Machine Vision and Applications (2016) 27:1259–1273
DOI 10.1007/s00138-016-0806-2

SPECIAL ISSUE PAPER

Pattern recognition in multilinear space and its applications: mathematics, computational algorithms and numerical validations

Hayato Itoh[1] · **Atsushi Imiya**[2] · **Tomoya Sakai**[3]

Received: 30 November 2015 / Revised: 22 July 2016 / Accepted: 13 August 2016 / Published online: 1 September 2016

Abstract We clarify the mathematical equivalence between low-dimensional singular value decomposition and low-order tensor principal component analysis for two- and three-dimensional images. Furthermore, we show that the two- and three-dimensional discrete cosine transforms are, respectively, acceptable approximations to two- and three-dimensional singular value decomposition and classical principal component analysis. Moreover, for the practical computation in two-dimensional singular value decomposition, we introduce the marginal eigenvector method, which was proposed for image compression. For three-dimensional singular value decomposition, we also show an iterative algorithm. To evaluate the performances of the marginal eigenvector method and two-dimensional discrete cosine transform for dimension reduction, we compute recognition rates for six datasets of two-dimensional image patterns. To evaluate the performances of the iterative algorithm and three-dimensional discrete cosine transform for dimension reduction, we compute recognition rates for datasets of gait patterns and human organs. For two- and three-dimensional images, the two- and three-dimensional discrete cosine transforms give almost the same recognition rates as the marginal eigenvector method and iterative algorithm, respectively.

Keywords Dimension reduction · Principal component analysis · Tensor principal component analysis · Singular value decomposition · Discrete cosine transform

✉ Hayato Itoh
hayato-itoh@graduate.chiba-u.jp

Atsushi Imiya
imiya@faculty.chiba-u.jp

Tomoya Sakai
tsakai@cis.nagasaki-u.ac.jp

[1] Graduate School of Advanced Integration Science, Chiba University, Yayoi-cho 1-33, Inage-ku, Chiba 263-8522, Japan

[2] Institute of Management and Information Technologies, Chiba University, Yayoi-cho 1-33, Inage-ku, Chiba 263-8522, Japan

[3] Graduate School of Engineering, Nagasaki University, Bunkyo-cho 1-14, Nagasaki 852-8521, Japan

1 Introduction

In this paper, we clarify the equivalence between Nth-order tensor principal component analysis (TPCA) and N-dimensional singular value decomposition (NDSVD) for $N = 2, 3$. We also show that the NDSVD is theoretically equivalent to classical principal component analysis (PCA), which is based on a vector representation. We refer to the classical PCA for vectors as vector PCA. Furthermore, for the practical computation in the 2DSVD, we introduce marginal eigenvector (MEV) method, which was proposed for image compression in 1981 by Otsu. Moreover, we present an extension of the properties represented in second-order tensors [1] to third-order tensors. These results imply that the TPCA for Nth-order tensors can be approximated by the N-dimensional discrete cosine transform (NDDCT). Table 1 summarises the abbreviations that we use in this paper.

A pattern is assumed to be a square integrable function defined on a finite support in N-dimensional Euclidean space. For planar and volumetric pattern, the dimensions of Euclidean spaces are two and three, respectively. Organs are essentially spatial textures which are functions defined in three-dimensional Euclidean space. Furthermore, for video sequence [2] and volumetric sequence [3], the dimensions of the data spaces are three and four, respectively, since in these applications for planar and spatio-temporal data are focused to analysis. Moreover, planar multichannel images [4,5] are

Table 1 Glossary of abbreviations

SVD	Singular value decomposition
PCA	Principal component analysis
TPCA	Tensor principal component analysis
MEV	Marginal eigenvector
FP	Full projection
FPT	Full projection truncation
2DPCA	Two-dimensional principal component analysis
GPCA	Generalised principal component analysis
2DSVD	Two-dimensional principal component analysis
2DDCT	Two-dimensional discrete cosine transform
ALS	Alternating least squares
NDSVD	N-dimensional singular value decomposition
HOSVD	Higher-order singular value decomposition
MPCA	Multilinear principal component analysis
3DDCT	Three-dimensional discrete cosine transform
NDDCT	N-dimensional discrete cosine transform

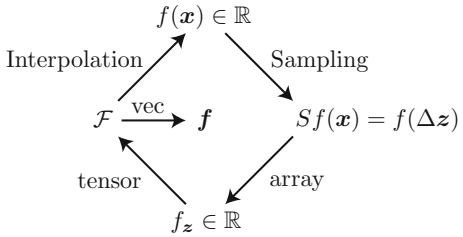

Fig. 1 Sampling, vectors and tensors. The sampled value $f(\Delta \boldsymbol{z})$, $\boldsymbol{z} \in \mathbb{Z}^n$ of a function $f(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^n$ yields an array $f_{\boldsymbol{z}}$, $\boldsymbol{z} \in \mathbb{Z}^{m \times m \times \cdots \times m}$. This array $f_{\boldsymbol{z}}$ is expressed as a tensor $\mathcal{F}$ to preserve multilinearity of $f(\boldsymbol{x})$. Interpolation procedure reconstructs $f(\boldsymbol{x})$ from $Sf(\boldsymbol{x})$ through $\mathcal{F}$. The vector $\boldsymbol{f}$ whose elements are sample values of $f(\boldsymbol{z})$ is constructed from $\mathcal{F}$ by vectorisation operator vec to the tensor $\mathcal{F}$

also expressed as three-way arrays. For these data, elements of two-dimensional array use an additional axis to express frequencies of elements. Multichannel image pattern recognition has been a central issue in remote sensing of earth and planets [6]. In seismic data analysis, the dimension of the data space is five, since waves stated by a planar source array migrated to planar receiver array are focused to analyse [7].

For numerical computation, we deal with sampled patterns. In traditional pattern recognition, these sampled patterns are embedded in an appropriate-dimensional Euclidean space as vectors. The other way is to deal with sampled patterns as higher-dimensional array data. These array data are expressed by tensor to preserve multilinearity of function in the original pattern space. Tensors allow expressing multidimensional array data in multilinear forms. Figure 1 illustrates the relation among sampling, vectors and tensor representation for multilinear structure.

For applications of modern pattern recognition techniques such as deep learning [8] and machine learning for big data [9], we are mathematically and numerically required to evaluate the performance of tensor-based pattern recognition of multilinear data. Importantly, for fast image pattern recognition, a compact representation of these image data is desirable. Tensor expressions fulfil these requirements in applications of pattern recognition of multidimensional array data.

The vector PCA method is a traditional method for data compression. It has been extended to higher-dimensional array data [10,11]. For example, tensor PCA method constructs a small-size tensor using the orthogonal decomposition of a tensor, while the classical PCA (vector PCA) estimates a low-dimensional linear subspace using PCA. Second-order TPCA, which directly decomposes an image matrix, is the tensor PCA method [10,11] for two-dimensional images. A survey [11] reported that there are three basic projections for a tensor. Second- and third-order TPCA use tensor-to-tensor projections consisting of 1- and 2-mode projections, and 1-, 2- and 3-mode projections for two-dimensional and three-dimensional images, respectively.

For image representation, two-dimensional principal component analysis (2DPCA) [12] has been proposed. However, the projection method in the 2DPCA is not a bilinear form since the 2DPCA uses only the 2-mode projection. The MEV method [13], which is based on both 1- and 2-mode projections, has been proposed for image compression. The 2DSVD [14,15], which is also based on both 1- and 2-mode projections, has been proposed for image compression as an extension of the SVD [16]. The projections in the MEV method and the 2DSVD are equivalent to the tensor-to-tensor projection for a second-order tensor. This mathematical property implies that the 2DSVD is a special case of the TPCA. However, the compression rate of the 2DSVD is smaller than that of the 2DDCT [14] for the same reconstruction quality. An iterative algorithm [17] for the second-order TPCA has been proposed. This iterative algorithm is referred to as generalised principal component analysis (GPCA). This GPCA method is a two-dimensional version of the iterative algorithm for the SVD [18,19].

The origin of the TPCA for the third-order tensors was proposed as the decomposition of tensors by Tucker [20]. For the Tucker decomposition of second- and third-order tensors, Kroonenberg and Leeuw [21] discussed the properties of convergence of alternating least squares (ALS) algorithms. In general for Tucker decomposition, orthogonality constraints on decomposed tensors are not required. Cichoki et al. [22] imposed that the existence of the constraints is the difference between the TPCA and parallel factor analysis. In Ref. [22], in addition to orthogonal constraints, sparse constraints and nonnegative constraints for tensor decomposition are studied. For practical computation in higher TPCA, higher-order singular value decomposition (HOSVD) [23] was formulated.

In Refs. [23,24], ALS algorithm for the smaller-size tensor approximation of higher-order tensors was studied. For pattern recognition, there are variants of the TPCA [10,25–27]. Multilinear principal component analysis (MPCA) [10] is an iterative algorithm used for TPCA and is the N-dimensional version of the GPCA. This iterative algorithm is the same as the ALS algorithm. Robust MPCA [25] is a robust version of TPCA for image pattern recognition including outliers. Uncorrelated MPCA [26] searches for a tensor-to-vector projection that obtains most of the variation in the original tensorial input by deciding the maximum number of uncorrelated features. Sparse higher-order principal component analysis [27] searches for the minimum number of bases for input tensors by assuming sparsity in tensor decomposition.

For image pattern recognition, the two-dimensional tensor subspace method (2DTSM) [28], which measures the similarity between an input image and each tensor subspace of a class, has been proposed as extension of the subspace method [29]. The 2DTSM adopts the MEV method to construct each tensor subspace of a class. We extend the 2DTSM for use with three-dimensional images in this paper.

To evaluate the performances of the MEV method and the 2DDCT for the dimension reduction of two-dimensional images, we compute recognition rates for image patterns. To evaluate the performances of the HOSVD and the NDDCT [30] for three-dimensional images, we compute recognition rates for image sequences in gait classification and voxel images of livers in computational anatomy. For the two-dimensional images, the results show that the MEV method and the 2DDCT have almost the same performances in terms of the recognition rates for images in six datasets. Furthermore, for the three-dimensional images, the results show that the HOSVD and the NDDCT have almost the same performances in terms of the recognition rates for images in a dataset of gait patterns.

The 2DDCT-II is used for the coding in MPEG. In MPEG, a digital image of $n \times n$ pixels is first partitioned to $N \times N$ blocks. Usually, the size of each block is 8×8 pixels. Then, each $N \times N$ block is transformed to the frequency domain using the 2DDCT-II. Finally, each transformed value in each block is encoded. If we use MPEG for a sequence of two-dimensional images, that is, for a tensor with size $N \times N \times N$, we apply N times partitioning with block and encoding with the 2DDCT for N frames. Note that we use the 2DDCT and 3DDCT without partitioning of two- and three-dimensional images, respectively, in this paper.

