization. Finally, an estimated tensor can be obtained by using the inverse multi-way delay embedding transform of the recovered higher-order tensor. Our experiments showed that the proposed method successfully recovered missing slices for some color images and functional magnetic resonance images.

1. Introduction

Matrix/tensor completion is a technique for recovering the missing elements in incomplete data and it has become a very important method in recent years [2, 3, 1, 9, 16, 21, 4, 31, 30]. In general, completion is an ill-posed problem without any assumptions. However, if we have useful prior knowledge or assumptions regarding the data structure, completion can be treated as a well-posed optimization problem, such as convex optimization. The assumption of the structure is also referred to as a "model."

The methods for modeling matrices/tensors can be categorized into two classes. In the first class, the methods directly represent data with the matrices/tensors themselves and some structures of the matrices/tensors are assumed, such as low-rank [2, 3, 1, 9, 16] and smooth properties [32, 11].

also called "tensorization"

By contrast, the methods in the second class "embed" the data into a high-dimensional feature space and it is assumed that the data can be represented by low-rank or a smooth manifold in the embedded space [17, 25, 6, 18] (see Figure 1). Typically, a time series is represented by a "Hankel matrix" (see Section 2.1.1) and its low-rank property has been employed widely for modeling a *linear time-invariant system* of signals [25, 18]. For example, Li *et al.* [15] proposed a method for modeling damped sinusoidal signals based on a low-rank Hankel approximation. Ding *et al.* [6] proposed the use of rank minimization of a Hankel matrix for the video inpainting problem by assuming an autoregressive moving average model. Figure 2 shows an example of occlusion recovery for a noisy time series, which indicates that total variation (TV) and quadratic variation (QV) regularization methods reconstruct a flat estimator, whereas minimization of the Hankel matrix (our proposed method) successfully reconstructs the signal.

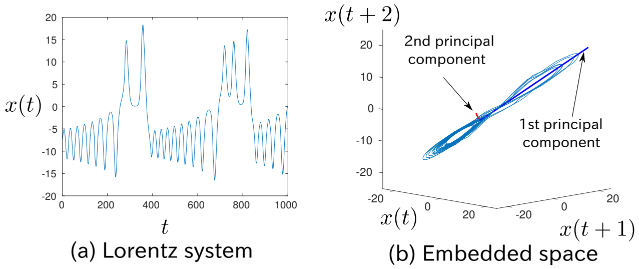
In the proposed method, the incomplete input data are not represented as a Hankel matrix, but instead they are represented as a "higher order Hankel tensor" via multi-way embedding with delay/shift along the time/space axes, and we solve the low-rank tensor completion problem in the embedded space. The minimization of the rank of a matrix/tensor is NP-hard [10] and the problem is often relaxed to nuclear-norm minimization [20]. A disadvantage of the relaxation to nuclear norm minimization is that it decreases the rank of the resultant matrix/tensor as well as the total component values in the matrix/tensor. In particular, nuclear norm minimization often obtains "dark" signals in denoising tasks. Thus, we employ Tucker decomposition for low-rank modeling of the higher order Hankel tensor completion.

The Tucker-based tensor completion is a non-convex optimization problem, and the existing methods usually have difficulty for selecting the step-size parameter. In this study, we propose to use an auxiliary function-based approach, ~~where it~~ improves the convergence characteristics of the optimization process. Moreover, we propose a rank increment scheme for determining the appropriate multi-linear tensor

---

This work was supported in part by JSPS Grant-in-Aid for Scientific Research on Innovative Areas (Multidisciplinary Computational Anatomy): JSPS KAKENHI Grant Number 26108003 and 15K16067.

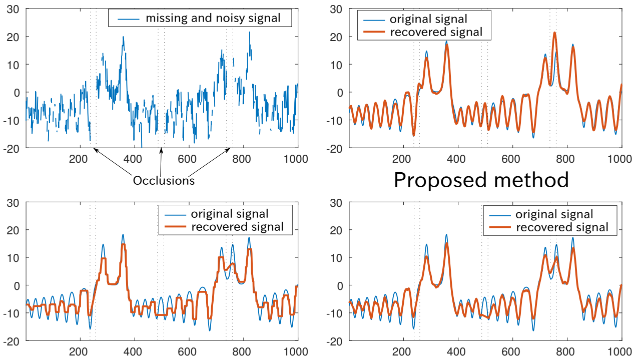
2575-7075/18 $31.00 © 2018 IEEE
DOI 10.1109/CVPR.2018.00861



(a) Lorentz system

(b) Embedded space

Figure 1: Lorentz system and its delay embedded space: A time series signal $x(t)$ can be embedded into a three-dimensional space with individual axes of $x(t)$, $x(t+1)$, and $x(t+2)$. Clearly, most of the points are located on some hyper-plane (i.e., low-rank) in the embedded space.



Proposed method

TV regularization

QV regularization

Figure 2: Recovering a dynamical signal using the proposed method ($\tau = 50$), TV regularization, and QV regularization.

ranks. According to our extensive experiments, the proposed method is highly suitable for tensor completion (*e.g.* recovery of "Lena" from only 1% of the randomly sampled voxels) and it outperforms state-of-the-art tensor completion methods.

1.1. Notations

A vector is denoted by a bold small letter $\boldsymbol{a} \in \mathbb{R}^I$. A matrix is denoted by a bold capital letter $\boldsymbol{A} \in \mathbb{R}^{I\times J}$. A higher-order ($N \geq 3$) tensor is denoted by a bold calligraphic letter $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$. The ith entry of a vector $\boldsymbol{a} \in \mathbb{R}^I$ is denoted by a_i, and the (i,j)th entry of a matrix $\boldsymbol{A} \in \mathbb{R}^{I\times J}$ is denoted by a_{ij}. The $(i_1, i_2, ..., i_N)$th entry of an Nth-order tensor $\boldsymbol{\mathcal{A}}$ is denoted by $a_{i_1 i_2\cdots i_N}$, where $i_n \in \{1, 2, ..., I_n\}$ and $n \in \{1, 2, ..., N\}$. The Frobenius norm of an Nth-order tensor is defined by $||\boldsymbol{\mathcal{X}}||_F := \sqrt{\sum_{i_1,i_2,...,i_N} x^2_{i_1 i_2\cdots i_N}}$.

A mode-k unfolding (matricization) of a tensor $\boldsymbol{\mathcal{X}}$ is denoted as $\boldsymbol{X}_{(k)} \in \mathbb{R}^{I_k\times\Pi_{n\neq k}I_n}$. A mode-$k$ multiplication between a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ and a matrix/vector $\boldsymbol{A} \in \mathbb{R}^{R\times I_k}$ is denoted by $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} \times_k \boldsymbol{A} \in \mathbb{R}^{I_1\times\cdots\times I_{k-1}\times R\times I_{k+1}\times\cdots\times I_N}$, where the entries are given by $y_{i_1\cdots i_{k-1} r i_{k+1}\cdots i_N} = \sum_{i_k} x_{i_1\cdots i_{k-1} i_k i_{k+1}\cdots i_N} a_{r i_k}$, and we have $\boldsymbol{Y}_{(k)} = \boldsymbol{A}\boldsymbol{X}_{(k)}$.

