Tensor Completion with Shift-invariant Cosine Bases

Tatsuya Yokota and Hidekata Hontani
Nagoya Institute of Technology, Nagoya, Japan
E-mail: t.yokota@nitech.ac.jp

***Abstract*—In this study, we discuss a technique of *tensor completion using multi-way delay-embedding*, which is an emerging framework for the tensor completion problem. This consists of simple three steps: (1) multi-way delay-embedding transform (MDT) of the input incomplete tensor, (2) completing the transformed high-order tensor, (3) inverse MDT of the completed high-order tensor. In spite of the simplicity, it can be used as a powerful tool for recovering the missing elements and slices of tensors. In this paper, we propose an improvement method for MDT based tensor completion by exploiting a common phenomenon that the most real signals are commonly having Fourier bases as shift-invariant features in its auto-correlation matrix. By considering the cosine bases in high-order tensor, several factor matrices in the low-rank tensor decomposition problem can be automatically decided. The experimental results show the advantages of the proposed method.**

I. Introduction

Completion is a signal processing technique to estimate the values of missing elements in incomplete data. The model/assumption of data is the most important factor to decide the completion results. Many models have been studied for the completion problem such as low rank matrix model [9]–[12], [14], [32], [33], [35], [40], [47], [54], [55], [60], [64], [72], low total variation (TV) models [17], [27]–[29], [38], [52], [68], [69], low Tucker rank tensor model [15], [23], [24], [31], [34], [41], [42], [77], low CP rank tensor model [1], [30], [50], [51], [61], [62], [65], [71], [74], [76], and tensor-train/network model [2], [3], [26], [57], [73], [75]. In general, the best universal model does not exist, and individual models have strong and weak points at the same time with respect to applications.

Usually, matrix/tensor completion problems can be separated into two types of formulations: The one formulation is given by

$$\underset{\boldsymbol{\mathcal{X}}}{\text{minimize}} \quad ||\boldsymbol{\mathcal{Q}} \circledast (\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{X}})||_F^2, \quad \text{s.t.} \quad \boldsymbol{\mathcal{X}} \in \mathcal{A}, \tag{1}$$

where $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is an input incomplete tensor, $\boldsymbol{\mathcal{Q}} \in \{0,1\}^{I_1 \times I_2 \times \cdots \times I_N}$ is a binary tensor indicating observed/missing elements by 1/0, $\boldsymbol{\mathcal{X}}$ is an output complete tensor, and $\mathcal{A} \subset \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is a model based subset. For example, $\mathcal{A}_R$ can be defined by a set of low-rank matrices, $\{\boldsymbol{X} = \boldsymbol{W}\boldsymbol{H} \in \mathbb{R}^{I_1 \times I_2} | \boldsymbol{W} \in \mathbb{R}^{I_1 \times R}, \boldsymbol{H} \in \mathbb{R}^{R \times I_2}\}$, in rank-$R$ matrix completion problem [5]–[7], [33], [46], [48]. In this way, the definition of set $\mathcal{A}$ represents the model/assumption in this formulation. Factorization based matrix/tensor completion models can be included in this formulation.

The another formulation is given by

$$\underset{\boldsymbol{\mathcal{X}}}{\text{minimize}} \quad f(\boldsymbol{\mathcal{X}}), \quad \text{s.t.} \quad ||\boldsymbol{\mathcal{Q}} \circledast (\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{X}})||_F^2 \leq \epsilon, \tag{2}$$

where $f : \mathbb{R}^{I_1 \times \cdots \times I_N} \rightarrow \mathbb{R}$ is a cost function which represents the model/assumption, and $\epsilon \geq 0$ is a noise threshold parameter. When observed entries of $\boldsymbol{\mathcal{T}}$ are noise-free, we set $\epsilon = 0$. For example, when the cost function is defined by the matrix nuclear-norm, $f_{LR}(\boldsymbol{X}) = ||\boldsymbol{X}||_*$, then the problem is characterized as the well-known nuclear norm minimization [11]. When the cost function is defined by the total variation (TV), $f_{TV}(\boldsymbol{X}) = ||\boldsymbol{X}||_{TV}$, then the problem is characterized as the TV minimization [78]. Many studies have considered convex functions for $f(\cdot)$, and the convex optimization algorithms have been well established at the same time [4], [8], [13], [16], [22], [25], [67].

Usually, the former approach must solve some complicated non-convex optimization, however it has a high model flexibility and the high level completion can be performed. In the previous study [66], we have considered the former model based on the low rank tensor factorization in embedded space, and reported the promising results. The new tensor completion model can capture the shift-invariant features of tensor, and succeed to recover the missing slices in tensors. In this paper, we tackle the further improvements of the method by using the cosine functions based on the common phenomenon in real world signals.

The rest of this paper is organized as follow: In Section I-A, we first define the notations used in this paper. Section II briefly review the technique of delay-embedding based tensor completion [66]. In Section III, we show some examples of shift-invariant features and propose an improvement technique for the delay-embedding based tensor completion. Section IV shows the experimental results to demonstrate the advantages of the proposed method. In Section V, we discuss the related works and some issue in the MDT based methods. Finally, we conclude this paper in Section VI.

A. 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 or $\boldsymbol{a}(i)$, 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 $a_{i_1i_2\cdots i_N}$, 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 $||\boldsymbol{\mathcal{A}}||_F := \sqrt{\sum_{i_1,i_2,...,i_N} a^2_{i_1i_2\cdots i_N}}$.

A mode-k unfolding (matricization) of a tensor $\boldsymbol{\mathcal{X}}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ 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 $\boldsymbol{\mathcal{X}}$ 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 $y_{i_1\cdots i_{k-1}ri_{k+1}\cdots i_N}=\sum_{i_k}x_{i_1\cdots i_{k-1}i_ki_{k+1}\cdots i_N}a_{ri_k}$, 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

$$\boldsymbol{\mathcal{G}}\times\{\boldsymbol{U}\} := \boldsymbol{\mathcal{G}}\times_1\boldsymbol{U}^{(1)}\times_2\boldsymbol{U}^{(2)}\cdots\times_N\boldsymbol{U}^{(N)}. \qquad (3)$$

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

$$\boldsymbol{\mathcal{G}}\times_{-n}\{\boldsymbol{U}\} := \boldsymbol{\mathcal{G}}\times_1\boldsymbol{U}^{(1)}\cdots\times_{n-1}\boldsymbol{U}^{(n-1)} \times_{n+1}\boldsymbol{U}^{(n+1)}\cdots\times_N\boldsymbol{U}^{(N)}. \qquad (4)$$

When we consider Tucker decomposition, $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{U}^{(n)}$ in Eq. (3) are referred to as the core tensor and factor matrices, respectively.

II. Delay-embedding based tensor completion

In this section, we review a technique of tensor completion using multi-way delay embedding [66]. The framework consists of simple three steps as follow:

Step 1: Multi-way delay embedding transform (MDT) of the input incomplete tensor, $\boldsymbol{\mathcal{T}}_H=\mathcal{H}(\boldsymbol{\mathcal{T}})$, where $\mathcal{H}(\cdot)$ is the MDT operator.

Step 2: Completion of the transformed high-order tensor, $\boldsymbol{\mathcal{X}}_H=\phi(\boldsymbol{\mathcal{T}}_H)$, where ϕ is a function of tensor completion.