2 Tensor projection for images

We briefly summarise the multilinear projection for multi-dimensional arrays from Ref. [11]. A Nth-order tensor $\mathcal{X}$ defined in $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is expressed as

$$\mathcal{X} = \left(x_{i_1,i_2,\ldots,i_N}\right) \tag{1}$$

for $x_{i_1,i_2,\ldots,i_N} \in \mathbb{R}$, using N indices i_n. Each subscript n denotes the n-mode of $\mathcal{X}$. For the outer products of N vectors, if the tensor $\mathcal{X}$ satisfies the condition

$$\mathcal{X} = \boldsymbol{u}^{(1)} \circ \boldsymbol{u}^{(2)} \circ \cdots \circ \boldsymbol{u}^{(N)}, \tag{2}$$

where $\circ$ denotes the outer product, we call this tensor $\mathcal{X}$ a rank-one tensor. For $\mathcal{X}$, the n-mode vectors, $n = 1, 2, \ldots, N$, are defined as the I_n-dimensional vectors obtained from $\mathcal{X}$ by varying this index i_n while fixing all the other indices. The unfolding of $\mathcal{X}$ along the n-mode vectors of $\mathcal{X}$ is defined as

$$\mathcal{X}_{(n)} \in \mathbb{R}^{I_n \times (I_1 \times I_2 \times \cdots I_{n-1} \times I_{n+1} \times \cdots \times I_N)}, \tag{3}$$

where the column vectors of $\mathcal{X}_{(n)}$ are the n-mode vectors of $\mathcal{X}$. Figure 2 shows two examples of n-mode unfolding for second- and third-order tensors. The n-mode product $\mathcal{X} \times_n \boldsymbol{U}$ of a matrix $\boldsymbol{U} \in \mathbb{R}^{J_n \times I_n}$ and a tensor $\mathcal{X}$ is a tensor $\mathcal{G} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N}$, with elements

$$g_{i_1,i_2,\ldots,i_{n-1},j_n,i_{n+1},\ldots,i_N} = \sum_{i_n=1}^{I_n} x_{i_1,i_2,\ldots,I_N} u_{j_n,i_n} \tag{4}$$

by the manner in Ref. [22]. We define the inner product of two tensors $\mathcal{X} = (x_{i_1,i_2,\ldots,i_N}), \mathcal{Y} = (y_{i_1,i_2,\ldots,i_N}) \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ by

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} x_{i_1,i_2,\ldots,i_N} y_{i_1,i_2,\ldots,i_N}. \tag{5}$$

Using this inner product, the Frobenius norm of a tensor $\mathcal{X}$ is

$$\|\mathcal{X}\|_{\mathrm{F}} = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}. \tag{6}$$

For the Frobenius norm of a tensor, we have

$$\|\mathcal{X}\|_{\mathrm{F}} = \|\mathrm{vec}\, \mathcal{X}\|_2, \tag{7}$$

where vec and $\|\cdot\|_2$ are the vectorisation operator and Euclidean norm of a tensor, respectively. For the two tensors $\mathcal{X}_1$ and $\mathcal{X}_2$, we define the distance between them as

$$d\left(\mathcal{X}_1, \mathcal{X}_2\right) = \|\mathcal{X}_1 - \mathcal{X}_2\|_{\mathrm{F}}. \tag{8}$$

Although this definition is a tensor-based measure, this distance is equivalent to the Euclidean distance between the vectorised tensors $\mathcal{X}_1$ and $\mathcal{X}_2$.

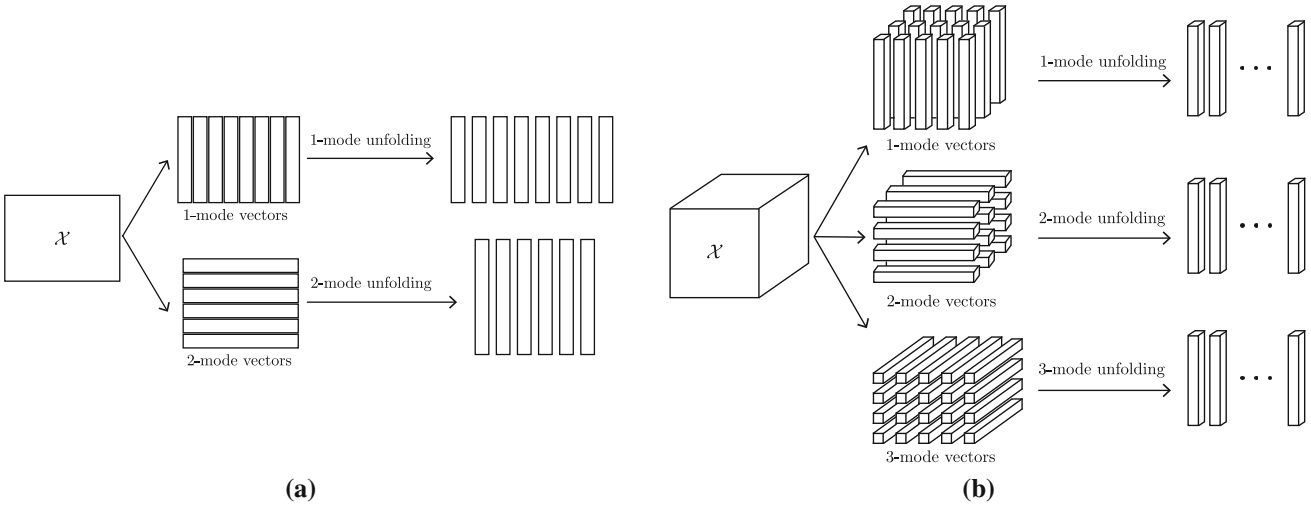

Fig. 2 Unfoldings of second- and third-order tensors. **a** 1- and 2-mode unfoldings of a second-order tensor $\mathcal{X} \in \mathbb{R}^{6\times 8}$. **b** 1-, 2- and 3-mode unfoldings of the third-order tensor $\mathcal{X} \in \mathbb{R}^{4\times 5\times 3}$

For a tensor, a multilinear projection maps the input tensor data from one space to another space. We have three basic multilinear projections, that is, the vector-to-vector projection (VVP), tensor-to-vector projection (TVP) and tensor-to-tensor projection (TTP). The VVP is a linear projection from a vector to another vector. To use the VVP for tensors, we need to reshape tensors into vectors before the projection. The TVP, which is also referred to as the rank-one projection [31,32], consists of elementary multilinear projections (EMPs). An EMP projects a tensor to a scalar. Using d EMPs, the TVP obtains a d-dimensional vector projected from a tensor. The TTP projects a tensor to another tensor of the same order. In this paper, we focus on methods of finding the optimal projection for the TTP.

As the tensor $\mathcal{X}$ is in the tensor space $\mathbb{R}^{I_1}\otimes\mathbb{R}^{I_2}\otimes\cdots\otimes\mathbb{R}^{I_N}$, the tensor space can be interpreted as the Kronecker product of N vector spaces $\mathbb{R}^{I_1}, \mathbb{R}^{I_2}, \ldots, \mathbb{R}^{I_N}$. To project $\mathcal{X} \in \mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2} \otimes \cdots \otimes \mathbb{R}^{I_N}$ to another tensor $\mathcal{Y}$ in a lower-dimensional tensor space $\mathbb{R}^{P_1} \otimes \mathbb{R}^{P_2} \otimes \cdots \otimes \mathbb{R}^{P_N}$, where $P_n \leq I_n$ for $n = 1, 2, \ldots, N$, we need N matrices $\{\boldsymbol{U}^{(n)} \in \mathbb{R}^{I_n\times P_n}\}_{n=1}^{N}$. Using the N matrices, the TTP is given by

$$\mathcal{Y} = \mathcal{X} \times_1 \boldsymbol{U}^{(1)\top} \times_2 \boldsymbol{U}^{(2)\top} \cdots \times_N \boldsymbol{U}^{(N)\top}. \tag{9}$$

This projection is established in N steps, where at the nth step, each n-mode vector is projected to a P_n-dimensional space by $\boldsymbol{U}^{(n)}$. Figures 3a and 4a show the steps for the projection of second- and third-order tensors to lower-dimensional tensors, respectively. Figures 3b, c and 4b–d show the procedures used to project second- and third-order tensors, respectively.

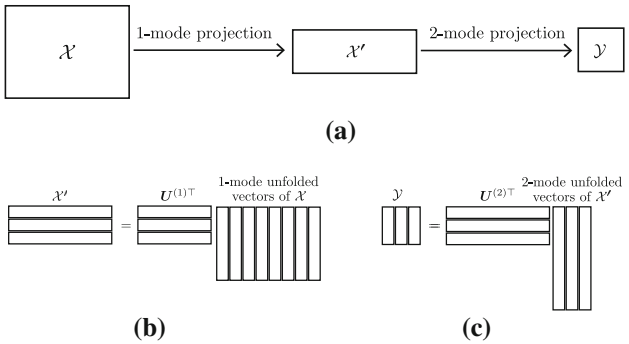

Fig. 3 Tensor-to-tensor projection of a second-order tensor $\mathcal{X} \in \mathbb{R}^{6\times 8}$ to a lower-dimensional tensor $\mathcal{Y} \in \mathbb{R}^{3\times 3}$. **a** Tensor-to-tensor projection for a tensor $\mathcal{X}$. **b** 1-mode projection for $\mathcal{X}$ represented by a linear projection. **c** 2-mode projection for $\mathcal{X}$ represented by a linear projection

3 Decomposition of tensors

3.1 Two-dimensional singular value decomposition

A second-order tensor $\mathcal{X} \in \mathbb{R}^{I_1\times I_2}$, which is the matrix $\boldsymbol{X} = (x_{i_1,i_2}) \in \mathbb{R}^{I_1\times I_2}$, is denoted as a pair of indices (i_1, i_2). For a tensor $\mathcal{X}$, the unfolding of $\mathcal{X}$ is defined by

$$\mathcal{X}_{(1)} = \boldsymbol{X} \in \mathbb{R}^{I_1\times I_2}, \quad \mathcal{X}_{(2)} = \boldsymbol{X}^\top \in \mathbb{R}^{I_2\times I_1}. \tag{10}$$

The 1- and 2-mode products of a tensor by a matrix $\boldsymbol{U}^\top$ are given by

$$\mathcal{X} \times_1 \boldsymbol{U}^\top = \boldsymbol{U}^\top \boldsymbol{X}, \quad \mathcal{X} \times_2 \boldsymbol{U}^\top = \boldsymbol{U}^\top \boldsymbol{X}^\top, \tag{11}$$