If we consider N matrices $\boldsymbol{U}^{(n)} \in \mathbb{R}^{I_n\times R_n}$ and an N-th order tensor $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{R_1\times R_2\times\cdots\times R_N}$, then the multi-linear tensor product is defined as

$$\boldsymbol{\mathcal{G}} \times \{\boldsymbol{U}\} := \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \cdots \times_N \boldsymbol{U}^{(N)}. \qquad (1)$$

Moreover, a multi-linear tensor product excluding the n-th mode is defined as

$$\boldsymbol{\mathcal{G}} \times_{-n} \{\boldsymbol{U}\} := \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U}^{(1)} \cdots \times_{n-1} \boldsymbol{U}^{(n-1)} \times_{n+1} \boldsymbol{U}^{(n+1)} \cdots \times_N \boldsymbol{U}^{(N)}. \qquad (2)$$

When we consider Tucker decomposition, $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{U}^{(n)}$ in Eq. (1) are referred to as the core tensor and factor matrices, respectively.

2. Proposed method

In this study, we assume a low-rank structure of a higher order Hankel tensor given by the MDT, which is defined in Section 2.1. We denote this by $\mathcal{H}(\cdot)$. The proposed method is conceptually quite simple where it comprises three steps: (1) MDT, (2) low-rank tensor approximation, and (3) inverse MDT.

Let $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I_1\times\cdots\times I_N}$ and $\boldsymbol{\mathcal{Q}} \in \{0,1\}^{I_1\times\cdots\times I_N}$ be the input incomplete tensor and its mask tensor, respectively, and the first step is given by

$$\boldsymbol{\mathcal{T}}_H = \mathcal{H}(\boldsymbol{\mathcal{T}}) \in \mathbb{R}^{J_1\times\cdots\times J_M}, \qquad (3)$$

$$\boldsymbol{\mathcal{Q}}_H = \mathcal{H}(\boldsymbol{\mathcal{Q}}) \in \{0,1\}^{J_1\times\cdots\times J_M}, \qquad (4)$$

=1 (0) if the corresponding entry of T is available (missing)

where $M \geq N$.

In the second step, we obtain a low-rank approximation of $\boldsymbol{\mathcal{T}}_H$ based on the Tucker decomposition model. For example, we first consider the following optimization problem:

$$\underset{\boldsymbol{\mathcal{G}},\{\boldsymbol{U}^{(m)}\}_{m=1}^M}{\text{minimize}} \quad ||\boldsymbol{\mathcal{Q}}_H \circledast (\boldsymbol{\mathcal{T}}_H - \boldsymbol{\mathcal{G}} \times \{\boldsymbol{U}\})||_F^2, \qquad (5)$$
$$\text{s.t.} \;\; \boldsymbol{\mathcal{G}} \in \mathbb{R}^{R_1\times\cdots\times R_M}, \boldsymbol{U}^{(m)} \in \mathbb{R}^{J_m\times R_m} (\forall m),$$

where $R_m \leq J_m (\forall m)$.

Finally, the resultant tensor can be obtained by the inverse MDT of Tucker decomposition:

$$\hat{\boldsymbol{\mathcal{X}}} = \mathcal{H}^{-1}(\boldsymbol{\mathcal{G}} \times \{\boldsymbol{U}\}). \qquad (6)$$

2.1. MDT

2.1.1 Standard delay embedding transform

In this section, we explain the delay embedding operation. For simplicity, we first define a standard delay embedding transform for a vector, which can be interpreted

Look for math relations (e.g., S has collinear columns/rows, could be expressed as a product, and S'*S reminds convolution)

as a time-series signal. Let us consider a vector $\boldsymbol{v} = (v_1, v_2, ..., v_L)^T \in \mathbb{R}^L$, a standard delay embedding transform of $\boldsymbol{v}$ with τ is given by

$$\mathcal{H}_\tau(\boldsymbol{v}) := \begin{pmatrix} v_1 & v_2 & \cdots & v_{L-\tau+1} \\ v_2 & v_3 & \cdots & v_{L-\tau+2} \\ \vdots & \vdots & \ddots & \vdots \\ v_\tau & v_{\tau+1} & \cdots & v_L \end{pmatrix} \in \mathbb{R}^{\tau\times(L-\tau+1)}. \tag{7}$$

Thus, a standard delay embedding transform produces a duplicated matrix from a vector, where this is also referred to as "Hankelization" since $\mathcal{H}_\tau(\boldsymbol{v})$ is a Hankel matrix. If $\boldsymbol{S} \in \{0,1\}^{\tau(L-\tau+1)\times L}$ is a duplication matrix that satisfies

$$\mathrm{vec}(\mathcal{H}_\tau(\boldsymbol{v})) = \boldsymbol{S}\boldsymbol{v}, \tag{8}$$

then the standard delay embedding transform can be obtained by

$$\mathcal{H}_\tau(\boldsymbol{v}) = \mathrm{fold}_{(L,\tau)}(\boldsymbol{S}\boldsymbol{v}), \tag{9}$$

where $\mathrm{fold}_{(L,\tau)} : \mathbb{R}^{\tau(L-\tau+1)} \to \mathbb{R}^{\tau\times(L-\tau+1)}$ is a folding operator from a vector to a matrix.

Next, we consider an inverse transform of standard delay embedding. The forward transform can be decomposed into duplication and folding, so the inverse transform can also be decomposed into the individual corresponding inverse transforms: a vectorization operation and the Moore–Penrose pseudo-inverse $\boldsymbol{S}^\dagger := (\boldsymbol{S}^T\boldsymbol{S})^{-1}\boldsymbol{S}^T$. Thus, the inverse delay embedding transform for a Hankel matrix $\boldsymbol{V}_H$ can be given by

$$\mathcal{H}_\tau^{-1}(\boldsymbol{V}_H) = \boldsymbol{S}^\dagger \mathrm{vec}(\boldsymbol{V}_H). \tag{10}$$

Figure 3 shows an example of the delay embedding transform for a vector, duplication matrix, and its inverse transform. We can see that the duplication matrix comprises multiple identity matrices. It should be noted that the diagonal elements of $(\boldsymbol{S}^T\boldsymbol{S})$ comprise the numbers of duplications for individual elements, which are usually τ, but low for marginal elements.