Step 3: Inverse MDT of the completed high-order tensor, $\boldsymbol{\mathcal{X}}=\mathcal{H}^{-1}(\boldsymbol{\mathcal{X}}_H)$, where $\mathcal{H}^{-1}(\cdot)$ is the inverse MDT operator.

Figure 1 shows the concept of MDT based tensor completion.

A. MDT and inverse MDT

In this section, we explain MDT and inverse MDT in details.

1) Standard delay-embedding transform: Delay-embedding is a technique for reconstructing the attractor from the generated/observed time-series signals, which is originally used for the study of dynamical systems [49], and system identification [56]. Recently, the delay-embedding comes into attentions as a useful and powerful tool for the time-series analysis such as brain signals [20], [21], [39], [45], [53].

Delay-embedding is a technically equivalent to the Hankelization. Let $\boldsymbol{v}=(v_1,...,v_L)^T\in\mathbb{R}^L$ be a vector, the standard delay-embedding transform of $\boldsymbol{v}$ with τ is given by

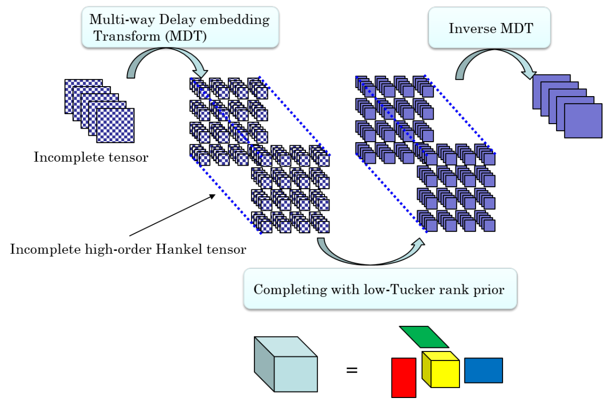

Fig. 1: Concept of MDT based tensor completion

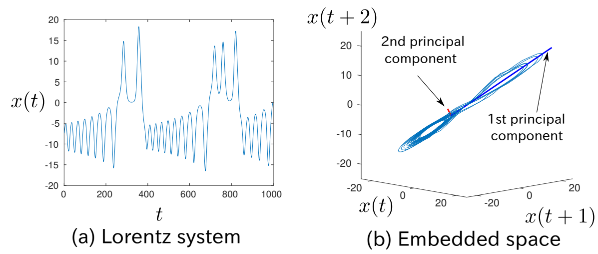

Fig. 2: Lorentz system and its delay embedded space [66]

$$\mathcal{H}_\tau(\boldsymbol{v}) := \begin{pmatrix} v_1 & v_2 & \cdots & v_{L-\tau+1} \\ v_2 & v_3 & \cdots & v_{L-\tau+2} \\ \vdots & \vdots & \ddots & \vdots \\ v_\tau & v_{\tau+1} & \cdots & v_L \end{pmatrix} \in\mathbb{R}^{\tau\times(L-\tau+1)}. \qquad (5)$$

Note that $\mathcal{H}_\tau(\boldsymbol{v})$ is a Hankel matrix. Considering Hankel matrix as a set of τ-dimensional vectors, $\mathcal{H}_\tau(\boldsymbol{v})=[\boldsymbol{h}_1,...,\boldsymbol{h}_{L-\tau+1}]$, a sequence of points in τ-dimensional space is referred to as the reconstructed attractor which generated the signal $\boldsymbol{v}$. Figure 2 shows example of an attractor and a signal. From the Figure 2, we can see the attractor is approximately spanned by the low-dimensional linear subspace.

Delay embedding can be regarded as a linear operation since a unique duplication matrix $\boldsymbol{S}\in\{0,1\}^{\tau(L-\tau+1)\times L}$ exists that satisfies

$$\mathrm{vec}(\mathcal{H}_\tau(\boldsymbol{v})) := [\boldsymbol{h}_1^T,...,\boldsymbol{h}_{L-\tau+1}^T]^T=\boldsymbol{S}\boldsymbol{v}, \qquad (6)$$

where vec$(\cdot)$ is a vectorization operator which unfolds a matrix to a vector. From Eq.(6), we have

$$\mathcal{H}_\tau(\boldsymbol{v})=\mathrm{fold}_{(L,\tau)}(\boldsymbol{S}\boldsymbol{v}), \qquad (7)$$

where $\mathrm{fold}_{(L,\tau)}:\mathbb{R}^{\tau(L-\tau+1)}\to\mathbb{R}^{\tau\times(L-\tau+1)}$ is a folding/reshaping operator from a vector to a matrix. The size of matrix $\boldsymbol{S}$ is little bit large but it is very sparse. The linear algebraic way (7) would be easier than copy-paste

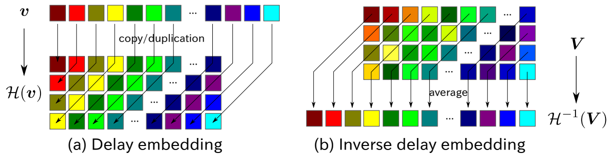

Fig. 3: An conceptual visualization of delay embedding and its inverse transform

while shifting for implementing using some language such as MATLAB and Python, where $\boldsymbol{S}$ should be defined as a sparse matrix.

2) Inverse delay-embedding transform: In this section, we consider to define the inverse transform of delay-embedding, $\mathcal{H}_\tau^{-1}(\cdot)$. Note that the delay embedding transform $\mathcal{H}(\cdot)$ is not a bijective function but an injective function. Thus, we have

$$\boldsymbol{v} = \mathcal{H}_\tau^{-1}(\mathcal{H}_\tau(\boldsymbol{v})) \text{ for any } \boldsymbol{v} \in \mathbb{R}^L, \tag{8}$$

but $\boldsymbol{V} \neq \mathcal{H}_\tau(\mathcal{H}_\tau^{-1}(\boldsymbol{V}))$ when $\boldsymbol{V} \in \mathbb{R}^{\tau\times(L-\tau+1)}$ is a general matrix. In this study, we consider the Moore-Penrose pseudo-inverse of duplication matrix as $\boldsymbol{S}^\dagger := (\boldsymbol{S}^T\boldsymbol{S})^{-1}\boldsymbol{S}^T$, and the inverse delay embedding transform for a matrix $\boldsymbol{V}$ is defined by

$$\mathcal{H}_\tau^{-1}(\boldsymbol{V}) := \boldsymbol{S}^\dagger \text{vec}(\boldsymbol{V}). \tag{9}$$

Figure 3 shows an image of operations of delay-embedding and inverse delay-embedding. We can see that inverse delay-embedding transform outputs the average values of individual anti-diagonal elements of the input matrix.

3) Multi-way extension: We consider to generalize the delay-embedding for high-order tensors, and we refer this as MDT. Roughly speaking, multi-way extension of delay-embedding is the same manner as the convolution. For example, 2d-convolution of matrix data with some window is calculated by shifting window along with x and y axes and obtaining the inner-product of the focused region and the weight function. Two-way delay-embedding is performed by shifting (τ_1, τ_2)-window along with x and y axes and stacking the copy of focused region in the form of a 2d grid (see Figure 5(a)). In similar way, MDT of N-th order tensor is performed by shifting $(\tau_1, ..., \tau_N)$-window along with individual axes and stacking the copy of focused volume in the form of an N-dimensional grid. Thus, MDT of N-th order tensor outputs a $2N$-th order tensor.