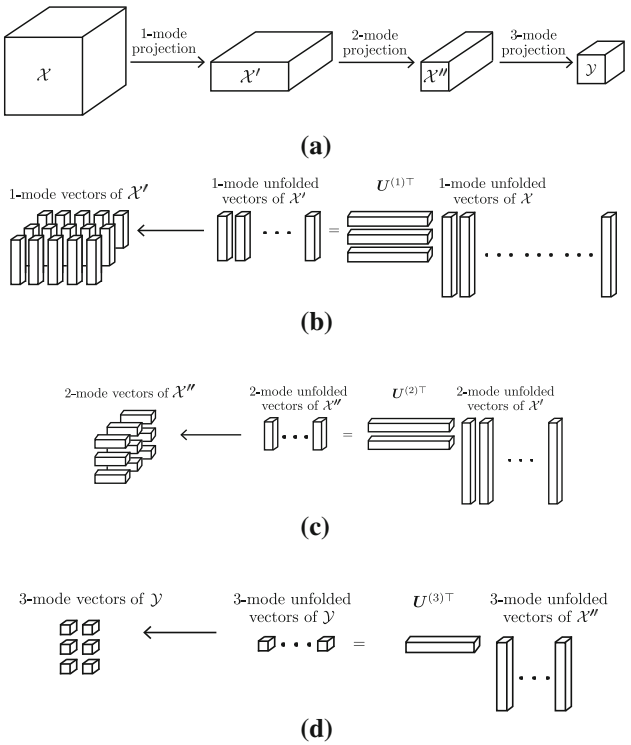

Fig. 4 Tensor-to-tensor projection of a third-order tensor $\mathcal{X} \in \mathbb{R}^{4\times5\times3}$ to a lower-dimensional tensor $\mathcal{Y} \in \mathbb{R}^{3\times2\times1}$. **a** Tensor-to-tensor projection for a tensor $\mathcal{X}$. **b** 1-mode projection for $\mathcal{X}$ represented by a linear projection. **c** 2-mode projection for $\mathcal{X}$ represented by a linear projection. **d** 3-mode projection for $\mathcal{X}$ represented by a linear projection

respectively. For a tensor $\mathcal{X}$ in the tensor space $\mathbb{R}^{I_1} \otimes \mathbb{R}^{I_2}$, using two matrices, we have the TTP

$$\mathcal{Y} = \mathcal{X} \times_1 \boldsymbol{U}^{(1)\top} \times_2 \boldsymbol{U}^{(2)\top}, \tag{12}$$

which projects $\mathcal{X}$ to a lower-dimensional tensor space. Replacing the tensor $\mathcal{X}$ with $m = I_1$ and $n = I_2$, and a matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$, we have the TTP

$$\boldsymbol{Y} = \boldsymbol{U}^{(1)\top} \boldsymbol{X} \boldsymbol{U}^{(2)} \tag{13}$$

for the second-order tensor with a matrix representation. From the reduced matrix $\boldsymbol{Y}$, we have the reconstruction given by

$$\boldsymbol{X} = \boldsymbol{U}^{(1)} \boldsymbol{Y} \boldsymbol{U}^{(2)\top}. \tag{14}$$

For a collection of matrices $\{\boldsymbol{X}_i\}_{i=1}^N \in \mathbb{R}^{m\times n}$ satisfying with zero expectation condition $\mathrm{E}(\boldsymbol{X}_i) = 0$, the orthogonal projection-based data reduction

$$\hat{\boldsymbol{X}}_i = \boldsymbol{U}^\top \boldsymbol{X}_i \boldsymbol{V}, \tag{15}$$

where $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m]$ and $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n]$, is performed by minimising the criterion

$$J_- = \mathrm{E}\left(\|\boldsymbol{X}_i - \boldsymbol{U}\hat{\boldsymbol{X}}_i\boldsymbol{V}^\top\|_{\mathrm{F}}^2\right) \tag{16}$$

and maximising the criteria

$$J_+ = \mathrm{E}\left(\left\|\boldsymbol{U}^\top \boldsymbol{X}_i \boldsymbol{V}\right\|_{\mathrm{F}}^2\right), \tag{17}$$

$$J_V = \mathrm{E}\left(\left\|\boldsymbol{V}^\top \boldsymbol{X}_i^\top \boldsymbol{X}_i \boldsymbol{V}\right\|_{\mathrm{F}}^2\right), \tag{18}$$

$$J_U = \mathrm{E}\left(\left\|\boldsymbol{U}^\top \boldsymbol{X}_i \boldsymbol{X}_i^\top \boldsymbol{U}\right\|_{\mathrm{F}}^2\right), \tag{19}$$

with respect to the conditions

$$\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}_m \quad \text{and} \quad \boldsymbol{V}^\top \boldsymbol{V} = \boldsymbol{I}_n, \tag{20}$$

where $\boldsymbol{I}_m$ and $\boldsymbol{I}_n$ are the identity matrices in $\mathbb{R}^{m\times m}$ and $\mathbb{R}^{n\times n}$, respectively.

Eigen decomposition problems are derived by computing the extremals of

$$E_- = J_- + tr\left((\boldsymbol{I} - \boldsymbol{V}^\top \boldsymbol{V})\boldsymbol{\Lambda}\right) + tr\left((\boldsymbol{I} - \boldsymbol{U}^\top \boldsymbol{U})\boldsymbol{\Sigma}\right), \tag{21}$$

$$E_+ = J_+ + tr\left((\boldsymbol{I} - \boldsymbol{V}^\top \boldsymbol{V})\boldsymbol{\Lambda}\right) + tr\left((\boldsymbol{I} - \boldsymbol{U}^\top \boldsymbol{U})\boldsymbol{\Sigma}\right), \tag{22}$$

$$E_V = J_V + tr\left((\boldsymbol{I} - \boldsymbol{V}^\top \boldsymbol{V})\boldsymbol{\Lambda}\right), \tag{23}$$

$$E_U = J_U + tr\left((\boldsymbol{I} - \boldsymbol{U}^\top \boldsymbol{U})\boldsymbol{\Sigma}\right). \tag{24}$$

For covariant matrices $\boldsymbol{M} = \frac{1}{N}\sum_{i=1}^N \boldsymbol{X}\boldsymbol{X}^\top$ and $\boldsymbol{N} = \frac{1}{N}\sum_{i=1}^N \boldsymbol{X}^\top\boldsymbol{X}$, the optimisation of J_- and J_+ derives the eigenvalue decomposition

$$\boldsymbol{M}\boldsymbol{U} = \boldsymbol{U}\boldsymbol{\Sigma} \quad \text{and} \quad \boldsymbol{N}\boldsymbol{V} = \boldsymbol{V}\boldsymbol{\Lambda}, \tag{25}$$

where $\boldsymbol{\Sigma} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{\Lambda} \in \mathbb{R}^{n\times n}$ are diagonal matrices satisfying the relationships $\lambda_i = \sigma_i$ and $\mathrm{rank}(\boldsymbol{M}) = \mathrm{rank}(\boldsymbol{N}) = K$ for

$$\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_K, 0 \ldots, 0), \tag{26}$$

$$\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_K, 0 \ldots, 0). \tag{27}$$

The optimisation of J_V and J_U derives the eigendecomposition problems in Eq. (25).[1] Using a set of orthonormal vectors $\{\boldsymbol{e}_k\}_{k=1}^K$, where only kth element of $\boldsymbol{e}_k$ is 1 and others are 0, we set orthogonal projection matrices $\boldsymbol{P}_1 = \sum_{k=1}^{k_1} \boldsymbol{e}_k\boldsymbol{e}_k^\top$ and $\boldsymbol{P}_2 = \sum_{k=1}^{k_2} \boldsymbol{e}_k\boldsymbol{e}_k^\top$. Using these $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$, the low-rank matrix approximation [33,34] is achieved by

[1] For an iterative method, see Refs. [18,19]. These iterative algorithms are a special case of the HOSVD [23].

$$\boldsymbol{Y}_i = (\boldsymbol{P}_1\boldsymbol{U})^\top \boldsymbol{X}_i\,(\boldsymbol{P}_2\boldsymbol{V}) = \boldsymbol{L}^\top \boldsymbol{X}_i \boldsymbol{R}, \tag{28}$$

where $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ select k_1 and k_2 bases of orthogonal matrices $\boldsymbol{U}$ and $\boldsymbol{V}$, respectively. The low-rank approximation using Eq. (28) is called the 2DSVD method in the context of image compression [14,15]. Moreover, the method based on the transform

$$\boldsymbol{Y}_i = \boldsymbol{X}_i \boldsymbol{R} \tag{29}$$

is called the 2DPCA [12]. This 2DPCA proposed by Yan and Zhang is not bilinear form. Therefore, this 2DPCA is not the TPCA for second-order tensors.

For the 2DSVD, we have the following theorem.

Theorem 1 *The 2DSVD method is equivalent to the vector PCA method.*

Proof The equation

$$(\boldsymbol{P}_1\boldsymbol{U})^\top \boldsymbol{X}(\boldsymbol{P}_2\boldsymbol{V}) = \boldsymbol{Y} \tag{30}$$

is equivalent to

$$(\boldsymbol{P}_2\boldsymbol{V} \otimes \boldsymbol{P}_1\boldsymbol{U})\,\mathrm{vec}\boldsymbol{X} = \mathrm{vec}\boldsymbol{Y}. \tag{31}$$

□

Furthermore, the 2DDCT is an acceptable approximation of the 2DSVD since the 2DDCT is an acceptable approximation of the PCA for the reduction of images [1,28,29]. Moreover, the projection that selects $K = k_1 k_2$ bases of the tensor space spanned by $\boldsymbol{u}_i \otimes \boldsymbol{v}_j$, $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$, is

$$(\boldsymbol{P}_2\boldsymbol{V} \otimes \boldsymbol{P}_1\boldsymbol{U}) = (\boldsymbol{P}_2 \otimes \boldsymbol{P}_1)\,(\boldsymbol{V} \otimes \boldsymbol{U}) = \boldsymbol{P}\boldsymbol{W}, \tag{32}$$

where $\boldsymbol{W}$ and $\boldsymbol{P}$ are an orthogonal matrix and a projection matrix, respectively. Therefore, the 2DSVD is equivalent to the TPCA for matrices because matrices are second-order tensors. In our application, an $n \times n$ digital array is directly compressed by the 2DDCT-II with order $\mathcal{O}(n^2)$. If we apply the fast Fourier transform to the computation of the 2DDCT-II, the computational complexity is $\mathcal{O}(n \log n)$.