2.1.2 Tensor extension

We now define the MDT for an N-th order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$. The MDT of $\boldsymbol{\mathcal{X}}$ with $\boldsymbol{\tau} \in \mathbb{N}^N$ is defined by

$$\mathcal{H}_{\boldsymbol{\tau}}(\boldsymbol{\mathcal{X}}) = \mathrm{fold}_{(\boldsymbol{I},\boldsymbol{\tau})}(\boldsymbol{\mathcal{X}} \times_1 \boldsymbol{S}_1 \cdots \times_N \boldsymbol{S}_N), \tag{11}$$

where $\mathrm{fold}_{(\boldsymbol{I},\boldsymbol{\tau})} : \mathbb{R}^{\tau_1(I_1-\tau_1+1)\times\cdots\times\tau_N(I_N-\tau_N+1)} \to \mathbb{R}^{\tau_1\times(I_1-\tau_1+1)\times\cdots\times\tau_N\times(I_N-\tau_N+1)}$ constructs a $2N$-th order tensor from the input N-th order tensor. In a similar manner to how the vector delay-embedding is a combination of linear duplication and folding operations, the MDT is also a combination of multi-linear duplication and multi-way folding operations. Figure 4 shows flowcharts to illustrate single-way and multi-way delay embedding for a matrix. Finally, the inverse MDT for a Hankel tensor $\boldsymbol{\mathcal{X}}_H$ is given by

$$\mathcal{H}_{\boldsymbol{\tau}}^{-1}(\boldsymbol{\mathcal{X}}_H) = \mathrm{unfold}_{(\boldsymbol{I},\boldsymbol{\tau})}(\boldsymbol{\mathcal{X}}_H) \times_1 \boldsymbol{S}_1^\dagger \cdots \times_N \boldsymbol{S}_N^\dagger, \tag{12}$$

where $\mathrm{unfold}_{(\boldsymbol{I},\boldsymbol{\tau})} = \mathrm{fold}_{(\boldsymbol{I},\boldsymbol{\tau})}^{-1}$.

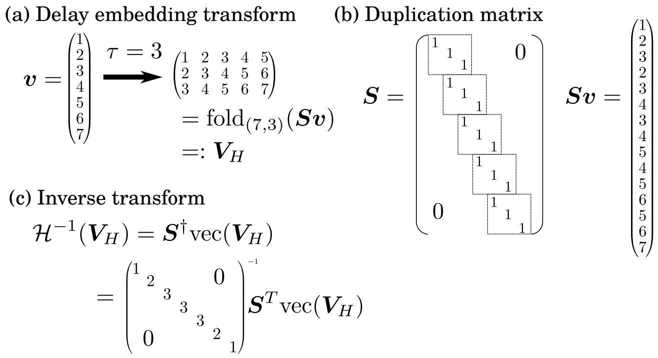

(a) Delay embedding transform
(b) Duplication matrix
(c) Inverse transform
$\boldsymbol{v} =$
$\tau = 3$
$= \mathrm{fold}_{(7,3)}(\boldsymbol{S}\boldsymbol{v})$
$=: \boldsymbol{V}_H$
$\boldsymbol{S} =$
$\boldsymbol{S}\boldsymbol{v} =$
$\mathcal{H}^{-1}(\boldsymbol{V}_H) = \boldsymbol{S}^\dagger \mathrm{vec}(\boldsymbol{V}_H)$
$\boldsymbol{S}^T \mathrm{vec}(\boldsymbol{V}_H)$

Figure 3: A delay embedding transform for a vector.

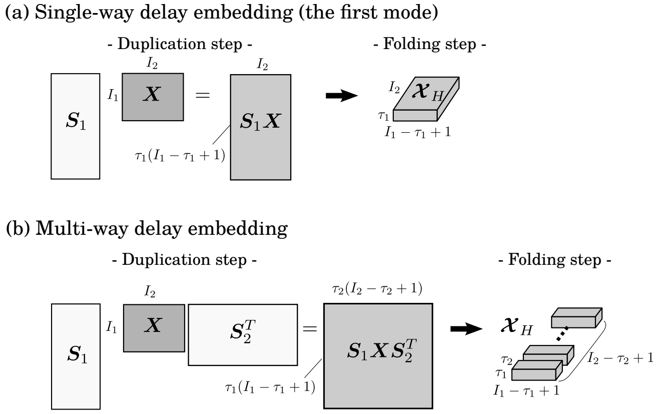

(a) Single-way delay embedding (the first mode)
- Duplication step -
- Folding step -
$\boldsymbol{S}_1$
$\boldsymbol{X}$
$\boldsymbol{S}_1\boldsymbol{X}$
$\boldsymbol{\mathcal{X}}_H$
I_1
I_2
$\tau_1(I_1-\tau_1+1)$
τ_1
$I_1-\tau_1+1$
(b) Multi-way delay embedding
- Duplication step -
- Folding step -
$\boldsymbol{S}_1$
$\boldsymbol{X}$
$\boldsymbol{S}_2^T$
$\boldsymbol{S}_1\boldsymbol{X}\boldsymbol{S}_2^T$
$\boldsymbol{\mathcal{X}}_H$
$\tau_2(I_2-\tau_2+1)$
$\tau_1(I_1-\tau_1+1)$
$I_2-\tau_2+1$
τ_2
τ_1
$I_1-\tau_1+1$

Figure 4: Multi-way delay embedding transform for a matrix. Single-way and multi-way delay embedding transforms convert a matrix into third and fourth order tensors, respectively.

2.2. Tucker decomposition algorithms

In this section, we explain the algorithm for solving Problem (5). It should be noted that Problem (5) is not convex, its solution is not unique, and it is not easy to obtain its global solution [13]. In the case of Tucker decomposition without missing elements, it is known that the alternating least squares (ALS) [5] can efficiently obtain its stationary point. In the case with missing elements, algorithms for obtaining solutions have been proposed that use the gradient descent method [8] and manifold optimization [14, 12] in recent years. Gradient descent is usually slow to converge and manifold optimization can accelerate it by correcting its update direction on the manifold. However, a common issue with both methods is step-size parameter selection because the convergence time is sensitive to the step-size parameter.

We also propose to use an "auxiliary function" based approach to perform Tucker decomposition with missing elements. The proposed algorithm is very simple but efficient because the ALS can be incorporated and it has no adjusting parameters. First, we define the original cost function and auxiliary function by

This is the majorization-minimization (MM) idea

$$f(\theta) := ||\boldsymbol{\mathcal{Q}}_H \circledast (\boldsymbol{\mathcal{T}}_H - \boldsymbol{\mathcal{X}}_\theta)||_F^2, \tag{13}$$

$$\begin{aligned} h(\theta|\theta') := & ||\boldsymbol{\mathcal{Q}}_H \circledast (\boldsymbol{\mathcal{T}}_H - \boldsymbol{\mathcal{X}}_\theta)||_F^2 \\ & + ||\overline{\boldsymbol{\mathcal{Q}}}_H \circledast (\boldsymbol{\mathcal{X}}_{\theta'} - \boldsymbol{\mathcal{X}}_\theta)||_F^2, \end{aligned} \tag{14}$$

where $\theta = \{\boldsymbol{\mathcal{G}}, \boldsymbol{U}^{(1)}, ..., \boldsymbol{U}^{(M)}\}$ is a set of parameters, $\overline{\boldsymbol{\mathcal{Q}}}_H = \mathbf{1} - \boldsymbol{\mathcal{Q}}_H$ represents a complement set of $\boldsymbol{\mathcal{Q}}_H$, and $\boldsymbol{\mathcal{X}}_\theta = \boldsymbol{\mathcal{G}} \times \{\boldsymbol{U}\}$ is a Tucker decomposition. Clearly, we have

$$h(\theta|\theta) = f(\theta), \text{and } h(\theta|\theta') \geq f(\theta) \quad (\theta \neq \theta'). \tag{15}$$