In mathematically, MDT of an N-th order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ with $\boldsymbol{\tau} = (\tau_1, ..., \tau_N)$ can be performed by

$$\mathcal{H}(\boldsymbol{\mathcal{X}}) := \text{fold}_{(\boldsymbol{I},\boldsymbol{\tau})}(\boldsymbol{\mathcal{X}} \times_1 \boldsymbol{S}_1 ... \times_N \boldsymbol{S}_N), \tag{10}$$

where $\text{fold}_{(\boldsymbol{I},\boldsymbol{\tau})}(\cdot)$ is a reshaping function of tensors from $\mathbb{R}^{\tau_1(I_1-\tau_1+1)\times\cdots\times\tau_N(I_N-\tau_N+1)}$ to $\mathbb{R}^{\tau_1\times(I_1-\tau_1+1)\times\cdots\times\tau_N\times(I_N-\tau_N+1)}$. The inverse MDT is given by

$$\mathcal{H}^{-1}(\boldsymbol{\mathcal{V}}) := \text{fold}_{(\boldsymbol{I},\boldsymbol{\tau})}^{-1}(\boldsymbol{\mathcal{V}}) \times_1 \boldsymbol{S}_1^\dagger ... \times_N \boldsymbol{S}_N^\dagger. \tag{11}$$

We have $\boldsymbol{\mathcal{X}} = \mathcal{H}^{-1}(\mathcal{H}(\boldsymbol{\mathcal{X}}))$.

B. Low-rank tensor factorization

Let us put

$$\boldsymbol{\mathcal{T}}_H = \mathcal{H}(\boldsymbol{\mathcal{T}}) \in \mathbb{R}^{J_1\times\cdots\times J_M}, \tag{12}$$

$$\boldsymbol{\mathcal{Q}}_H = \mathcal{H}(\boldsymbol{\mathcal{Q}}) \in \{0,1\}^{J_1\times\cdots\times J_M}, \tag{13}$$

then we consider to optimize the following optimization problem

$$\underset{\boldsymbol{\mathcal{G}},\{\boldsymbol{U}^{(m)}\}_{m=1}^M}{\text{minimize}} \ ||\boldsymbol{\mathcal{Q}}_H \circledast (\boldsymbol{\mathcal{T}}_H - \boldsymbol{\mathcal{G}} \times \{\boldsymbol{U}\})||_F^2, \tag{14}$$
$$\text{s.t. } \boldsymbol{\mathcal{G}} \in \mathbb{R}^{R_1\times\cdots\times R_M}, \boldsymbol{U}^{(m)} \in \mathbb{R}^{J_m\times R_m}(\forall m),$$

where $R_m \leq J_m(\forall m)$. The values of R_m decide the multi-linear rank of the completed tensor. Because of the issue of non-uniqueness in the tensor completion, we employ rank increment approach. In other words, we solve Problem (14) with $R_m = 1$ at the first. Furthermore, using the result as initialization, we solve Problem (14) again with increased R_m. We repeat its procedure until that the cost function in (14) is sufficiently small.

Finally, the result of tensor completion is given by

$$\widehat{\boldsymbol{\mathcal{X}}} = \mathcal{H}^{-1}(\widehat{\boldsymbol{\mathcal{G}}} \times \{\widehat{\boldsymbol{U}}\}). \tag{15}$$

III. Proposed Method

A. Visualizing common shift-invariant features

In this section, we show some common properties about the shift-invariant features in visual/time-series data. It has a close relationship with MDT. First, Fig. 4(b) shows an example of the shift-invariant features of the time-series, which is generated as the left singular vectors of singular value decomposition (SVD) of the Hankel matrix (see Fig. 4(a)) with $\tau = 50$. Fig. 4(c) shows the right singular vectors multiplied by individual singular values. We can see that the shift-invariant features is quite similar to the Fourier bases in Fig. 4(b). By contrast, the specific information of the signal affects the global coefficient features in Fig. 4(c).

Next, we consider the shift invariant feature of 2D-image in Fig. 5. MDT with $\boldsymbol{\tau} = [32, 32]$ is performed and Hankel tensor is decomposed by the higher-order SVD (HOSVD) [18] (see Fig. 5(a)). From the first and third factor matrices of HOSVD, we obtain similar patterns to 2D-Fourier bases (see Fig. 5(b)). Global coefficient matrices have the multi-view features related with the convolution of corresponding shift-invariant patterns.

B. MDT based cosine model fitting

In Section III-A, we explained that some real time-series signals/images have shift-invariant features, and it is almost the same as Fourier bases. Thus, we consider it as prior information for the data recovery problem. In other words, we assume that some $\boldsymbol{U}^{(k)}$ $(k \in \mathcal{K})$ corresponding to local shift invariant features are cosine bases, and we remove these $\boldsymbol{U}^{(k)}$ from the optimization parameters in Problem (14). Furthermore, we consider the low-rank assumptions for only

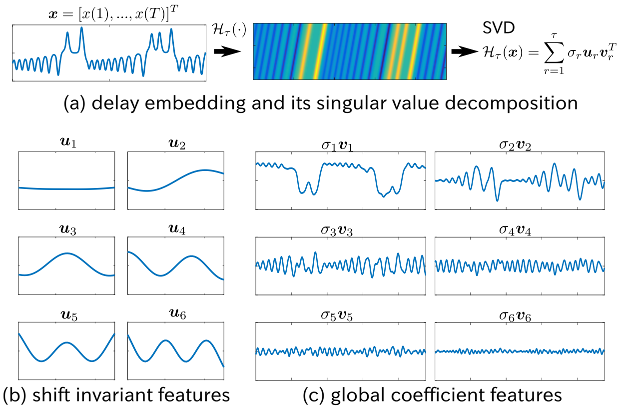

Fig. 4: Shift-invariant features of the time-series signal

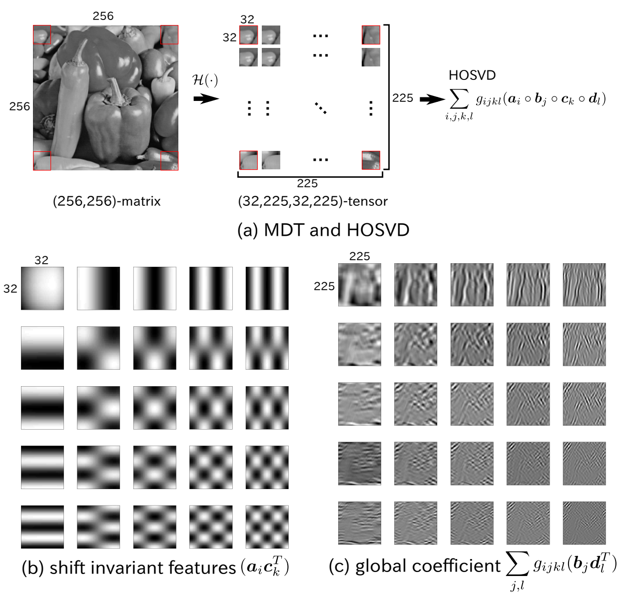

Fig. 5: Shift-invariant features of the gray-scale image

$U^{(k)}$ $(k \in \mathcal{K})$ and some additional modes $r \in \mathcal{R}$ such as RGB colors. Hence, the optimization problem is modified as