3.2 Three-dimensional singular value decomposition

A third-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, which is the array $\boldsymbol{X} = (x_{i_1,i_2,i_3}) \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, is denoted as a triple of indices (i_1, i_2, i_3). Here we summarise the HOSVD for third-order tensors. For a collection of tensors $\{\mathcal{X}_i\}_{i=1}^N \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ satisfying the zero expectation condition $\mathrm{E}(\mathcal{X}_i) = 0$, we compute the

$$\hat{\mathcal{X}}_i = \mathcal{X}_i \times_1 \boldsymbol{U}^{(1)\top} \times_2 \boldsymbol{U}^{(2)\top} \times_3 \boldsymbol{U}^{(3)\top}, \tag{33}$$

where $\boldsymbol{U}^{(j)} = [\boldsymbol{u}_1^{(j)}, \ldots, \boldsymbol{u}_{I_j}^{(j)}]$, that minimises the criterion

$$J_- = \mathrm{E}\left(\|\mathcal{X}_i - \hat{\mathcal{X}}_i \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \times_3 \boldsymbol{U}^{(3)}\|_{\mathrm{F}}^2\right) \tag{34}$$

and maximises the criteria

$$J_+ = \mathrm{E}\left(\left\|\hat{\mathcal{X}}_i\right\|_{\mathrm{F}}^2\right), \tag{35}$$

with respect to the conditions

$$\boldsymbol{U}^{(j)\top}\boldsymbol{U}^{(j)} = \boldsymbol{I}_j, \tag{36}$$

where $\boldsymbol{I}_j$, $j = 1, 2, 3$, is the identity matrices in $\mathbb{R}^{I_j \times I_j}$. For this criterion, by fixing two of $\boldsymbol{U}^{(1)}$, $\boldsymbol{U}^{(2)}$ and $\boldsymbol{U}^{(3)}$, we have the following criteria

$$J_j = \mathrm{E}\left(\left\|\boldsymbol{U}^{(j)\top}\mathcal{X}_{i,(j)}\mathcal{X}_{i,(j)}^\top\boldsymbol{U}^{(j)}\right\|_{\mathrm{F}}^2\right), \tag{37}$$

where $\mathcal{X}_{i,(j)}$, $j = 1, 2, 3$, are the j-mode unfolded tensor $\mathcal{X}_i$, with respect to Eq. (36).

Eigendecomposition problems are derived by computing the extremal of

$$E_- = J_- + \sum_{j=1}^{N} tr((\boldsymbol{I}_j - \boldsymbol{U}^{(j)\top}\boldsymbol{U}^{(j)})\boldsymbol{\Sigma}^{(j)}). \tag{38}$$

As an extension of the two-dimensional problem, we define the system of minimisation problems

$$E_j = J_j + tr((\boldsymbol{I}_j - \boldsymbol{U}^{(j)\top}\boldsymbol{U}^{(j)})\boldsymbol{\Sigma}^{(j)}),\ j = 1, 2, 3. \tag{39}$$

For matrices $\boldsymbol{M}^{(j)} = \frac{1}{N}\sum_{i=1}^N \mathcal{X}_{i,(j)}\mathcal{X}_{i,(j)}^\top$, $j = 1, 2, 3$, the optimisation of J_- derives the eigenvalue decomposition

$$\boldsymbol{M}^{(j)}\boldsymbol{U}^{(j)} = \boldsymbol{U}^{(j)}\boldsymbol{\Sigma}^{(j)}, \tag{40}$$

where $\boldsymbol{\Sigma}^{(j)} \in \mathbb{R}^{I_j \times I_j}$, $j = 1, 2, 3$, are diagonal matrices satisfying the relationships $\sigma_k^{(j)} = \sigma_k^{(j')}$, $k \in \{1, 2, \ldots, K\}$, $K = \mathrm{rank}(\boldsymbol{M}^{(1)}) = \mathrm{rank}(\boldsymbol{M}^{(2)}) = \mathrm{rank}(\boldsymbol{M}^{(3)})$ for

$$\boldsymbol{\Sigma}^{(j)} = \mathrm{diag}\left(\lambda_1^{(j)}, \lambda_2^{(j)}, \ldots, \lambda_K^{(j)}, 0 \ldots, 0\right). \tag{41}$$

The optimisation of each J_j derives the eigendecomposition problems in Eq. (40). However, for the optimisation of $\{J_j\}_{j=1}^3$, there is no closed-form solution to this maximisation problem [23,24]. Algorithm 1 is the iterative procedure of the MPCA. For Algorithm 1, we have the following property.

Property 1 *The MPCA without iteration is equivalent to the HOSVD if dimensions of a projected tensor are coincident with ones of the modes of the original tensor.*

Algorithm 1: Iterative method in the MPCA (ALS algorithm)

Input: A set of tensors $\{\mathcal{X}_i\}_{i=1}^N$. Dimension of projected tensors $\{k_j\}_{j=1}^3$. A maximum number of iteration K. An order ξ_1, ξ_2, ξ_3 to select the unfolded tensors. A sufficiently small number η.

Output: A set of projection matrices $\{\boldsymbol{U}^{(j)}\}_{j=1}^3$.

1: Compute the eigendecomposition of a covariant matrix $\boldsymbol{M}^{(j)} = \frac{1}{N}\sum_{i=1}^N \mathcal{X}_{i,(j)}\mathcal{X}_{i,(j)}^\top$, where $\mathcal{X}_{i,(j)}$ is an j-mode unfolded $\mathcal{X}_i$, for $j = 1, 2, 3$.

2: Construct projection matrices by selecting eigenvectors corresponding to the k_j largest eigenvalues for $j = 1, 2, 3$.

3: Compute $\Psi_0 = \sum_{i=1}^N \|\mathcal{X}_i \times_{\xi_1} \boldsymbol{U}^{(\xi_1)\top} \times_{\xi_2} \boldsymbol{U}^{(\xi_2)\top} \times_{\xi_3} \boldsymbol{U}^{(\xi_3)\top}\|_{\mathrm{F}}$.

4: Iteratively compute the following procedure.

for $k = 1, 2, \ldots, K$

for $j = \xi_1, \xi_2, \xi_3$

Update $\boldsymbol{U}^{(j)}$ by decomposing a covariant matrix $\boldsymbol{M}^{(j)} = \sum_{i=1}^N \mathcal{W}_{i,(j)}\mathcal{W}_{i,(j)}^\top$, where $\mathcal{W}_{i,(j)}$ is an j-mode unfolded $\mathcal{W}_i = \mathcal{X}_i \times_{\xi_\alpha} \mathbf{U}^{(\xi_\alpha)\top} \times_{\xi_\beta} \mathbf{U}^{(\xi_\beta)\top}$ with $\xi_\alpha, \xi_\beta \in \{\xi_1, \xi_2, \xi_3\} \setminus j, \xi_\alpha \neq \xi_\beta$.

end

Compute $\Psi_k = \sum_{i=1}^N \|\mathcal{X}_i \times_{\xi_1} \boldsymbol{U}^{(\xi_1)\top} \times_{\xi_2} \boldsymbol{U}^{(\xi_2)\top} \times_{\xi_3} \boldsymbol{U}^{(\xi_3)\top}\|_{\mathrm{F}}$

if $|\Psi_k - \Psi_{k-1}| < \eta$

break

end

For third-order tensors, there are 3! combinations in selecting the order of modes ξ_1, ξ_2 and ξ_3 in a tensor-to-tensor projection for Algorithm 1. On the other hand, one combination exists in selecting the order of modes for the second-order tensors.

Property 2 *For third-order tensors, the selection of order of modes does not effect to the results of a tensor-to-tensor projection.*

From these two properties, we adopt Algorithm 1 [10] to solve the optimisation of $\{J_j\}_{j=1}^3$. For a set of orthonormal vectors $\{\boldsymbol{e}_k\}_{k=1}^K$, where only kth element of $\boldsymbol{e}_k$ is 1 and others are 0, we set orthogonal projection matrices $\boldsymbol{P}^{(j)} = \sum_{k=1}^{k_j} \boldsymbol{e}_k\boldsymbol{e}_k^\top$ for $j = 1, 2, 3$. Using these $\{\boldsymbol{P}^{(j)}\}_{j=1}^3$, the low-rank tensor approximation [24] is achieved by

$$\mathcal{Y} = \mathcal{X} \times_1 \left(\boldsymbol{P}^{(1)}\boldsymbol{U}^{(1)}\right)^\top \times_2 \left(\boldsymbol{P}^{(2)}\boldsymbol{U}^{(2)}\right)^\top \times_3 \left(\boldsymbol{P}^{(3)}\boldsymbol{U}^{(3)}\right)^\top, \tag{42}$$

where $\boldsymbol{P}^{(j)}$ selects k_j bases of orthogonal matrices $\boldsymbol{U}^{(j)}$. The low-rank approximation using Eq. (42) is used for compression in the TPCA.

For the HOSVD for third-order tensors, we have the following theorem.

Theorem 2 *The HOSVD method is equivalent to the vector PCA method.*

Proof The equation

$$\mathcal{X} \times_1 \left(\boldsymbol{P}^{(1)}\boldsymbol{U}^{(1)}\right)^\top \times_2 \left(\boldsymbol{P}^{(2)}\boldsymbol{U}^{(2)}\right)^\top \times_3 \left(\boldsymbol{P}^{(3)}\boldsymbol{U}^{(3)}\right)^\top = \mathcal{Y} \tag{43}$$

is equivalent to

$$\left(\boldsymbol{P}^{(3)}\boldsymbol{U}^{(3)} \otimes \boldsymbol{P}^{(2)}\boldsymbol{U}^{(2)} \otimes \boldsymbol{P}^{(1)}\boldsymbol{U}\right)^\top \mathrm{vec}\mathcal{X} = \mathrm{vec}\mathcal{Y}. \tag{44}$$

□

This theorem implies that the 3DDCT is an acceptable approximation of the HOSVD for third-order tensors since this is the analogy of the approximation of the PCA by the 2DDCT [29].

Furthermore, we have the following theorem.

Theorem 3 *The compression computed by the HOSVD is equivalent to the compression computed by the TPCA.*

Proof The projection that selects $K = k_1k_2k_3$ bases of the tensor space spanned by $u_{i_1}^{(1)} \circ u_{i_2}^{(2)} \circ u_{i_3}^{(3)}$, $i_j = 1, 2, \ldots, k_j$ for $j = 1, 2, 3$, is

$$\begin{aligned}&\left(\boldsymbol{P}^{(3)}\boldsymbol{U}^{(3)} \otimes \boldsymbol{P}^{(2)}\boldsymbol{U}^{(2)} \otimes \boldsymbol{P}^{(1)}\boldsymbol{U}^{(1)}\right)\\ &= \left(\boldsymbol{P}^{(3)} \otimes \boldsymbol{P}^{(2)} \otimes \boldsymbol{P}^{(1)}\right)\left(\boldsymbol{U}^{(3)} \otimes \boldsymbol{U}^{(2)} \otimes \boldsymbol{U}^{(1)}\right) = \boldsymbol{P}\boldsymbol{W},\end{aligned} \tag{45}$$

where $\boldsymbol{W}$ and $\boldsymbol{P}$ are an orthogonal matrix and a projection matrix, respectively. Therefore, HOSVD is equivalent to TPCA for third-order tensors. □

In our application, an $n \times n \times n$ digital array is directly compressed by the 3DDCT-II with order $\mathcal{O}(n^3)$. If we apply the fast Fourier transform to the computation of the 3DDCT-II, the computational complexity is $\mathcal{O}(n \log n)$.