Let us consider the following algorithm

$$\theta^{k+1} = \underset{\theta}{\operatorname{argmin}}\, h(\theta|\theta^k), \tag{16}$$

where the cost function is monotonically non-increasing since we have

$$f(\theta^k) = h(\theta^k|\theta^k) \geq h(\theta^{k+1}|\theta^k) \geq f(\theta^{k+1}). \tag{17}$$

It should be noted that θ^{k+1} only has to satisfy $h(\theta^k|\theta^k) \geq h(\theta^{k+1}|\theta^k)$ to have a non-increasing property. Furthermore, the auxiliary function can be transformed by

$$\begin{aligned} h(\theta|\theta^k) &= ||\boldsymbol{\mathcal{Q}}_H \circledast (\boldsymbol{\mathcal{T}}_H - \boldsymbol{\mathcal{X}}_\theta)||_F^2 \\ &\quad + ||\overline{\boldsymbol{\mathcal{Q}}}_H \circledast (\boldsymbol{\mathcal{X}}_{\theta^k} - \boldsymbol{\mathcal{X}}_\theta)||_F^2 \\ &= ||(\boldsymbol{\mathcal{Q}}_H \circledast \boldsymbol{\mathcal{T}}_H + \overline{\boldsymbol{\mathcal{Q}}}_H \circledast \boldsymbol{\mathcal{X}}_{\theta^k}) \\ &\quad - (\boldsymbol{\mathcal{Q}}_H + \overline{\boldsymbol{\mathcal{Q}}}_H) \circledast \boldsymbol{\mathcal{X}}_\theta||_F^2 \\ &= ||\widetilde{\boldsymbol{\mathcal{T}}}_H^k - \boldsymbol{\mathcal{X}}_\theta||_F^2, \end{aligned} \tag{18}$$

Why?
(Here is why h was so defined)

where $\widetilde{\boldsymbol{\mathcal{T}}}_H^k = \boldsymbol{\mathcal{Q}}_H \circledast \boldsymbol{\mathcal{T}}_H + \overline{\boldsymbol{\mathcal{Q}}}_H \circledast \boldsymbol{\mathcal{X}}_{\theta^k}$. Thus, the minimization of the auxiliary function itself can be regarded as the standard Tucker decomposition without missing elements, which can be solved efficiently using the ALS.

In practice, the proposed algorithm comprises the following two steps: (1) calculate the auxiliary tensor by

$$\boldsymbol{\mathcal{Z}} \leftarrow \boldsymbol{\mathcal{Q}}_H \circledast \boldsymbol{\mathcal{T}}_H + \overline{\boldsymbol{\mathcal{Q}}}_H \circledast \boldsymbol{\mathcal{X}}_{\theta^k}; \tag{19}$$

and (2) update $\boldsymbol{\mathcal{G}}$ and $\{\boldsymbol{U}^{(m)}\}_{m=1}^M$ using the ALS [5] to optimize

$$\begin{aligned} \underset{\boldsymbol{\mathcal{G}}, \{\boldsymbol{U}^{(m)}\}_{m=1}^M}{\text{minimize}} \quad & ||\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{G}} \times \{\boldsymbol{U}\}||_F^2, \\ \text{s.t.} \quad & \boldsymbol{U}^{(m)T}\boldsymbol{U}^{(m)} = \boldsymbol{I}_{R_m} \quad (\forall m). \end{aligned} \tag{20}$$

Tucker with orthogonal factors

The orthogonality constraint for each $\boldsymbol{U}^{(m)}$ supports the uniqueness of the solution for Tucker decomposition and it does not change the reconstructed tensor from the original optimization problem. Finally, the proposed Tucker-based tensor completion is summarized in Algorithm 1. Algorithm 1 updates $\boldsymbol{U}^{(m)}$ and $\boldsymbol{\mathcal{G}}$ for only one cycle of the ALS, and it does not achieve a strict minimization of the auxiliary function. However, it is guaranteed to decrease the auxiliary function because each update obtains the global minimum of the sub-optimization problem of (20) with respect to the corresponding parameter. Thus, Algorithm 1 still has the monotonic convergence property.

Algorithm 1 Tucker-based tensor completion

```
input: $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{J_1 \times \cdots \times J_M}$, $\boldsymbol{\mathcal{Q}} \in \{0,1\}^{J_1 \times \cdots \times J_M}$, and $(R_1, ..., R_M)$.
initialize: $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{R_1 \times \cdots \times R_M}$, and $\{\boldsymbol{U}^{(m)} \in \mathbb{R}^{J_m \times R_m}\}_{m=1}^M$, randomly.
repeat
    $\boldsymbol{\mathcal{X}} \leftarrow \boldsymbol{\mathcal{G}} \times \{\boldsymbol{U}\}$;
    $\boldsymbol{\mathcal{Z}} \leftarrow \boldsymbol{\mathcal{Q}} \circledast \boldsymbol{\mathcal{T}} + \overline{\boldsymbol{\mathcal{Q}}} \circledast \boldsymbol{\mathcal{X}}$;
    for $m = 1, ..., M$ do
        $\boldsymbol{\mathcal{Y}} \leftarrow \boldsymbol{\mathcal{Z}} \times_{-m} \{\boldsymbol{U}^T\}$;
        $\boldsymbol{U}^{(m)} \leftarrow R_m$ leading singular vectors of $\boldsymbol{Y}_{(m)}$;
    end for
    $\boldsymbol{\mathcal{G}} \leftarrow \boldsymbol{\mathcal{Z}} \times \{\boldsymbol{U}^T\}$;
until convergence
output: $\boldsymbol{\mathcal{G}}, \boldsymbol{U}^{(1)}, ..., \boldsymbol{U}^{(M)}$;
```

2.3. Tucker decomposition with rank increment

A difficult and important issue with the Tucker-based tensor completion method is determining an appropriate rank setting $(R_1, ..., R_M)$. If we aim to obtain the lowest rank setting for sufficient approximation, the rank estimation problem can be considered as

$$\begin{aligned} \underset{(R_1,...,R_M)}{\text{minimize}} \quad & \sum_m R_m, \\ \text{s.t.} \quad & ||\boldsymbol{\mathcal{Q}}_H \circledast (\boldsymbol{\mathcal{T}}_H - \boldsymbol{\mathcal{X}})||_F^2 \leq \epsilon, \\ & \text{rank}(\boldsymbol{\mathcal{X}}) = (R_1, ..., R_M), \end{aligned} \tag{21}$$

where ϵ is a noise threshold parameter. However, we do not know the existence of the unique solution $(R_1^*, ..., R_M^*)$ for Problem (21) and it will be dependent on ϵ. Furthermore, even if the best rank setting is unique, the resultant low-rank tensor $\boldsymbol{\mathcal{X}}^*$ is not unique. To address this issue, we propose the use of a very important strategy called the "rank increment" method. Figure 5 provides a flowchart to illustrate the concept of Tucker decomposition with/without rank increment. The rank increment strategy has been discussed in several studies of matrix and tensor completion [19, 7, 22, 24, 27, 30], but the present study is its first