$$\begin{aligned} \underset{\theta=\{\mathcal{G}, U^{(r)}, r\in\mathcal{R}\}}{\text{minimize}} \; & f(\theta) := ||\mathcal{Q}_H \circledast (\mathcal{T}_H - \mathcal{G} \times \{U\})||_F^2, \quad (16) \\ \text{s.t. } & \mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_M}, R_m = J_m (m \notin \{\mathcal{K} \cup \mathcal{R}\}), \\ & U^{(m)} = I_{J_m} (m \notin \{\mathcal{K} \cup \mathcal{R}\}), \\ & U^{(k)} = B_{J_k, R_k} (k \in \{\mathcal{K}\}), \\ & U^{(r)} \in \mathbb{R}^{J_r \times R_r} (r \in \{\mathcal{R}\}), \end{aligned}$$

where θ is a set of optimization parameters, I_T is an $(T \times T)$-identity matrix, and $B_{T,L} = [b^{(1)}, ..., b^{(L)}] \in \mathbb{R}^{T \times L}$ is the first L-th discrete cosine bases in T-dimensional space which is defined by

$$b_t^{(l)} := \cos\left\{\frac{\pi(l-1)}{T}\left(t - \frac{1}{2}\right)\right\}, \tag{17}$$

for $t \in \{1, ..., T\}$ and $l \in \{1, ..., L\}$.

1) Optimization algorithm: Here, we consider to solve Problem (16) by using the auxiliary function approach. When we put the auxiliary function with $\theta^c := \{\mathcal{G}^c, U^c\}$ as

$$\begin{aligned} h(\theta|\theta^c) &:= ||\mathcal{T}_H^c - \mathcal{G} \times \{U\}||_F^2, \quad (18) \\ \mathcal{T}_H^c &:= \mathcal{Q}_H \circledast \mathcal{T}_H + (\mathbf{1} - \mathcal{Q}_H) \circledast (\mathcal{G}^c \times \{U^c\}), \end{aligned}$$

then the update rule $\theta^{c+1} \leftarrow \text{argmin}_\theta \, h(\theta|\theta^c)$ has a non-increasing property for the original cost function in (16). Also note that we have $f(\theta^{c+1}) \leq h(\theta^{c+1}|\theta^c) \leq h(\theta^c|\theta^c) = f(\theta^c)$, and the complete minimization is not necessary but it is sufficient that the decreasing the $h(\theta|\theta^c)$ for decreasing $f(\theta)$.

For the implementation, we consider the following two steps: (1) update the auxiliary tensor by

$$\mathcal{Z} \leftarrow \mathcal{Q}_H \circledast \mathcal{T}_H + (\mathbf{1} - \mathcal{Q}_H) \circledast (\mathcal{G}^c \times \{U^c\}), \tag{19}$$

and (2) update $\mathcal{G}$ and $U^{(r)}$ $(r \in \mathcal{R})$ using the alternating least squares (ALS) algorithm to optimize

$$\begin{aligned} \underset{\theta=\{\mathcal{G}, U^{(r)}, r\in\mathcal{R}\}}{\text{minimize}} \; & f(\theta) := ||\mathcal{Z} - \mathcal{G} \times \{U\}||_F^2, \quad (20) \\ \text{s.t. } & \mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_M}, R_m = J_m (m \notin \{\mathcal{K} \cup \mathcal{R}\}), \\ & U^{(m)} = I_{J_m} (m \notin \{\mathcal{K} \cup \mathcal{R}\}), \\ & U^{(k)} = B_{J_k, R_k} (k \in \{\mathcal{K}\}), \\ & U^{(r)T} U^{(r)} = I_{R_r} (r \in \{\mathcal{R}\}). \end{aligned}$$

From the ALS, $\mathcal{G}$ and $U^{(r)}$ are updated by

$$U^{(r)} \leftarrow R_r \text{ leading singular vectors of } Y_{(r)}^{(r)}; (r \in \mathcal{R}) \tag{21}$$

$$\mathcal{G} \leftarrow \mathcal{Z} \times \{U^T\}; \tag{22}$$

where $\mathcal{Y}^{(r)} = \mathcal{Z} \times_{-r} \{U^T\}$.

2) Tensor factorization with rank increment: Rank estimation is an important issue of tensor factorization [63], [70]. In this paper, we employ the rank increment method for obtain optimal rank setting of tensor factorization. We consider the following problem

$$\begin{aligned} \underset{R_l (l \in \{\mathcal{K} \cup \mathcal{R}\})}{\text{minimize}} \; & \sum_l R_l, \\ \text{s.t. } & ||\mathcal{Q}_H \circledast (\mathcal{T}_H - \mathcal{X})||_F^2 \leq \epsilon, \quad (23) \\ & \text{rank}(\mathcal{X}) = (R_1, ..., R_M), \\ & R_m = J_m (m \notin \{\mathcal{K} \cup \mathcal{R}\}), \end{aligned}$$

where ϵ is a noise threshold parameter. This criterion implies a strategy that the low-rank solution is better in a set of the solutions $\mathcal{X}$ which having the same error from the target tensor. However, it is not easy to obtain because there are many combinations of rank setting $(R_1, ..., R_M)$ such like $\prod_l R_l$, and impossible to try all the combinations. In addition, we have another difficulty of the non-uniqueness of the approximated tensors $\mathcal{X}$ even if the best rank setting $(R_1, ..., R_M)$ is known in advance. The non-uniqueness of approximated tensor would be caused by the high ill-posedness of the tensor completion problem.

In order to tackle the issue of non-uniqueness, the rank increment strategy can be applied. In the rank increment strategy, we first obtain $\boldsymbol{\mathcal{X}}$ with very low-rank setting. After that, we re-optimize $\boldsymbol{\mathcal{X}}$ with little increased rank setting using the pre-result as its initial value. We repeat the rank increment and re-optimization of $\boldsymbol{\mathcal{X}}$ until that the error is sufficiently small.

Finally, we summarize the proposed method in Algorithm 1. In the step of low-rank tensor factorization, the 6-11th lines perform the initialization of optimization parameters. Individual factor matrices $\boldsymbol{U}^{(m)}$ are initialized by three ways: identity matrices, cosine basis matrices, and random matrices. The core tensor $\boldsymbol{\mathcal{G}}$ is initialized, randomly. The 15-21th lines updates the optimization parameters $\boldsymbol{U}^{(r)}(r \in \mathcal{R})$ and $\boldsymbol{\mathcal{G}}$ by using the ALS method. The 24-30th lines perform the rank increment. The rank increment has two parts: increment mode selection, and rank increment. In this algorithm, the increment mode i is selected based on the mode-residual δ_m:

$$\delta_m := ||\boldsymbol{\mathcal{E}} \times_1 \boldsymbol{U}^{(1)T} \times \cdots \times_{m-1} \boldsymbol{U}^{(m-1)T} \times_{m+1} \boldsymbol{U}^{(m+1)T} \times \cdots \times_M \boldsymbol{U}^{(M)T}||_F^2, \quad (24)$$