4 Tensor subspace method

4.1 Second-order tensors

As an extension of the subspace method for vector data, we introduced a linear tensor subspace method for a matrix called the 2DTSM [28]. For a matrix $\boldsymbol{X}$, setting $\boldsymbol{P}_{\mathrm{L}}$ and $\boldsymbol{P}_{\mathrm{R}}$ to be orthogonal projections, we call the operation

$$\boldsymbol{Y} = \boldsymbol{P}_{\mathrm{L}}^\top \boldsymbol{X} \boldsymbol{P}_{\mathrm{R}} \tag{46}$$

the orthogonal projection of $\boldsymbol{X}$ to $\boldsymbol{Y}$. Therefore, using this expression for a collection of matrices $\{\boldsymbol{X}_i\}_{i=1}^N$, such that $\boldsymbol{X}_i \in \mathbb{R}^{m\times n}$ and $\mathrm{E}(\boldsymbol{X}_i) = 0$, the solutions of

$$\begin{aligned}(\boldsymbol{P}_{\mathrm{L}}, \boldsymbol{P}_R) = \arg\max \mathrm{E}\left(\frac{\|\boldsymbol{P}_{\mathrm{L}}^\top \boldsymbol{X}_i \boldsymbol{P}_{\mathrm{R}}\|_{\mathrm{F}}}{\|\boldsymbol{X}_i\|_{\mathrm{F}}}\right)\\ w.r.t.\ \boldsymbol{P}_{\mathrm{L}}^\top \boldsymbol{P}_{\mathrm{L}} = \boldsymbol{I},\ \boldsymbol{P}_{\mathrm{R}}^\top \boldsymbol{P}_{\mathrm{R}} = \boldsymbol{I}\end{aligned} \tag{47}$$

define a bilinear subspace that approximates $\{\boldsymbol{X}_i\}_{i=1}^N$. Here, the norm $\|\boldsymbol{X}\|_{\mathrm{F}}$ for matrix $\boldsymbol{X}$ represents the Frobenius norm. Therefore, using the solutions of Eq. (47), if a query matrix $\boldsymbol{G}$ satisfies the condition

$$\arg\left(\max_i \frac{\|\boldsymbol{P}_{\mathrm{L},i}^\top \boldsymbol{G}\boldsymbol{P}_{\mathrm{R},i}\|_{\mathrm{F}}}{\|\boldsymbol{G}\|_{\mathrm{F}}}\right) = \{\boldsymbol{P}_{\mathrm{L},k}, \boldsymbol{P}_{\mathrm{R},k}\}, \tag{48}$$

we conclude that $\boldsymbol{G} \in \mathcal{C}_k(\delta)$, $k = 1, 2, \ldots, N_{\mathcal{C}}$ for $N_{\mathcal{C}}$ categories if

$$\mathcal{C}_k(\delta) = \left\{\boldsymbol{X} \mid \|P_{\mathrm{L},k}^\top \boldsymbol{X} P_{\mathrm{R},k} - \boldsymbol{X}\|_{\mathrm{F}} \ll \delta\right\}. \tag{49}$$

In practical computation to find the projections P_{L} and P_{R} in Eq. (47), we adopt the MEV [13] method. This is a projection considering the distributions of column and row vectors of sampled images. We define two matrices

$$\boldsymbol{M}_{\mathrm{r}} = \frac{1}{N}\sum_{i=1}^N \boldsymbol{X}_i \boldsymbol{X}_i^\top \in \mathbb{R}^{m\times m}, \tag{50}$$

$$\boldsymbol{M}_{\mathrm{c}} = \frac{1}{N}\sum_{i=1}^N \boldsymbol{X}_i^\top \boldsymbol{X}_i \in \mathbb{R}^{n\times n}. \tag{51}$$

Using these two matrices, we have

$$\begin{aligned}\mathrm{E}\left(\|\boldsymbol{P}_{\mathrm{L}}^\top \boldsymbol{X}_i \boldsymbol{P}_{\mathrm{R}}\|_{\mathrm{F}}^2\right) &= \frac{1}{N}\sum_{i=1}^N \left(\boldsymbol{P}_{\mathrm{L}}^\top \boldsymbol{X}_i \boldsymbol{P}_{\mathrm{R}}\right)\left(\boldsymbol{P}_{\mathrm{L}}^\top \boldsymbol{X}_i \boldsymbol{P}_{\mathrm{R}}\right)^\top \\ &= \boldsymbol{P}_{\mathrm{L}}^\top \boldsymbol{M}_{\mathrm{r}} \boldsymbol{P}_{\mathrm{L}}, \end{aligned}\tag{52}$$

$$\begin{aligned}\mathrm{E}\left(\|\boldsymbol{P}_{\mathrm{L}}^\top \boldsymbol{X}_i \boldsymbol{P}_{\mathrm{R}}\|_{\mathrm{F}}^2\right) &= \frac{1}{N}\sum_{i=1}^N \left(\boldsymbol{P}_{\mathrm{L}}^\top \boldsymbol{X}_i \boldsymbol{P}_{\mathrm{R}}\right)^\top \left(\boldsymbol{P}_{\mathrm{L}}^\top \boldsymbol{X}_i \boldsymbol{P}_{\mathrm{R}}\right) \\ &= \boldsymbol{P}_{\mathrm{R}}^\top \boldsymbol{M}_{\mathrm{c}} \boldsymbol{P}_{\mathrm{R}}. \end{aligned}\tag{53}$$

Furthermore, for the two matrices $\boldsymbol{M}_{\mathrm{r}}$ and $\boldsymbol{M}_{\mathrm{c}}$, using the Lagrange multipliers Λ_{r} and Λ_{c}, we find projections satisfying

$$J(\boldsymbol{P}_{\mathrm{L}}) = tr\left(\boldsymbol{P}_{\mathrm{L}}^\top \boldsymbol{M}_{\mathrm{r}} \boldsymbol{P}_{\mathrm{L}}\right) - tr\left(\left(\boldsymbol{P}_{\mathrm{L}}^\top \boldsymbol{P}_{\mathrm{L}} - \boldsymbol{I}\right)\Lambda_{\mathrm{r}}\right), \tag{54}$$

$$J(\boldsymbol{P}_Y) = tr\left(\boldsymbol{P}_{\mathrm{R}}^\top \boldsymbol{M}_{\mathrm{c}} \boldsymbol{P}_{\mathrm{R}}\right) - tr\left(\left(\boldsymbol{P}_{\mathrm{R}}^\top \boldsymbol{P}_{\mathrm{R}} - \boldsymbol{I}\right)\Lambda_{\mathrm{c}}\right), \tag{55}$$

where $\boldsymbol{I}$ is the identity matrix. The solutions of Eqs. (54) and (55) are given as the solutions of the eigenproblems of $\boldsymbol{M}_{\mathrm{r}}$ and $\boldsymbol{M}_{\mathrm{c}}$, respectively. We set $\{\boldsymbol{u}_j\}_{j=1}^{k_1}$ and $\{\boldsymbol{v}_j\}_{j=1}^{k_2}$ as the eigenvectors of $\boldsymbol{M}_{\mathrm{r}}$ and $\boldsymbol{M}_{\mathrm{c}}$, respectively. We define the eigenvectors of $\boldsymbol{M}_{\mathrm{r}}$ and $\boldsymbol{M}_{\mathrm{c}}$ as $\|\boldsymbol{u}_j\|_2 = 1$ and $\|\boldsymbol{v}_j\|_2 = 1$ for eigenvalues $\lambda_1^{\mathrm{r}} \geq \lambda_2^{\mathrm{r}} \geq \cdots \geq \lambda_j^{\mathrm{r}} \geq \cdots \geq \lambda_n^{\mathrm{r}}$ and $\lambda_1^{\mathrm{c}} \geq \lambda_2^{\mathrm{c}} \geq \cdots \geq \lambda_j^{\mathrm{c}} \geq \cdots \geq \lambda_n^{\mathrm{c}}$, respectively. Therefore, for given numbers $k_1 \leq m$ and $k_2 \leq n$, the operators $\boldsymbol{P}_{\mathrm{L}}$ and $\boldsymbol{P}_{\mathrm{R}}$ are defined as $\boldsymbol{P}_{\mathrm{L},k_1} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{k_1}]$ and $\boldsymbol{P}_{\mathrm{R},k_2} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{k_2}]$ as the matrices consist of each set of eigenfunctions, respectively. These obtained projections are equivalent to the projections obtained by 2DSVD [14] using Eq. (28). Practically, the computational complexity of solving the eigendecomposition problem for an $n \times n$-matrix is $\mathcal{O}(n^3)$.

4.2 Third-order tensors

As an extension of the subspace method for vector data, we introduce a new linear tensor subspace method for third-order tensors. This method is a three-dimensional version of the 2DTSM [28]. For a third-order tensor $\mathcal{X}$, setting $\boldsymbol{P}_j$, $j = 1, 2, 3$, to be orthogonal projections, we call the operation

$$\mathcal{Y} = \mathcal{X}_i \times_1 \boldsymbol{P}_1^\top \times_2 \boldsymbol{P}_2^\top \times_3 \boldsymbol{P}_3^\top \tag{56}$$

the orthogonal projection of $\mathcal{X}$ to $\mathcal{Y}$. Therefore, using this expression for a collection of matrices $\{\mathcal{X}_i\}_{i=1}^M$, such that $\mathcal{X}_i \in \mathbb{R}^{I_1\times I_2\times I_e}$ and $\mathrm{E}(\mathcal{X}_i) = 0$, the solutions of

$$\begin{aligned}\{P_j\}_{j=1}^3 = \arg\max \mathrm{E}&\left(\frac{\|\mathcal{X} \times_1 \boldsymbol{P}_1^\top \times_2 \boldsymbol{P}_2^\top \times_3 \boldsymbol{P}_3^\top\|_{\mathrm{F}}}{\|\mathcal{X}_i\|_{\mathrm{F}}}\right) \\ &w.r.t.\ \ \boldsymbol{P}_j^\top \boldsymbol{P}_j = \boldsymbol{I}\ \ for\ j = 1, 2, 3\end{aligned}\tag{57}$$

define a trilinear subspace that approximates $\{\mathcal{X}_i\}_{i=1}^M$. Here, the norm $\|\mathcal{X}\|_{\mathrm{F}}$ for matrix $\mathcal{X}$ represents the Frobenius norm. Therefore, using the solutions of Eq. (57), if a query tensor $\mathcal{G}$ satisfies the condition

$$\arg\left(\max_i \frac{\left\|\mathcal{G} \times_1 \boldsymbol{P}_{k,1}^\top \times_2 \boldsymbol{P}_{k,2}^\top \times_3 \boldsymbol{P}_{k,3}^\top\right\|_{\mathrm{F}}}{\|\mathcal{G}\|_{\mathrm{F}}}\right) = \{\boldsymbol{P}_{k,j}\}_{j=1}^3, \tag{58}$$

we conclude that $\mathcal{G} \in \mathcal{C}_k(\delta)$, $k = 1, 2, \ldots, N_{\mathcal{C}}$ for $N_{\mathcal{C}}$ categories if

$$\mathcal{C}_k(\delta) = \left\{\mathcal{X} \mid \|\mathcal{G} \times_1 \boldsymbol{P}_{k,1}^\top \times_2 \boldsymbol{P}_{k,2}^\top \times_3 \boldsymbol{P}_{k,3}^\top - \mathcal{X}\|_{\mathrm{F}} \ll \delta\right\}. \tag{59}$$

For the practical computation of projections $\{\boldsymbol{P}_{k,j}\}_{j=1}^3$, we adopt the iterative method described in Algorithm 1.