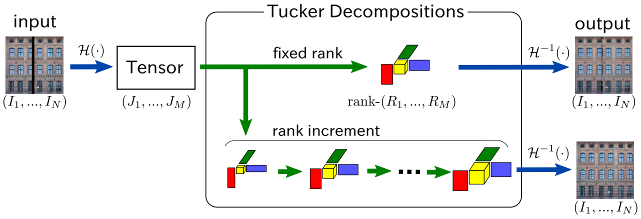


Figure 5: Conceptual illustrations of the proposed methods.

application to Tucker-based completion to the best of our knowledge. The main reason for using the rank increment method is the non-uniqueness of the solution for the tensor $\boldsymbol{\mathcal{X}}$. Thus, the resultant tensor depends on its initialization. The main feature of the rank increment method is that the tensor should be initialized by a lower rank approximation than its target rank. Based on this strategy, the proposed algorithm can be described as follows.

- Step 1: Set initial $R_m = 1$ for all m.
- Step 2: Obtain $\boldsymbol{\mathcal{G}}$ and $\{\boldsymbol{U}^{(m)}\}_{m=1}^M$ with $(R_1, ..., R_M)$ using Algorithm 1 and obtain $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}} \times \{\boldsymbol{U}\}$.
- Step 3: Check the noise condition $||\boldsymbol{\mathcal{Q}}_H \circledast (\boldsymbol{\mathcal{T}}_H - \boldsymbol{\mathcal{X}})||_F^2 \leq \epsilon$, where the algorithm is terminated if it is satisfied; otherwise, go to the next step.
- Step 4: Choose the incremental mode m' and increment $R_{m'}$, and then go back to step 2.

The problem is how to choose m' and how to increase the rank $R_{m'}$. We propose choosing m' using the "m-th mode residual" of the cost function, which is defined as a residual on the multi-linear subspace spanned by all the factor matrices excluding the m-th mode factor. This is mathematically formulated as:

$$m' = \underset{m}{\text{argmax}} \, ||(\boldsymbol{\mathcal{Q}}_H \circledast (\boldsymbol{\mathcal{T}}_H - \boldsymbol{\mathcal{X}})) \times_{-m} \{\boldsymbol{U}^T\}||_F^2. \tag{22}$$

We can interpret this as meaning that the selected m'-th mode has a high expectation of cost reduction when $R_{m'}$ increases while the other-mode ranks remain fixed.

For the rank increment process, we consider the rank sequences for individual modes. For example, the rank sequence for the m-th mode is set as $\boldsymbol{L}_m = [1, 2, 4, 8, ..., J_m]$ because the contribution rates of the singular vectors usually decrease exponentially. Thus, a small rank increment is important for the phase of a low-rank approximation, whereas a small rank increment is not effective for the phase of a relatively high-rank approximation. Large rank steps for the high-rank phase help to accelerate the algorithm, but they should be selected carefully because excessively large rank steps may lead to problems with non-unique solutions. The proposed method for Tucker-based tensor completion with rank increment is summarized in Algorithm 2.

Algorithm 2 Tucker-based tensor completion with rank increment

```
input: $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{J_1 \times \cdots \times J_M}$, $\boldsymbol{\mathcal{Q}} \in \{0,1\}^{J_1 \times \cdots \times J_M}$, $\{\boldsymbol{L}_1, ..., \boldsymbol{L}_M\}$, $\epsilon$, tol.
initialize: $k_m \leftarrow 1, R_m \leftarrow \boldsymbol{L}_m(k_m) \quad (\forall m)$, $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{R_1 \times \cdots \times R_M}$, and $\{\boldsymbol{U}^{(m)} \in \mathbb{R}^{J_m \times R_m}\}_{m=1}^M$;
$\boldsymbol{\mathcal{X}} \leftarrow \boldsymbol{\mathcal{G}} \times \{\boldsymbol{U}\}$;
$f_1 \leftarrow ||\boldsymbol{\mathcal{Q}} \circledast (\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{X}})||_F^2$;
repeat
    Do lines 5-10 in Algorithm 1;
    $\boldsymbol{\mathcal{X}} \leftarrow \boldsymbol{\mathcal{G}} \times \{\boldsymbol{U}\}$;
    $f_2 \leftarrow ||\boldsymbol{\mathcal{Q}} \circledast (\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{X}})||_F^2$;
    if $|f_2 - f_1| \leq$ tol then
        $\widetilde{\boldsymbol{\mathcal{X}}} \leftarrow \boldsymbol{\mathcal{Q}} \circledast (\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{X}})$;
        $m' \leftarrow \text{argmax}_m ||\widetilde{\boldsymbol{\mathcal{X}}} \times_{-m} \{\boldsymbol{U}^T\}||_F^2$;
        $k_{m'} \leftarrow k_{m'} + 1$, and $R_{m'} \leftarrow \boldsymbol{L}_{m'}(k_{m'})$;
    else
        $f_1 \leftarrow f_2$;
    end if
until $f_2 \leq \epsilon$
output: $\boldsymbol{\mathcal{G}}, \boldsymbol{U}^{(1)}, ..., \boldsymbol{U}^{(M)}$;
```

3. Experiments

3.1. Verification of the proposed method using a typical color image

First, in our experiments, we tried to fill the missing slices in a typical color image by using MDT and fixed rank Tucker decomposition. The input image is depicted in Figure 5. We set $\boldsymbol{\tau} = (32, 32, 1)$ and a $(256, 256, 3)$ color image was converted into a $(32, 225, 32, 225, 1, 3)$ tensor. The fifth mode can be ignored so this Hankel tensor was regarded as a fifth-order tensor with a size of $(32, 225, 32, 225, 3)$. Figure 6 shows the images obtained with various settings for the rank parameter. Clearly, low-rank Tucker decomposition with the Hankel tensor successfully filled the missing area. However, an important issue is how to treat the difference between the meanings of (R_1, R_3) and (R_2, R_4). The fundamental difference between (R_1, R_3) and (R_2, R_4) is due to the window sizes of 32 and 225. Thus, it should be noted that a lower (R_1, R_3) may contribute to the representation of the local structure (e.g., smoothness), whereas a lower (R_2, R_4) may contribute to the representation of the global structure (e.g., recursive textures) of the image. In Figure 6, when we compare two flows from the bottom right to the top right, and from the bottom right to the bottom left, the low-rankness of (R_2, R_4) is clearly more important than that of (R_1, R_3) for recovering the missing area.