where $\boldsymbol{\mathcal{E}}$ is a residual tensor. Interpretation of δ_m can be regarded as a kind of expected residual improvement when m-th mode is increased. Note that we have $||\boldsymbol{U}^{(m')}\boldsymbol{U}^{(m')T}\boldsymbol{E}_{(m')}||_F^2 = ||\boldsymbol{U}^{(m')T}\boldsymbol{E}_{(m')}||_F^2$ since $\boldsymbol{U}^{(m')}$ is an orthonormal basis matrix. Thus, the tensor-matrix product of $\boldsymbol{\mathcal{E}} \times_{m'} \boldsymbol{U}^{(m')T}$ $(m \neq m')$ banishes the residual related with the complementary space of $\boldsymbol{U}^{(m')}$ $(m \neq m')$. This means that we do not consider to increase m'-th mode rank in this moment. Thus, δ_m is a value of residual considering to increase only the m-th mode rank with fixing the others. After deciding the mode, we increase the i-th mode rank based on rank sequence $\boldsymbol{p}_i$. For example, when we increase the i-th mode rank one by one, then the rank sequence is set as $\boldsymbol{d}_i = [1, 2, 3, 4, 5, ..., J_i]$. In some other way, the rank sequence can be freely and efficiently set such as $\boldsymbol{d}_i = [1, 2, 4, 8, 16, ..., J_i]$ for skipping some boring steps. After the rank increment, some corresponding elements of factor matrices and the core tensor should be updated. If $i \in \mathcal{K}$, then we update the cosine basis matrix by 28-th line. Furthermore, we should expand the size of $\boldsymbol{\mathcal{G}}$ to fit the new rank setting. Note that this expansion does not affect to $\boldsymbol{\mathcal{X}}_H$ since we use zero-padding.

IV. Experiments

A. Behaviors in optimization

In this section, we show the behaviors of cost function and recovering signal in optimization process. In this experiment, we used a color image the size of $(256, 256, 3)$. Note that a color image is interpreted as the third order tensor since it has multiple color frames: red, green, and blue. When we set $\boldsymbol{\tau} = [32, 32, 1]$, then MDT outputs the 6th order tensor the size of $(32, 225, 32, 225, 1, 3)$. In the step of low-rank tensor factorization, we set $\mathcal{K} = \{1, 3, 5\}$, $\mathcal{R} = \{6\}$, and $\overline{\mathcal{K} \cup \mathcal{R}} = \{2, 4\}$ as an example that the three subsets are not empty. Furthermore, the rank sequences are set by $\boldsymbol{d}_1 = \boldsymbol{d}_3 = [1, 2, 4, 8, 16, 32]$, $\boldsymbol{d}_5 = 1$, and $\boldsymbol{d}_6 = [1, 2, 3]$.

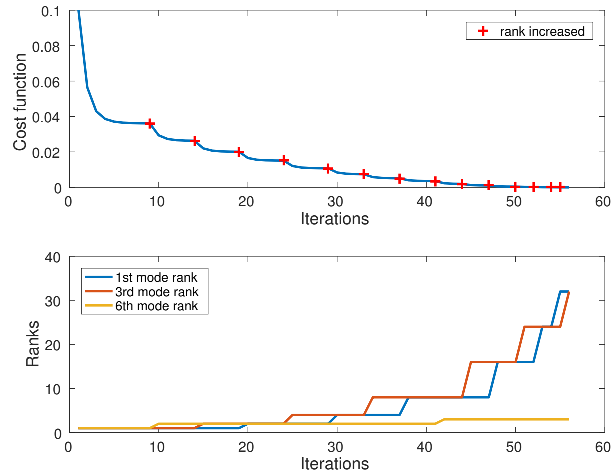

Fig. 6: Convergence behavior

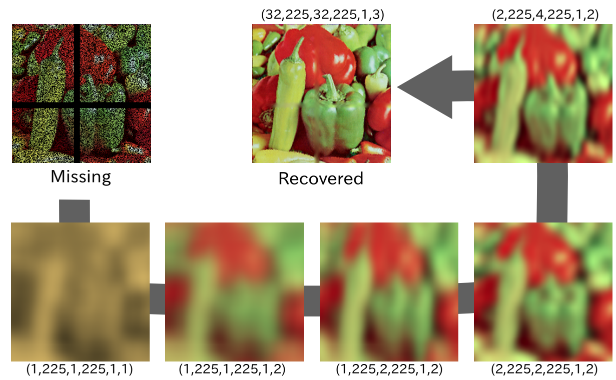

Fig. 7: Optimizing the image by rank increment

Figure 6 shows the convergence behaviors of the cost function and the multilinear tensor ranks. We can observe the monotonically decreasing/non-increasing of the cost function which property is theoretically guaranteed in the auxiliary function method [19], [36]. Red markers put at the points when some mode-rank was increased.

Figure 7 shows the changes of the image in optimization process. We can see that the initial low-rank image was blurred and it became sharper little by little with rank increasing. Finally, missing elements were successfully recovered.

B. Comparison of recovery performance

Next, we compare the recovery performance of the proposed method with the state-of-the-art tensor completion algorithms: low rank tensor completion (LR) [42], total variation method (TV) [68], low-rank and total variation method (LRTV) [68], smooth PARAFAC tensor completion with quadratic variation (SPCQV) [71], and MDT-Tucker [66]. For this comparison,