5 Numerical examples

5.1 Second-order tensors

To validate the relation between the 2DDCT and the MEV method (2DSVD), we compute the recognition rate using

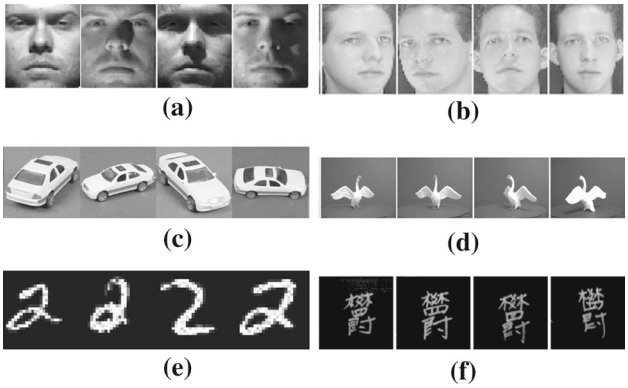

Fig. 5 Examples of images belonging to the same class in each dataset. **a** Face images of the same person with different conditions of illumination. **b** Face images of the same person with different camera positions. **c** Images of the same object with different camera positions, where the number of degrees of freedom is two. **d** Images of the same object with different camera positions, where the number of degrees of freedom is one. **e** Images of the same handwritten digit. **f** Images of the same handwritten Chinese character written by different people

Table 2 Sizes and number of images in each dataset

	#class	#data/class	Image size [pixel]	Reduced image size [pixel]
YaleB	38	64	192 × 168	32 × 32
ORL	40	10	112 × 92	32 × 32
ETH80	30	41	128 × 128	32 × 32
NEC	60	72	480 × 580	32 × 32
MNIST	10	7000	28 × 28	15 × 15
ETL9G	152	200	127 × 128	32 × 32

♯class and ♯data/class represent the number of classes and the number of data in each class, respectively. The image size is the original size of the images in each dataset. The reduced image size is the size of the images after image representation-based dimension reduction

six image datasets: cropped versions of the extended YaleB dataset [35], ORL face dataset [36], ETH80 dataset [4], NEC animal dataset [37], MNIST dataset [38] and ETL9G character dataset [39]. Figure 5 shows examples of images belonging to the same class in each dataset. For the validation, we compress images in these datasets by the MEV method and the 2DDCT. Table 2 shows the sizes and the number of images in each dataset and the parameters used in the compression.

Using the compressed images belonging to the same class in each dataset, we compute the cumulative contribution ratio (CCR) for the eigenvalues of the covariance matrices of the 1- and 2-modes for the images.

Figure 6 shows the CCRs for the six datasets. In Fig. 6a–f, we define the compression ratio as $32^2/k^2$ for the case of using $k \in \{1, 2, \ldots, 32\}$ eigenvectors, corresponding to the k largest eigenvalues of each mode to construct the linear subspace. As shown in Fig. 6a–d, the CCR curves for the MEV method and the 2DDCT are coincident. In Fig. 6b, the CCR curves for the 1- and 2-modes are coincident for the MEV method and the 2DDCT. This means that the eigenvalues of the 1- and 2-modes are coincident. In Fig. 6a, c–f, the CCR curves are approximately the same except for a few of the largest eigenvalues.

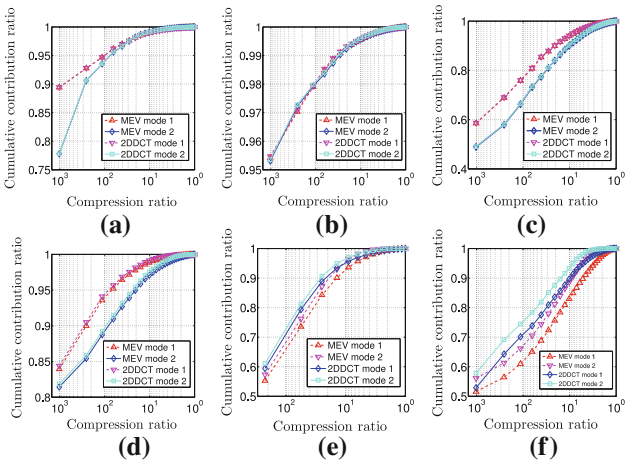

Fig. 6 Cumulative contribution ratios for the six datasets. The *vertical* and *horizontal axes* represent the cumulative contribution ratio and the compression ratio, respectively. For the reduced size $1024 = 32 \times 32$ and the reduced size $K = k^2$ for the case of selecting the k highest eigenvalues, the compression ratio is given as D/K. *Upward triangles* and *diamonds* represent the cumulative contribution ratios of the eigenvalues of the covariance matrices for the 1- and 2-modes, respectively, of the images compressed by the MEV method. *Downward triangles* and *squares* represent the cumulative contribution ratios of the eigenvalues of the covariance matrices for the 1- and 2-modes, respectively, of the images compressed by the 2DDCT. **a** YaleB, **b** ORL, **c** ETH80, **d** NEC, **e** MNIST, **f** ETL9G

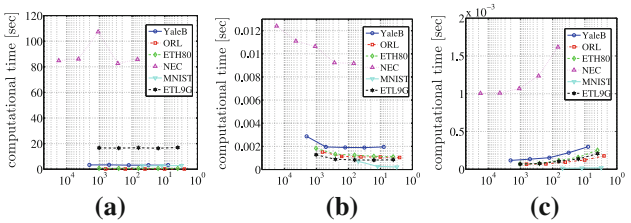

Fig. 7 Computational time of dimension reduction for tensors of the second order. **a**, **b** the computational time of construction of projection matrices for the MEV and 2DDCT, respectively. **c** The mean computational time of projecting images to low-dimensional tensor space for six datasets. In **a**–**c**, the *vertical* and *horizontal axes* represent the computational time and compression ratio, respectively. For the original reduced size D shown in Table 2 and the reduced size $K = k_1 \times k_2$, the compression ratio is given as D/K. Here, we adopt $k_1, k_2 \in \{4, 8, 16, 32, 64\}$

Figure 7 summarises the computational time of the dimension reduction for six datasets. For computation, we use Intel Xeon X5570 Quad core 2.93 GHz. For computation of 2DDCT, we construct three DCT-II matrices without FFT. Note that the computational times of TTP in the MEV and

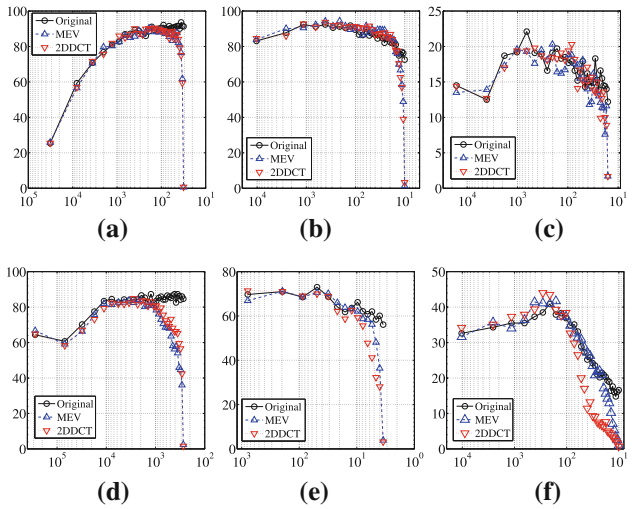

Fig. 8 Recognition rates for original and compressed images. The vertical and horizontal axes represent the recognition rate [%] and compression ratio, respectively. For the first reduced size D and the second reduced size $K = k^2$, where $k \in \{1, 2, \ldots, 32\}$ is the number of selected bases, the compression ratio is given as D/K. *Circles*, *upward triangles* and *downward triangles* show the recognition rates for the original dimension, the MEV method and the 2DDCT, respectively. **a** YaleB, **b** ORL, **c** ETH80, **d** NEC, **e** MNIST, **f** ETL9G

the 2DDCT are the same. In Fig. 7, computational time of the 2DDCT is smaller than one of the MEV.

For the validation, we compute the recognition rate using the original and dimension-reduced images of the six datasets with the 2DTSM as the classifier. The MNIST dataset is predivided into training and test data before it is distributed. For the YaleB, ORL, ETH80 and NEC datasets, images labelled with even numbers are used as training data and the other images are used as test data. The recognition rate is defined as the successful label estimation ratio for 1000 label estimations. In each estimation of a label for a query, queries are randomly chosen from the test data. For both the 1- and 2-modes, we evaluated the results for linear subspaces with one to 32 dimensions.

Figure 8a–f shows the recognition rates for the original and compressed images in the six datasets. In Fig. 8, we define the compression ratio as the original dimension divided by k^2, where k is the number of selected principal axes for both the 1- and 2-modes. According to these results, the MEV method and the 2DDCT give almost the same recognition rate.