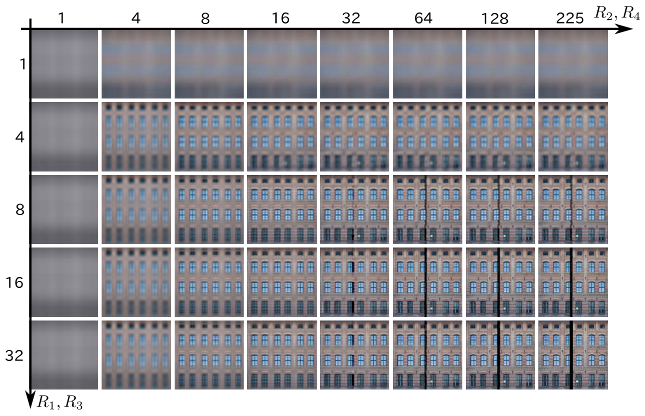


Figure 6: Results obtained by fixed rank MDT Tucker decomposition with $\boldsymbol{\tau} = (32, 32, 1)$ for various rank settings.

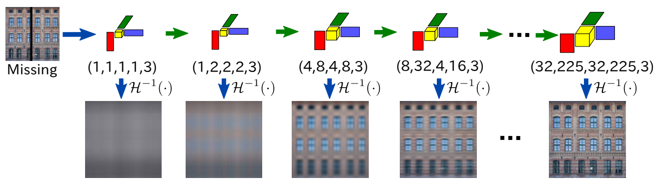

Missing
(1,1,1,1,3)
(1,2,2,2,3)
(4,8,4,8,3)
(8,32,4,16,3)
(32,225,32,225,3)
$\mathcal{H}^{-1}(\cdot)$
$\mathcal{H}^{-1}(\cdot)$
$\mathcal{H}^{-1}(\cdot)$
$\mathcal{H}^{-1}(\cdot)$
$\mathcal{H}^{-1}(\cdot)$
...

Figure 7: Results obtained by rank increment MDT Tucker decomposition with $\boldsymbol{\tau} = (32, 32, 1)$.

Next, we tried to fill missing color images using MDT and Tucker decomposition with the rank increment method. The rank sequences for the proposed method were set as $\boldsymbol{L}_1 = \boldsymbol{L}_3 = (1, 2, 4, 8, 16, 24, 32)$, $\boldsymbol{L}_2 = \boldsymbol{L}_4 = (1, 2, 4, 8, 16, 32, 64, 96, 128, 160, 192, 225)$, and $\boldsymbol{L}_5 = 3$. Figure 7 shows the main flow for the processed images using the proposed method. Using the rank increment method, we first obtained a Tucker decomposition with a very low-rank setting (e.g., rank-one tensor) and a higher-rank decomposition was then obtained by using the previous lower-rank decomposition for its initialization. We repeatedly obtained a higher-rank decomposition until the residual was sufficiently small. Thus, the rank increment method automatically selected an appropriate rank setting.

3.2. Comparison using color images

We compared the performance of the proposed method with those of state-of-the-art tensor completion algorithms: HaLRTC (nuclear-norm regularization) [16], TV regularization [32], nuclear-norm and TV regularization (LR&TV) [28], STDC (constrained Tucker decomposition) [4], and SPCQV (constrained PARAFAC tensor decomposition) [30]. We prepared six missing images for this experiment. The first image had 11 continuous missing slices along the vertical axis. Several horizontal and vertical slices and many voxels were missing from the second image. Random vertical and horizontal slices were missing from the third and fourth images. In addition, 95% and 99% of the random voxels were missing from the fifth and sixth images, respectively.

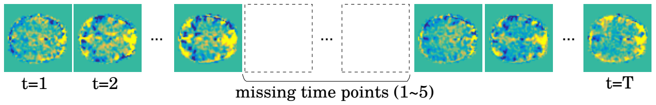


Figure 9: Missing time frames in functional magnetic resonance images.

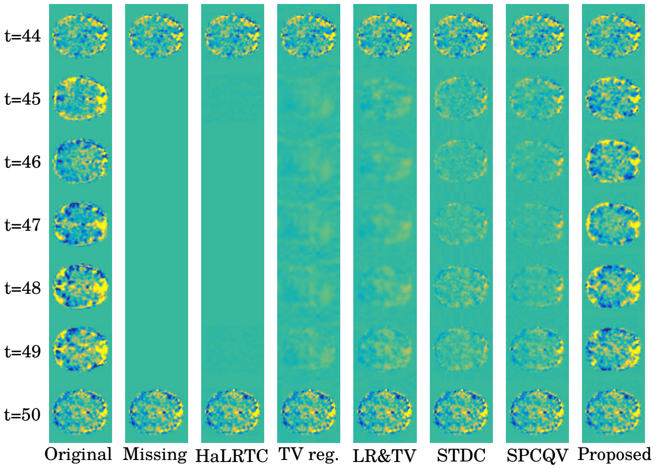


Figure 10: Results recovered for functional magnetic resonance image slices.

Figure 8 shows the experimental results obtained after the completion of various incomplete images. Magnified regions are depicted at the bottom right in the first to fourth images. HaLRTC recovered random missing voxels for the second image, but it failed to recover the missing slices for all of the images. The TV regularization and LR&TV regularization methods filled the missing areas, but the recovered areas were unnaturally flat. STDC failed to recover the missing slices and SPCQV retained “shadows” of the missing slices. By contrast, the proposed method recovered most of the missing slices without shadows. For the image with 95% missing voxels, the proposed method and SPCQV obtained similar results. However, for the image with 99% missing voxels, the result obtained by the proposed method was much better than that by SPCQV. Table 1 shows the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [26] for these comparisons, where the best PSNR and SSIM values are emphasized in bold font. According to this quantitative evaluation, the proposed method performed better than the state-of-the-art methods in terms of the PSNR, and the results in terms of the SSIM were very competitive with SPCQV for some of the images.