Algorithm 1 MDT-COS for tensor completion

input: $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I_1 \times \cdots \times J_N}$, $\boldsymbol{\mathcal{Q}} \in \{0,1\}^{I_1 \times \cdots \times I_N}$, $(\tau_1, ..., \tau_N)$, $\{\boldsymbol{d}_1, ..., \boldsymbol{d}_M\}$, $\mathcal{K}$, $\mathcal{R}$, ϵ, tol.
% <u>Step 1: MDT</u>
$\boldsymbol{\mathcal{T}}_H \leftarrow \mathcal{H}(\boldsymbol{\mathcal{T}})$ with $(\tau_1, ..., \tau_N)$;
$\boldsymbol{\mathcal{Q}}_H \leftarrow \mathcal{H}(\boldsymbol{\mathcal{Q}})$ with $(\tau_1, ..., \tau_N)$;
% <u>Step 2: Low-rank Tensor Factorization</u>
$R_m \leftarrow \begin{cases} J_m & m \notin \{\mathcal{K} \cup \mathcal{R}\} \\ \boldsymbol{d}_m(1) & \text{otherwise} \end{cases}$ for $m = \{1, 2, ..., M\}$;
$\boldsymbol{U}^{(m)} = \boldsymbol{I}_{J_m}$ for $m \notin \{\mathcal{K} \cup \mathcal{R}\}$;
$\boldsymbol{U}^{(k)} \leftarrow \boldsymbol{B}_{J_k, R_k}$ for $k \in \mathcal{K}$;
$\boldsymbol{U}^{(r)}$ is initialized by a (J_r, R_r)-random-matrix for each $r \in \mathcal{R}$;
$l_m \leftarrow 1$ for $m = \{1, 2, ..., M\}$;
$\boldsymbol{\mathcal{G}} \in \mathbb{R}^{R_1 \times \cdots \times R_M}$ is initialized by a random-tensor;
$\boldsymbol{\mathcal{X}}_H \leftarrow \boldsymbol{\mathcal{G}} \times \{\boldsymbol{U}\}$;
$f_1 \leftarrow ||\boldsymbol{\mathcal{Q}}_H \circledast (\boldsymbol{\mathcal{T}}_H - \boldsymbol{\mathcal{X}}_H)||_F^2$;
repeat
  $\boldsymbol{\mathcal{Z}} \leftarrow \boldsymbol{\mathcal{Q}}_H \circledast \boldsymbol{\mathcal{T}}_H + \overline{\boldsymbol{\mathcal{Q}}}_H \circledast \boldsymbol{\mathcal{X}}_H$;
  for $r \in \mathcal{R}$ **do**
    $\boldsymbol{\mathcal{Y}} \leftarrow \boldsymbol{\mathcal{Z}} \times_1 \boldsymbol{U}^{(1)T} \times \cdots \times_{r-1} \boldsymbol{U}^{(r-1)T} \times_{r+1} \boldsymbol{U}^{(r+1)T} \times \cdots \times_M \boldsymbol{U}^{(M)T}$;
    $\boldsymbol{U}^{(r)} \leftarrow R_r$ leading singular vectors of $\boldsymbol{Y}_{(r)}$;
  end for
  $\boldsymbol{\mathcal{G}} \leftarrow \boldsymbol{\mathcal{Z}} \times_1 \boldsymbol{U}^{(1)T} \times \cdots \times_M \boldsymbol{U}^{(M)T}$;
  $\boldsymbol{\mathcal{X}}_H \leftarrow \boldsymbol{\mathcal{G}} \times \{\boldsymbol{U}\}$;
  $f_2 \leftarrow ||\boldsymbol{\mathcal{Q}}_H \circledast (\boldsymbol{\mathcal{T}}_H - \boldsymbol{\mathcal{X}}_H)||_F^2$;
  if $|f_2 - f_1| \leq$ tol **then**
    $\boldsymbol{\mathcal{E}} \leftarrow \boldsymbol{\mathcal{Q}}_H \circledast (\boldsymbol{\mathcal{T}}_H - \boldsymbol{\mathcal{X}}_H)$;
    $i \leftarrow \text{argmax}_m ||\boldsymbol{\mathcal{E}} \times_1 \boldsymbol{U}^{(1)T} \times \cdots \times_{m-1} \boldsymbol{U}^{(m-1)T} \times_{m+1} \boldsymbol{U}^{(m+1)T} \times \cdots \times_M \boldsymbol{U}^{(M)T}||_F^2$;
    $l_i \leftarrow l_i + 1$, and $R_i \leftarrow \min(\boldsymbol{d}_i(l_i), J_i)$;
    if $i \in \mathcal{K}$ **then**
      $\boldsymbol{U}^{(i)} \leftarrow \boldsymbol{B}_{J_i, R_i}$;
    end if
    Expand the size of $\boldsymbol{\mathcal{G}}$ to $(R_1, ..., R_M)$ with zero-padding;
  end if
  $f_1 \leftarrow f_2$;
until $f_2 \leq \epsilon$
% <u>Step 3: Inverse MDT</u>
$\boldsymbol{\mathcal{X}} \leftarrow \mathcal{H}^{-1}(\boldsymbol{\mathcal{X}}_H)$;
output: $\boldsymbol{\mathcal{X}}$;

we used the three video data[1]. The summary of data property and parameter settings is given in Table I.

Figure 8 shows the single slice view of resultant videos recovered by using various tensor completion methods. We considered three types of missing: 70% random voxel missing, 95% random voxel missing, and random slice missing. LR only recovered the 70% voxel missing case. TV/LRTV output smoothed videos that the textures of waves/smokes were banished. SPCQV nicely recovered almost all cases but it reduced high frequency components such as edges. Both MDT based methods seemed to be more accurate that the textures and edges are almost recovered, and results were sharper than others.

Table II shows a quantitative comparison by the peak signal-to-noise ratio (PSNR) and the frame average of structured similarity (SSIM) [58]. The highest PSNR $\pm$ 0.2 [dB] and SSIM $\pm$ 0.001 were emphasized by bold font. We can see that both MDT based methods outperformed the other methods in almost all cases. For only ocean with 70% missing, SPCQV was the best with wide difference. For smoke1 with 95% missing, SPCQV was the best but SSIM is very similar to MDT-COS. For smoke2 with 95% missing, SPCQV was the best in PSNR, but MDT-COS was the best in SSIM. Comparing both MDT based methods, the MDT-COS improved or almost similar to MDT-Tucker for all cases.

[1] smoke1 and smoke2 are obtained from NHK CREATIVE LIBRARY (https://www.nhk.or.jp/archives/creative/), and these were down-sampled for the experiments.

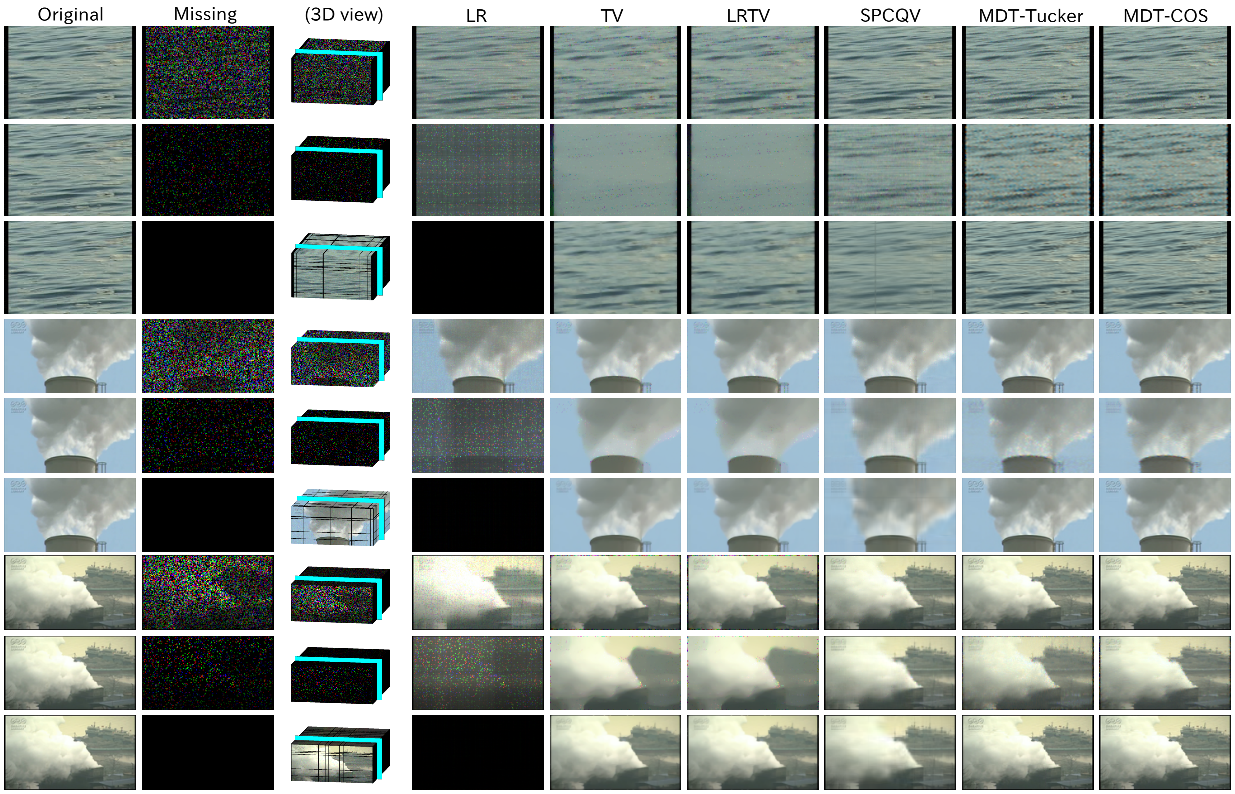

Fig. 8: Slice view of recovered videos.