5.2 Third-order tensors

To validate the relation between the HOSVD and the 3DDCT, we compute recognition rates using the OU-ISIR dataset [2] and the computational anatomy (CA) dataset. Figure 9 shows examples of sequences of silhouette images from two different categories in the OU-ISIR dataset. Figure 10

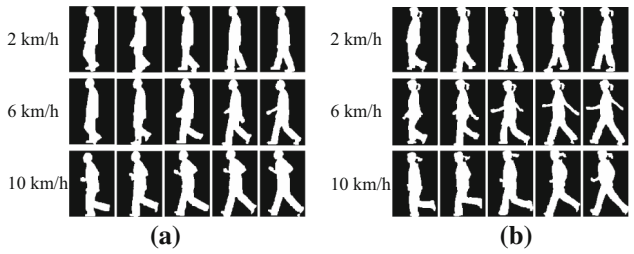

Fig. 9 Examples of sequences of silhouette images. These are binary images whose pixel values are 0 or 255. The figure illustrates the 1st, 21st, 41st, 61st, 81st silhouette images of sequences from a person walking at different speeds. Each sequence consists of 90 silhouette images of four steps. For each sequence, we manually selected the start and finish of the sequence from the original OU-ISIR treadmill dataset. Each sequence is obtained by resampling of the selected sequence with linear interpolation, where the linear interpolation is only used for mode 3. **a** Person ♯001. **b** Person ♯128

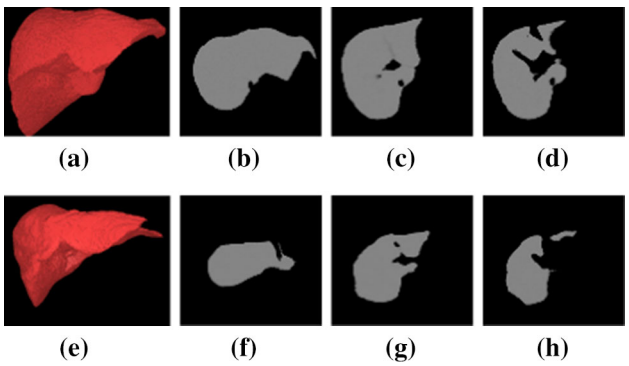

Fig. 10 Examples of the voxel images of human livers in CA dataset. **a**, **e** show the volume rendering of shape of livers for male and female, respectively. These rendering is computed from CT images of livers. **b–d**, and **f–h** are axial CT images used in the rendering for **a** and **e**, respectively

shows examples of voxel images of human livers in the CA datasets. In Fig. 10, voxel images are rendered using the computed tomography (CT) images of human livers. Table 3 summarises the sizes of tensors of the two datasets. For the compression of the silhouette-image sequences, we use the HOSVD and the 3DDCT. For the practical computation of the HOSVD, we use the iterative method described in Algorithm 1 [10]. If we set the number of iterations to 0 in Algorithm 1, we have the three-dimensional version of the MEV method. If we set the number of bases to the size of the original tensors in Algorithm 1, we call the method full projection (FP). If we set the number of bases to less than the size of the original tensors in Algorithm 1, we call the method full projection truncation (FPT).

Firstly, we observe the properties of the iterative method. Using sequences of silhouette images from a category, we compute the sum of the energies of projected tensors after k iterations and the CCR of the eigenvalues for the three modes. For the computation of tensor products, we select different

Table 3 Sizes and number of tensors of the resampled OU-ISIR and CA datasets

	♯class	♯data/class	Tensor size	Reduced tensor size
OU-ISIR	34	9	$128 \times 88 \times 90$	$d \times d \times d$
CA	2	25 (male)	$89 \times 97 \times 76$	$d \times d \times d$
		7 (female)		

♯class and ♯data/class represent the number of classes and the number of data in each class, respectively. That CA dataset consists of two classes, that is, male and female. The tensor size is the size of the dataset before dimension reduction. The reduced tensor size is the size of the tensor after dimension reduction. We set $d \in \{32, 16, 8\}$ for the size in the dimension reduction

orders of selection of the modes. Since there are three modes, we have $3! = 6$ orders. For FPT, we set the numbers of bases for each mode to $64 \times 64 \times 64$ and $32 \times 32 \times 32$. Figure 11 shows the sum of the energies Ψ_k after every 10 iterations for the FP and FPT with the six different orders of computation of the tensor projection. Figure 12 shows the CCRs of the three modes in the FP for the different orders of computation of the tensor projections. Figure 13 summarises the CCRs of the three modes for projections to the three different sizes. Figure 14 shows the CCRs of each mode for FP and FPT.

In Fig. 11, the iterations do not significantly affect the sum of the energies of the projected tensors. According to both Figs. 11 and 12, changing the order of computation of the tensor projections does not give different results. In Fig. 13, the CCRs of the three decompositions are almost coincident in the three modes. Figure 14 shows that the eigenvalues obtained by FP and FPT are not coincident. The decomposition of tensors for the FPT gives larger eigenvalues with a smaller number of bases than those obtained by FP.

Next, we compute the CCRs of the eigenvalues obtained by 10 iterations of Algorithm 1 for compressed tensors. For the compression of the tensors from $128 \times 88 \times 90$ to $32 \times 32 \times 32$, we adopt the FP, the FPT and the 3DDCT. Figure 15 shows the CCRs of each mode for the three types of compressed tensor. The tensors compressed by the 3DDCT give larger eigenvalues than those compressed by the FP and the FPT with a smaller number of bases. The FP and the FPT gives the same CCRs for each mode.

Thirdly, we compute the recognition rate of the sequences of silhouette images by the TSM. In this validation, we use the original sizes of the tensors and compressed tensors for comparison. For the compression, we adopt the HOSVD, the FP, the FPT and the 3DDCT. Using these four methods, we compress the tensors to the sizes $32 \times 32 \times 32$, $16 \times 16 \times 16$ and $8 \times 8 \times 8$. The OU-ISIR dataset contains sequences of images of 34 people with nine different walking speeds. We use the sequences with walking speeds of 2, 4, 6, 8 and 10 km/h for learning data and the sequences with walking speeds of 3, 5, 7 and 9 km/h for test data. The recognition rate is defined as the successful label estimation ratio for 1000 label estimations. In each estimation of a label for a query, queries are randomly chosen from the test dataset. For the 1-, 2- and 3-modes, we

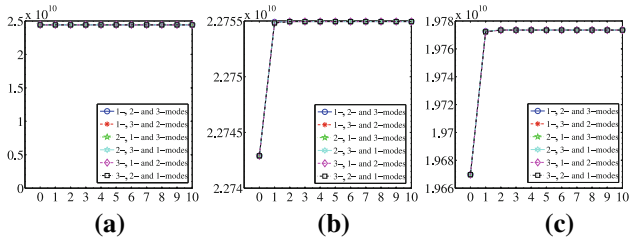

Fig. 11 Convergence of iteration described in Algorithm 1. **a–c** The sum of energies Ψ_k in each iteration for the given numbers of bases of $128 \times 88 \times 90$, $64 \times 64 \times 64$ and $32 \times 32 \times 32$, respectively. In the **a–c**, *horizontal* and *vertical axes* represent the number of iterations and Ψ_k, respectively. For the computation of the tensor projections using Algorithm 1, we adopt the six orders of selection of the modes, where the legends in the figures summarises the six orders

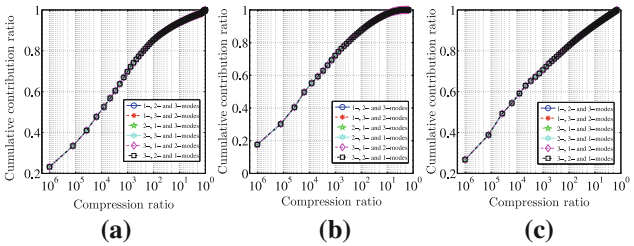

Fig. 12 Cumulative contribution ratio of eigenvalues obtained by 10 iterations using Algorithm 1. **a–c** The cumulative contribution ratios for the 1-, 2- and 3-modes, respectively. Here, the given number of bases is $128 \times 88 \times 90$ for Algorithm 1. For the computation of the tensor projections using Algorithm 1, we adopt six orders of the selection of modes, where the legends in the figures summarises the six orders. The *horizontal* and *vertical axes* represent the compression ratio and cumulative contribution ratio, respectively. For the original size $D = 128 \times 88 \times 90$ and the reduced size $K = k \times k' \times k''$, the compression ratio is given as D/K

evaluate the results for multilinear subspaces with sizes from one to the dimension of the compressed tensors.

Figure 16a–c shows the recognition rates for the four compression methods with three different sizes of the compressed tensors of OU-ISIR dataset. For the images of size $32 \times 32 \times 32$ shown in Fig. 16a, the recognition rates for all four types of compressed tensor are almost coincident with those of the original tensors, if the compression ratio is higher than 10^3. If the compression ratio is less than 10^3, the recognition ratio of the FPT is higher than those of the

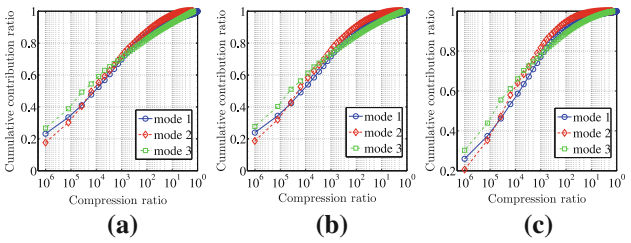

Fig. 13 Cumulative contribution ratio of three modes. **a**–**c** The cumulative contribution ratios of the three modes for the input sizes $128 \times 88 \times 90$, $64 \times 64 \times 64$ and $34 \times 34 \times 34$ in Algorithm 1, respectively. The *horizontal* and *vertical axes* represent the compression ratio and cumulative contribution ratio, respectively. For the original size $D = 128 \times 88 \times 90$ and the reduced size $K = k \times k' \times k''$, the compression ratio is given as D/K

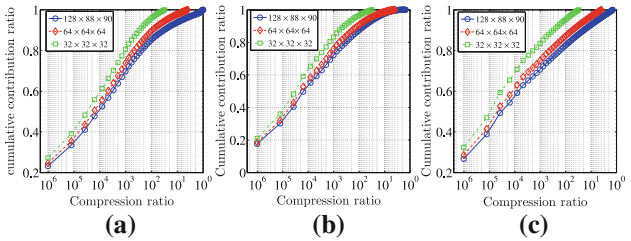

Fig. 14 Comparison of cumulative contribution ratio between full projection and full projection truncation. **a**–**c** A comparison of the cumulative contribution ratio for the 1-, 2- and 3-modes, respectively. The *horizontal* and *vertical axes* represent the compression ratio and cumulative contribution ratio, respectively. For the original size $D = 128 \times 88 \times 90$ and the reduced size $K = k \times k' \times k'$, the compression ratio is given as D/K

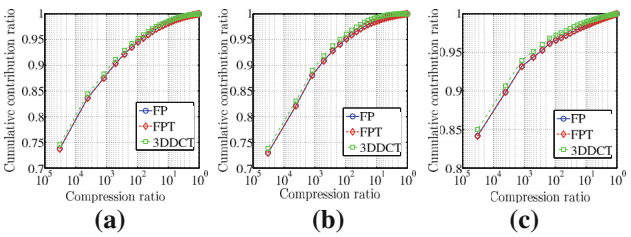

Fig. 15 Comparison of cumulative contribution ratio for three types of compressed tensor. For the compression of tensors, we use Algorithm 1 and the 3DDCT. In Algorithm 1, we respectively adopt sizes of $128 \times 88 \times 90$ and $32 \times 32 \times 32$ for the computation by FP and FPT. For the three types of compressed tensor of $32 \times 32 \times 32$, we apply 10 iterations of Algorithm 1. **a**–**c** The cumulative contribution ratios for the 1-, 2- and 3-modes, respectively. In **a**–**c**, the *horizontal* and *vertical axes* represent the compression ratio and cumulative contribution ratio, respectively. For the first reduced size $K_1 = 32 \times 32 \times 32$ and the second reduced size $K_2 = k \times k' \times k''$, the compression ratio is given as K_1/K_2