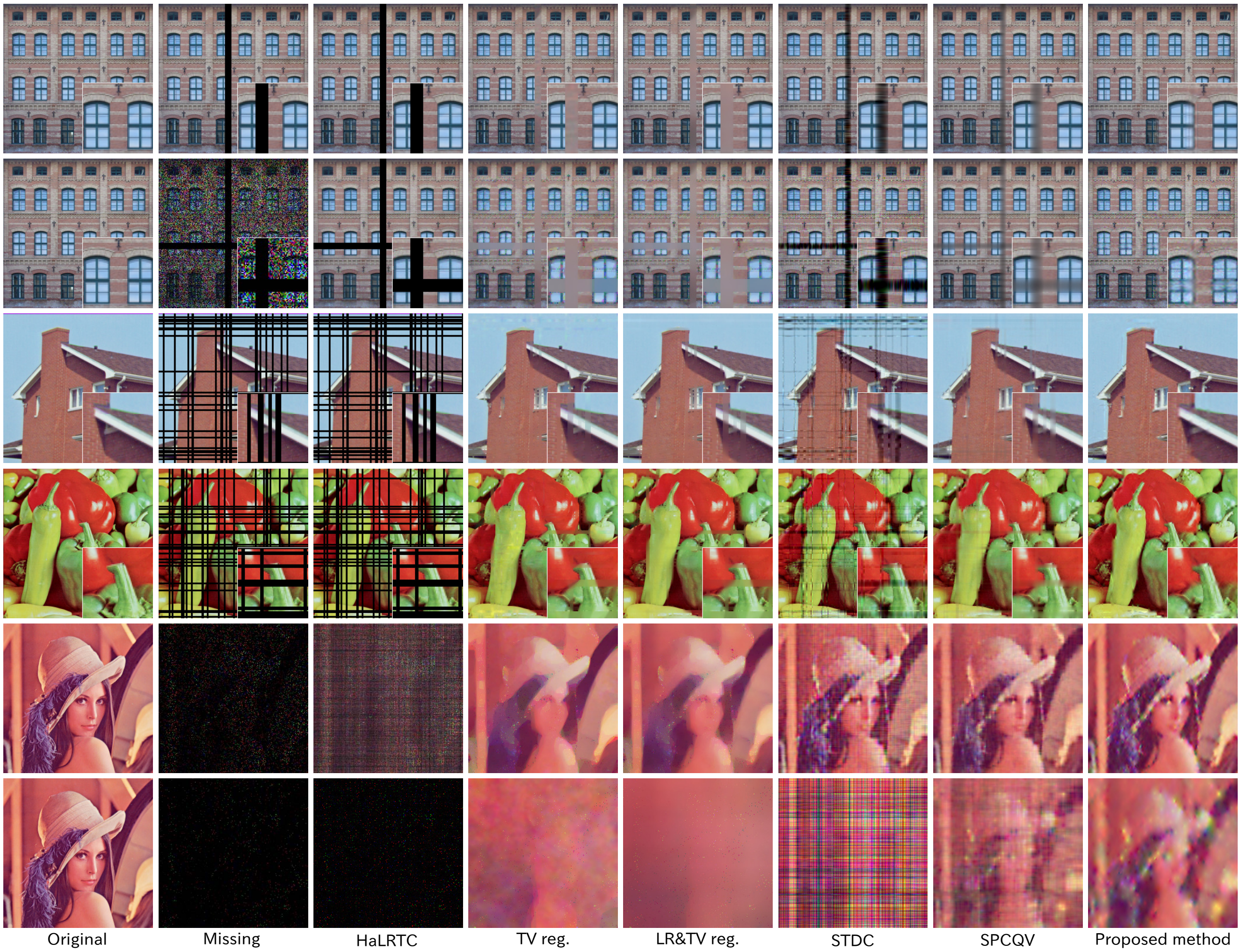


Figure 8: Color images completed with various methods. Six missing color images were artificially generated comprising: slice missing case “facade 1,” slice+voxel missing case “facade 2,” random slice missing cases “house’ and “peppers,” and random voxel missing cases “Lena (95%)’ and “Lena (99%)’.

Table 1: Comparisons of the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) after color image completion.

(PSNR,SSIM)	HaLRTC	TV reg.	LR&TV reg.	STDC	SPCQV	Proposed
facade 1	(19.31,0.937)	(30.08,0.964)	(30.18,0.964)	(24.03,0.958)	(30.04,0.972)	(**36.52**,**0.988**)
facade 2	(16.03,0.852)	(24.28,0.838)	(24.47,0.842)	(17.89,0.826)	(25.65,**0.916**)	(**26.96**,**0.916**)
house	(8.81,0.271)	(26.75,0.909)	(26.70,0.913)	(20.65,0.624)	(27.04,**0.909**)	(**27.50**,0.908)
peppers	(9.92,0.288)	(25.84,0.877)	(25.81,0.885)	(21.61,0.662)	(26.59,0.888)	(**27.62**,**0.898**)
Lena (5%)	(9.63,0.145)	(20.86,0.626)	(20.89,0.621)	(22.07,0.596)	(23.52,0.700)	(**23.57**,**0.738**)
Lena (1%)	(5.32,0.021)	(15.47,0.440)	(15.49,0.443)	(12.75,0.089)	(18.49,0.508)	(**19.68**,**0.565**)

3.3. Comparison using fMRI images

In the next experiment, we tried to recover continuous missing time frames in fMRI images. Figure 9 shows the image prepared with missing data. There were 94 time frames of the fMRI slices and one to five continuous time frames were removed. The size of the tensor was $(64 \times 64 \times 94)$. We applied the proposed method with $\boldsymbol{\tau} = (2, 2, 32)$ for the given tensor. The rank sequences were set as $\boldsymbol{L}_1 = \boldsymbol{L}_3 = (1, 2)$, $\boldsymbol{L}_2 = \boldsymbol{L}_4 = \boldsymbol{L}_6 = (1, 2, 4, 8, 16, 32, 63)$, and $\boldsymbol{L}_5 = (1, 2, 4, 8, 16, 32)$.

Figure 10 shows the results of th~~is~~ experiments. Similar to the color image experiments, the ordinary low-rank

Table 2: SNR and mean SSIM after the completion of fMRI slices.

# of missing slices	HaLRTC	TV reg.	LR&TV reg.	STDC	SPCQV	Proposed
1	(20.15,0.991)	(23.13,0.992)	(23.16,0.993)	(23.30,0.993)	(**24.20**,**0.994**)	(23.25,**0.994**)
2	(14.71,0.982)	(16.20,0.982)	(16.29,0.984)	(17.09,0.985)	(17.80,0.985)	(**18.45**,**0.987**)
3	(14.02,0.973)	(15.20,0.972)	(15.17,0.974)	(15.99,0.977)	(**16.58**,0.976)	(14.54,**0.978**)
4	(12.17,0.964)	(12.45,0.962)	(12.46,0.964)	(12.43,0.966)	(12.98,0.966)	(**14.08**,**0.970**)
5	(11.31,0.955)	(11.41,0.952)	(11.59,0.956)	(11.59,0.957)	(12.15,0.957)	(**13.85**,**0.964**)

model could not recover the missing slices. TV regularization obtained flat results and LR&TV regularization produced similar results. STDC and SPCQV obtained some improvements compared with the convex methods, but they were still unclear. By contrast, the results obtained by the proposed method were very clear, although their accuracy was not high. Table 2 shows the signal-to-noise ratio (SNR) and mean SSIM results obtained for all of the completion methods, which demonstrates that the proposed method performed better than the state-of-the-art methods in terms of the mean SSIM measure and it was also very competitive in terms of the SNR.

4. Discussion

4.1. Novelty and contributions

4.1.1 Low-rank model in embedded space

The idea of tensor completion using MDT and its inverse transform is novel. Most of the existing methods for tensor completion are based on structural assumptions in the signal space, such as nuclear-norm regularization and TV regularization. By contrast, our method considers a structural assumption in the embedded space, which can be regarded as a novel strategy for the tensor completion problem. Furthermore, we employ delay-embedding in this approach and it is extended it in a multi-way manner for tensors.