TABLE II: Comparison with PSNR and SSIM

	LR [42]	TV [68]	LRTV [68]	SPCQV [71]	MDT-Tucker [66]	MDT-COS
ocean, 70% voxel missing	26.7/0.805	26.2/0.759	26.2/0.760	**33.3/0.950**	30.4/0.911	30.5/0.913
ocean, 95% voxel missing	13.0/0.184	20.3/0.289	20.6/0.292	**24.1**/0.603	**24.0**/0.668	**24.2/0.687**
ocean, slice missing	12.2/0.570	30.1/0.904	30.1/0.904	25.9/0.907	30.8/**0.939**	**31.3/0.940**
smoke1, 70% voxel missing	27.5/0.723	35.0/0.952	35.1/0.952	34.7/0.912	**38.6/0.964**	**38.7/0.963**
smoke1, 95% voxel missing	9.0/0.189	27.7/0.832	27.7/0.831	**31.7/0.877**	28.3/0.832	30.7/0.874
smoke1, slice missing	9.9/0.412	40.6/0.985	40.6/0.985	37.6/0.965	**41.7/0.987**	**41.7/0.987**
smoke2, 70% voxel missing	21.0/0.520	26.0/0.934	26.2/0.934	35.1/0.935	41.3/**0.985**	**41.8/0.986**
smoke2, 95% voxel missing	8.5/0.188	19.6/0.765	19.6/0.764	**31.6**/0.891	26.3/0.835	30.0/**0.906**
smoke2, slice missing	9.3/0.403	32.7/0.978	32.9/0.978	33.2/0.960	**40.6/0.985**	**40.4/0.984**

V. Discussion

A. Related works

Filters guided tensor completion approach for image inpainting [37], [43], [44], [59] relates the MDT based methods. The filters guided methods consider the low-rank properties or non-local similarities in the multiple filtered image features. The filters used in the method is decided as priors, and assuming the low-rank structures [59] or non-local similarity [37], [43], [44] in filtered signals.

On the other hand, when we regard the several factor matrices (e.g., related to cosine matrices) in the Hankel tensor as some filters, the other factor matrices and core tensor would construct the filtered signals. Then, the filtered signals are modeled by the low-rank tensor decomposition in the MDT based methods. The MDT based methods explicitly represent the signal in the embedded space, and could consider the structure assumptions of filters, coefficients, and their interactions.

In this way, both methods have some similarities. However, the MDT based methods would have some other possibilities such as learning the data specific filters (shift-invariant features), and modeling the generative system of signals.

B. Computational issue of MDT based tensor completion

There is a computational problem in the MDT based methods. This is because that the MDT expands the N-th order tensor into $2N$-th order tensor, and the size of data increases roughly $(\prod_{n=1}^{N} \tau_n)$-fold. Since τ_n is a key parameter which controls the embedding space, it is not good to set τ_n to

TABLE I: Data setting

name	ocean	smoke1	smoke2
size of $\mathcal{T}$	(112, 160, 3, 32)	(90, 160, 3, 64)	(90, 160, 3, 100)
size of $\mathcal{T}_H$	(8, 105, 8, 153, 1, 3, 8, 25)	(4, 87, 4, 157, 1, 3, 8, 57)	(4, 87, 4, 157, 1, 3, 8, 93)
$\boldsymbol{\tau}$	[8, 8, 1, 8]	[4, 4, 1, 8]	[4, 4, 1, 8]
$\mathcal{K}$	$\{1,3,5,7\}$	$\{1,3,5,7\}$	$\{1,3,5,7\}$
$\mathcal{R}$	$\{2,4,6,8\}$	$\{2,4,6,8\}$	$\{2,4,6,8\}$
$\boldsymbol{d}_1, \boldsymbol{d}_3$	[1, 2, 4, 8]	[1, 2, 4]	[1, 2, 4]
$\boldsymbol{d}_2$	[1, 2, 4, 8, 16, 32, 64, 105]	[1, 2, 4, 8, 16, 32, 64, 87]	[1, 2, 4, 8, 16, 32, 64, 87]
$\boldsymbol{d}_4$	[1, 2, 4, 8, 16, 32, 64, 128, 153]	[1, 2, 4, 8, 16, 32, 64, 128, 157]	[1, 2, 4, 8, 16, 32, 64, 128, 157]
$\boldsymbol{d}_5$	[1]	[1]	[1]
$\boldsymbol{d}_6$	[1, 2, 3]	[1, 2, 3]	[1, 2, 3]
$\boldsymbol{d}_7$	[1, 2, 4, 8]	[1, 2, 4, 8]	[1, 2, 4, 8]
$\boldsymbol{d}_8$	[1, 2, 4, 8, 16, 25]	[1, 2, 4, 8, 16, 32, 57]	[1, 2, 4, 8, 16, 32, 64, 93]

be small. Thus the trade-off relationship exists between the completion ability and the computational times.

VI. Conclusions

In this paper, we proposed a new method of multi-way delay embedding transform based cosine tensor modeling. The proposed model captures the shift-invariant features in visual data, and successfully recovered the missing elements in tensors: random voxels, and slices. The tensor order expansion by MDT causes the computational bottleneck, then it is still a open problem and further study is necessary.

Acknowledgment

This study is supported by THE HORI SCIENCES AND ARTS FOUNDATION.

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] J. A. Bengua, H. N. Phien, H. D. Tuan, and M. N. Do. Efficient tensor completion for color image and video recovery: Low-rank tensor train. *IEEE Transactions on Image Processing*, 26(5):2466–2479, 2017.

[3] J. A. Bengua, H. D. Tuan, H. N. Phien, and M. N. Do. Concatenated image completion via tensor augmentation and completion. In *Proceedings of ICSPCS*, pages 1–7. IEEE, 2016.

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

[5] N. Boumal and P.-a. Absil. Rtrmc: A Riemannian trust-region method for low-rank matrix completion. In *Proceedings of NIPS*, pages 406–414, 2011.

[6] N. Boumal and P.-A. Absil. Low-rank matrix completion via preconditioned optimization on the Grassmann manifold. *Linear Algebra and its Applications*, 475:200–239, 2015.

[7] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *The Journal of Machine Learning Research*, 15(1):1455–1459, 2014.

[8] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[9] J. F. Cai, E. J. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

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

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

[12] 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.

[13] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

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

[15] 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.

[16] L. Condat. A primal–dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *Journal of Optimization Theory and Applications*, 158(2):460–479, 2013.

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

[18] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[19] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (methodological)*, pages 1–38, 1977.

[20] B. Erem, D. E. Hyde, J. M. Peters, F. H. Duffy, and S. K. Warfield. Dynamic electrical source imaging (DESI) of seizures and interictal epileptic discharges without ensemble averaging. *IEEE Transactions on Medical Imaging*, 36(1):98–110, 2017.