4.1.2 An auxiliary function-based approach for Tucker decomposition of a tensor with missing elements

The auxiliary function-based algorithm for Tucker decomposition is efficiently employed. The existing algorithms used for the Tucker decomposition of a tensor with missing elements [8, 14, 12] are based on gradient methods. However the convergence speed of the gradient method is quite sensitive to the step-size parameter. By contrast, the proposed algorithm does not have any hyper-parameters and its monotonic convergence is guaranteed by the auxiliary function theory.

4.1.3 Model selection and uniqueness improvement by the rank increment method

The rank increment method for the Tucker decomposition of incomplete tensors is firstly applied ~~in~~ the best of our knowledge. Several methods for estimating multi-linear tensor ranks have been studied only for complete tensors [23, 29]. Methods for estimating the ranks of matrix and PARAFAC decompositions have been proposed for incomplete data [22, 24, 27, 30], but these methods cannot be applied to our problem. Thus, we proposed a new method for estimating multi-linear ranks for an incomplete tensor, where it can also handle the issue of non-unique solutions.

4.2. Computational bottleneck

The proposed method has an issue with data volume expansion due to MDT. An N-th order tensor is converted into a $2N$-th order tensor by MDT and its data size increases roughly $\prod_{n=1}^{N} \tau_n$-fold. This issue makes it difficult to apply the proposed method to large-scale tensors and this problem will be addressed in future research.

5. Conclusions

In this study, we proposed a novel method and algorithm for tensor completion problems that include missing slices. The recovery of missing slices is recognized as a difficult problem that ordinary tensor completion methods usually fail to solve. To address this problem, we introduced the concept of "delay embedding" from the study of dynamical systems and extended it for our problem. We showed that missing slices can be recovered by considering low-rank models in embedded spaces and that MDT is a good choice for this purpose.

At present, the proposed method is very basic but it has many potential extensions such as using different embedding transformations and constrained tensor decompositions (e.g., non-negative, sparse, and smooth). The MATLAB code will be available via our website[1].

---

[1] `https://sites.google.com/site/yokotatsuya/home/software`

References

[1] E. J. Candes and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010. 1

[2] E. J. Candes and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009. 1

[3] E. J. Candes and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010. 1

[4] 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. 1, 6

[5] L. De Lathauwer, B. De Moor, and J. Vandewalle. On the best rank-1 and rank-(r_1, r_2,..., r_n) approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1324–1342, 2000. 3, 4

[6] T. Ding, M. Sznaier, and O. I. Camps. A rank minimization approach to video inpainting. In *Proceedings of ICCV*, pages 1–8. IEEE, 2007. 1

[7] S. V. Dolgov and D. V. Savostyanov. Alternating minimal energy methods for linear systems in higher dimensions. Part I: SPD systems. *arXiv preprint arXiv:1301.6068*, 2013. 4

[8] M. Filipovic and A. Jukic. Tucker factorization with missing data with application to low-n-rank tensor completion. *Multidimensional Systems and Signal Processing*, 26(3):677–692, 2015. 3, 8

[9] S. Gandy, B. Recht, and I. Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2), 2011. 1

[10] N. Gillis and F. Glineur. Low-rank matrix approximation with weights or missing data is NP-hard. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1149–1165, 2011. 1

[11] X. Guo and Y. Ma. Generalized tensor total variation minimization for visual data recovery. In *Proceedings of CVPR*, pages 3603–3611, 2015. 1

[12] H. Kasai and B. Mishra. Low-rank tensor completion: a Riemannian manifold preconditioning approach. In *Proceedings of ICML*, pages 1012–1021, 2016. 3, 8

[13] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009. 3

[14] D. Kressner, M. Steinlechner, and B. Vandereycken. Low-rank tensor completion by Riemannian optimization. *BIT Numerical Mathematics*, 54(2):447–468, 2014. 3, 8

[15] Y. Li, K. R. Liu, and J. Razavilar. A parameter estimation scheme for damped sinusoidal signals based on low-rank Hankel approximation. *IEEE Transactions on Signal Processing*, 45(2):481–486, 1997. 1

[16] 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. 1, 6

[17] E. N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, 1963. 1

[18] I. Markovsky. Structured low-rank approximation and its applications. *Automatica*, 44(4):891–909, 2008. 1

[19] B. Mishra, G. Meyer, F. Bach, and R. Sepulchre. Low-rank optimization with trace norm penalty. *SIAM Journal on Optimization*, 23(4):2124–2149, 2013. 4

[20] 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. 1

[21] F. Shi, J. Cheng, L. Wang, P.-T. Yap, and D. Shen. Low-rank total variation for image super-resolution. In *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 155–162. Springer, 2013. 1

[22] M. Tan, I. W. Tsang, L. Wang, B. Vandereycken, and S. J. Pan. Riemannian pursuit for big matrix recovery. In *Proceedings of ICML*, pages 1539–1547, 2014. 4, 8

[23] M. E. Timmerman and H. A. Kiers. Three-mode principal components analysis: Choosing the numbers of components and sensitivity to local optima. *British Journal of Mathematical and Statistical Psychology*, 53(1):1–16, 2000. 8

[24] A. Uschmajew and B. Vandereycken. Greedy rank updates combined with Riemannian descent methods for low-rank optimization. In *Proceedings of International Conference on Sampling Theory and Applications*, pages 420–424, 2015. 4, 8

[25] P. Van Overschee and B. De Moor. Subspace algorithms for the stochastic identification problem. In *Proceedings of IEEE Conference on Decision and Control*, pages 1321–1326. IEEE, 1991. 1

[26] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 6

[27] T. Yokota and A. Cichocki. A fast automatic rank determination algorithm for noisy low-rank matrix completion. In *Proceedings of APSIPA ASC*, pages 43–46, 2015. 4, 8

[28] T. Yokota and H. Hontani. Simultaneous visual data completion and denoising based on tensor rank and total variation minimization and its primal-dual splitting algorithm. In *Proceedings of CVPR*, pages 3732–3740, 2017. 6

[29] T. Yokota, N. Lee, and A. Cichocki. Robust multilinear tensor rank estimation using higher order singular value decomposition and information criteria. *IEEE Transactions on Signal Processing*, 65(5):1196–1206, 2017. 8

[30] T. Yokota, Q. Zhao, and A. Cichocki. Smooth PARAFAC decomposition for tensor completion. *IEEE Transactions on Signal Processing*, 64(20):5423–5436, 2016. 1, 4, 6, 8

[31] Q. Zhao, L. Zhang, and A. Cichocki. Bayesian CP factorization of incomplete tensors with automatic rank determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1751–1763, 2015. 1

[32] M. Zhu and T. Chan. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. *UCLA CAM Report*, pages 08–34, 2008. 1, 6