[21] B. Erem, R. M. Orellana, D. E. Hyde, J. M. Peters, F. H. Duffy, P. Stovicek, S. K. Warfield, R. S. MacLeod, G. Tadmor, and D. H. Brooks. Extensions to a manifold learning framework for time-series analysis on dynamic manifolds in bioelectric signals. *Physical Review E*, 93(4):042218, 2016.

[22] E. Esser, X. Zhang, and T. Chan. A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM Journal on Imaging Sciences*, 3(4):1015–1046, 2010.

[23] 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.

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

[25] T. Goldstein, M. Li, and X. Yuan. Adaptive primal-dual splitting methods for statistical learning and image processing. In *Proceedings of NIPS*, pages 2089–2097, 2015.

[26] L. Grasedyck, M. Kluge, and S. Kramer. Variants of alternating least squares tensor completion in the tensor train format. *SIAM Journal on Scientific Computing*, 37(5):A2424–A2450, 2015.

[27] F. Guichard and F. Malgouyres. Total variation based interpolation. In *Proceedings of EUSIPCO*, pages 1–4. IEEE, 1998.

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

[29] X. 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.

[30] L. Karlsson and A. Uschmajew. Parallel algorithms for tensor completion in the CP format. *Parallel Computing*, 2015. In press.

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

[32] R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from noisy entries. In *Proceedings of NIPS*, pages 952–960, 2009.

[33] R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, 2010.

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

[35] A. Krishnamurthy and A. Singh. Low-rank matrix and tensor completion via adaptive sampling. In *Proceedings of NIPS*, pages 836–844, 2013.

[36] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Proceedings of NIPS*, pages 556–562, 2001.

[37] S. Li and Q. Liu. Multi-filters guided low-rank tensor coding for image inpainting. In *Proceedings of ICIVC*, pages 418–422, 2017.

[38] X. Li, Y. Ye, and X. Xu. Low-rank tensor completion with total variation for visual data inpainting. In *Proceedings of AAAI*, pages 2210–2216, 2017.

[39] 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.

[40] 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.

[41] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. In *Proceedings of ICCV*, pages 2114–2121. IEEE, 2009.

[42] 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.

[43] H. Lu, S. Li, Q. Liu, and Y. Wang. Mf-lrtc: Multi-filters guided low-rank tensor coding for image restoration. In *Proceedings of ICIP*, pages 2104–2108, 2017.

[44] H. Lu, S. Li, Q. Liu, and M. Zhang. Mf-lrtc: Multi-filters guided low-rank tensor coding for image restoration. *Neurocomputing*, 2018.

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

[46] B. Mishra, K. A. Apuroop, and R. Sepulchre. A Riemannian geometry for low-rank matrix completion. *arXiv preprint arXiv:1211.1550*, 2012.

[47] 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.

[48] B. Mishra, G. Meyer, S. Bonnabel, and R. Sepulchre. Fixed-rank matrix factorizations and Riemannian low-rank optimization. *Computational Statistics*, 29(3-4):591–621, 2014.

[49] E. Ott, T. Sauer, and J. A. Yorke. Coping with chaos. analysis of chaotic data and the exploitation of chaotic systems. *Wiley Series in Nonlinear Science, New York: John Wiley,— c1994, edited by Ott, Edward; Sauer, Tim; Yorke, James A.*, 1994.

[50] P. Rai, Y. Wang, S. Guo, G. Chen, D. Dunson, and L. Carin. Scalable Bayesian low-rank decomposition of incomplete multiway tensors. In *Proceedings of ICML*, pages 1800–1808, 2014.

[51] P. Shah, N. Rao, and G. Tang. Optimal low-rank tensor recovery from separable measurements: Four contractions suffice. *arXiv preprint arXiv:1505.04085*, 2015.

[52] F. Shi, J. Cheng, L. Wang, P.-T. Yap, and D. Shen. Low-rank total variation for image super-resolution. In *Proceedings of MICCAI*, pages 155–162. Springer, 2013.

[53] M. Szummer and R. W. Picard. Temporal texture modeling. In *Proceedings of ICIP*, volume 3, pages 823–826. IEEE, 1996.

[54] 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.

[55] A. Uschmajew and B. Vandereycken. Greedy rank updates combined with Riemannian descent methods for low-rank optimization. In *Proceedings of SampTA*, pages 420–424, 2015.

[56] P. Van Overschee and B. De Moor. Subspace algorithms for the stochastic identification problem. In *Proceedings of CDC*, pages 1321–1326. IEEE, 1991.

[57] N. Vervliet, O. Debals, L. Sorber, and L. De Lathauwer. Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis. *IEEE Signal Processing Magazine*, 31(5):71–79, 2014.

[58] 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.

[59] B. Xiong, Q. Liu, J. Xiong, S. Li, S. Wang, and D. Liang. Field-of-experts filters guided tensor completion. *IEEE Transactions on Multimedia*, 2018.

[60] Y. Xu and W. Yin. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on Imaging Sciences*, 6(3):1758–1789, 2013.

[61] Y. Yamaguchi and K. Hayashi. Tensor decomposition with missing indices. In *Proceedings of IJCAI*, pages 3217–3223. AAAI Press, 2017.

[62] Y. Yang, Y. Feng, and J. A. Suykens. A rank-one tensor updating algorithm for tensor completion. *IEEE Signal Processing Letters*, 22(10):1633–1637, 2015.

[63] T. Yokota and A. Cichocki. Multilinear tensor rank estimation via sparse Tucker decomposition. In *Proceedings of SCIS&ISIS*, pages 478–483. IEEE, 2014.

[64] 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.

[65] T. Yokota and A. Cichocki. Tensor completion via functional smooth component deflation. In *Proceedings of ICASSP*, pages 2514–2518. IEEE, 2016.

[66] T. Yokota, B. Erem, S. Guler, S. K. Warfield, and H. Hontani. Missing slice recovery for tensors using a low-rank model in embedded space. In *Proceedings of CVPR*, pages 8251–8259. IEEE, 2018.

[67] T. Yokota and H. Hontani. An efficient method for adapting step-size parameters of primal-dual hybrid gradient method in application to total variation regularization. In *Proceedings of APSIPA ASC*, pages 973–979, 2017.

[68] 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.

[69] T. Yokota and H. Hontani. Simultaneous tensor completion and denoising by noise inequality constrained convex optimization. *arXiv preprint arXiv:1801.03299*, 2018.

[70] 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.

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

[72] H.-F. Yu, C.-J. Hsieh, S. Si, and I. S. Dhillon. Parallel matrix factorization for recommender systems. *Knowledge and Information Systems*, 41(3):793–819, 2014.

[73] L. Yuan, Q. Zhao, and J. Cao. Completion of high order tensor data with missing entries via tensor-train decomposition. In *Proceedings of ICONIP*, pages 222–229. Springer, 2017.

[74] M. Yuan and C.-H. Zhang. On tensor completion via nuclear norm minimization. *Foundations of Computational Mathematics*, pages 1–38, 2014.

[75] L. Y. Zhao, J. Cao, et al. High-dimension tensor completion via gradient-based optimization under tensor-train format. *arXiv preprint arXiv:1804.01983*, 2018.

[76] 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.

[77] Q. Zhao, L. Zhang, and A. Cichocki. Bayesian sparse tucker models for dimension reduction and tensor completion. *arXiv preprint arXiv:1505.02343*, 2015.

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