HOSVD, the FP and the 3DDCT. The recognition ratio of the 3DDCT is lower than those of other methods since the silhouette images are binary images. For the images of sizes $16 \times 16 \times 16$ and $8 \times 8 \times 8$ shown in Fig. 16b, c, although the recognition rates for the four types of compressed tensor are almost the same, the recognition rates are smaller than those for the original tensors. This recognition property

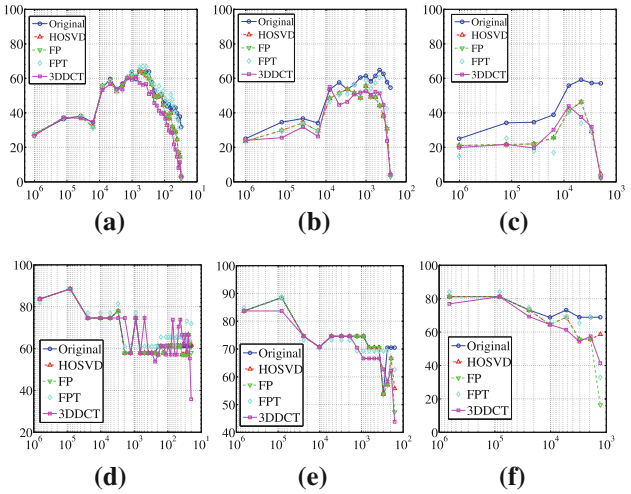

Fig. 16 Recognition rates of gait patterns and liver data for original and compressed tensors. We adopt the reduces sizes of $32 \times 32 \times 32$, $16 \times 16 \times 16$ and $8 \times 8 \times 8$. **a**–**c** and **d**–**f** The recognition rates for three reduced sizes of OU-ISIR and CA datasets, respectively. For compression, we use the HOSVD, FP, FPT and 3DDCT. In **a**–**f**, the *horizontal* and *vertical axes* represent the compression ratio and recognition ratio [%], respectively. In **a**–**c** and **d**–**f**, for the original reduced sizes $D = 128 \times 88 \times 90$ and $D = 89 \times 97 \times 76$, respectively the compression ratio is given as D/K for reduced size k

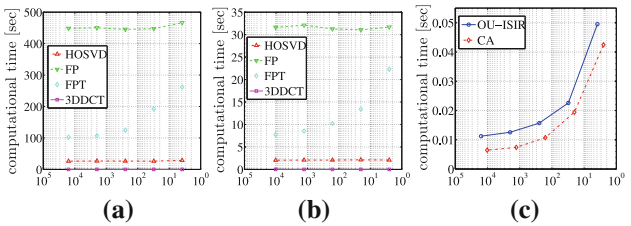

Fig. 17 Computational time of dimension reduction for tensors of the order three. **a**, **b** The computational time of construction of projection matrices for 306 sequences of silhouette images and 35 voxel images of livers, respectively. **c** The mean computational time of projecting images to low-dimensional tensor space for OU-ISIR and CA datasets. In **a**, **b**, we compare the HOSVD, FP, FPT and 3DDCT. In **a**–**c**, the *vertical* and *horizontal axes* represent the computational time and compression ratio, respectively

depends on the size of the images, and the images used for the comparison are too small to evaluate our methods of the recognition of sequences of silhouette images. In Fig. 16a–c, the HOSVD and the FP give the same recognition rate. These results imply that the decomposition for the FP is independent of the number of iterations.

Fourthly, we compute the recognition rate of the voxel images of human livers by the TSM. For the computation, we use the CA dataset. The CA dataset contains livers of 25 males and 7 females. We use the voxel images of livers of 13 males and 4 females as training data. The residual voxel images are used as test data. In the recognition, we estimate the gender of livers. The recognition rate is defined as the successful estimation ratio for 1000 gender estimations. In

each estimation of a gender for a query, queries are randomly chosen from the test dataset. For the 1-, 2- and 3-modes, we evaluate the results for multilinear subspaces with sizes from one to the dimension of the compressed tensors.

Figure 16d–f shows the recognition rates for the four compression methods with three different sizes of the compressed tensors. For the voxel images of sizes $32 \times 32 \times 32$ and $16 \times 16 \times 16$ shown in Fig. 16d, e, the recognition rates for all four types of compressed tensor are almost coincident with those of the original tensors. For the voxel images of sizes $8 \times 8 \times 8$ shown in Fig. 16f, the recognition rates for all four types of compressed tensor are almost coincident with those of the original tensors, if the compression ratio is higher than 10^4. Compared to the recognition rates of the sequences of silhouette images, the 3DDCT gives better approximation for ones of the volume data of livers since livers are essentially volumetric objects, which consists of textures of liver tissues.

Finally, using Intel Xeon X5570 Quad core 2.93 GHz, we compare the computational times of the four reduction methods for OU-ISIR data and CA data. In the comparison, for TPCA, we set 3 iterations for Algorithm 1. For computation of 3DDCT, we construct three DCT-II matrices. Therefore, we do not use FFT. Note that the computational times of TTP in the four methods are the same. Therefore, we show only the mean of the computational time. Figure 17 shows the comparison of computational times for two datasets. In both cases for OU-ISIR and CA datasets, 3DDCT gives the fastest construction of projection matrices in the four methods.

For voxel images, the 3DDCT gives an acceptable approximation of the HOSVD, the FP and the FPT in the context of pattern recognition. Even for sequences of binary silhouette images, the 3DDCT gives an acceptable approximation of the HOSVD, the FP and the FPT in the context of pattern recognition. Moreover, from Figs. 14, 15, and 16a–c, changes in the energies of the projected tensors and the CCRs of the eigenvalues in the decomposition of tensors in these methods are not important in the context of pattern recognition.

6 Conclusions

We first clarified the equivalence between Nth-order tensor principal component analysis and N-dimensional singular value decomposition for $N = 2, 3$. Second, we introduced the marginal eigenvalue method and the iterative algorithm as practical computation methods for two- and three-dimensional singular value decomposition, respectively. Furthermore, we introduced the N-dimensional discrete cosine transform as an approximation of N-dimensional singular value decomposition. Finally, to evaluate the effects of the N-dimensional discrete cosine transform and N-dimensional singular value decomposition for $N = 2, 3$ on tensor pattern recognition, we presented two numerical examples.

For the first example, we presented two validations for two-dimensional images. Using the images compressed by the marginal eigenvalue method and two-dimensional discrete cosine transform, we computed the cumulative contribution ratio of the eigenvalues of a tensor subspace as the first validation. As the second validation, we computed the accuracy of image pattern recognition for images compressed by the marginal eigenvalue method and two-dimensional discrete cosine transform. All the results in these two validations demonstrated the equivalent performance of the marginal eigenvalue method and two-dimensional discrete cosine transform for two-dimensional image pattern recognition.

For the second example, we presented two validations for sequences of the two-dimensional images and voxel images of human livers. Using the sequences of binary images compressed by the iterative algorithm of the higher-order singular value decomposition and the three-dimensional discrete cosine transform, we computed the cumulative contribution ratio of the eigenvalues of a tensor subspace as the first validation. As the second validation, using the sequence images and voxel images, we computed the accuracy of tensor pattern recognition for tensors compressed by the iterative algorithm and the three-dimensional discrete cosine transform for the sequences and volumetric data. All the results in these two validations demonstrated the equivalent performance of the higher-order singular value decomposition and three-dimensional discrete cosine transform for third-order tensor pattern recognition. Furthermore, for the decomposition procedure, the results showed that tensor projection is independent of the order of selection of the modes in tensor projections. Moreover, for the sequences compressed the higher-order singular value decomposition, these results showed that the cumulative contribution ratio and recognition ratio are independent of the number of iterations in the decomposition procedure.

These two numerical examples illustrated that the N-dimensional discrete cosine transform can be an acceptable approximation for N-dimensional singular value decomposition for $N = 2, 3$ in tensor pattern recognition if we adopt the Euclidean distance as the metric of the pattern space. These examples also imply that the approximation of the higher-order singular value decomposition by the discrete cosine transform may be valid for tensors with order higher than the third order without an iterative computation method. These approximations are useful for the practical and fast computation of tensor recognition.

Acknowledgements This research was supported by the "Multidisciplinary Computational Anatomy and Its Application to Highly Intelligent Diagnosis and Therapy" project funded by a Grant-in-Aid for Scientific

Research on Innovative Areas from MEXT, Japan, and by Grants-in-Aid for Scientific Research funded by the Japan Society for the Promotion of Science.

References

1. Itoh, H., Imiya, A., Sakai, T.: Low-dimensional tensor principle component analysis. Proc. CAIP Part (I) **9256**, 715–726 (2015)
2. Makihara, Y., Mannami, H., Tsuji, A., Hossain, M.A., Sugiura, K., Mori, A., Yagi, Y.: The OU-ISIR gait database comprising the treadmill dataset. IPSJ Trans. Comput. Vis. Appl. **4**, 53–62 (2012)
3. von Siebenthal, M., Cattin, P.H., Gamper, U., Lomax, A., Székely, G.: 4D MR imaging using internal respiratory gating. In: Proceedings of the MICCAI. pp. 336–343 (2005)
4. Leibe, B., Schiele, B.: Analyzing appearance and contour based methods for object categorization. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 409–415 (2003)
5. Everingham, M., Eslami, S.M.A., Gool, L.V., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL visual object classes challenge: a retrospective. Int. J. Comput. Vis. **111**, 98–136 (2015)
6. Boyer, K.L., Ünsalan, C.: Multispectral Satellite Image Understanding: From Land Classification to Building and Road Detection. Springer, New York (2012)
7. Ely, G., Aeron, S., Hao, N., Kilmer, M.E.: 5D seismic data completion and denoising using a novel class of tensor decompositions. Geophysics **80**(4), V83–V95 (2015)
8. Cohen, N., Shashua, A.: Simnets: a generalization of convolutional networks. In: Proceedings of the NIPS Workshop on Deep Learning (2014)
9. Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Ranzato, M., Senior, A., Tucker, P., Yang, K., Le, Q.V., Ng, A.Y.: Large scale distributed deep networks. In: Proceedings of the NIPS, pp 1232–1240 (2012)
10. Lu, H., Plataniotis, K.N., Venetsanopoulos, A.N.: MPCA: multilinear principal component analysis of tensor objects. IEEE Trans. Neural Netw. **19**(1), 18–39 (2008)
11. Lu, H., Plataniotis, K.N., Venetsanopoulos, A.N.: A survey of multilinear subspace learning for tensor data. Pattern Recognit. **44**, 1540–1551 (2011)
12. Yang, J., Zhang, D., Frangi, A.F., Yang, J.-Y.: Two-dimensional PCA: a new approach to appearance-based face representation and recognition. IEEE Trans. PAMI **26**, 131–137 (2004)
13. Otsu. N.: Mathematical studies on feature extraction in pattern recognition. PhD thesis, Electrotechnical Laboratory (1981)
14. Aase, S.O., Husoy, J.H., Waldemar, P.: A critique of SVD-based image coding systems. In: Proceedings of the IEEE International Symposium on Circuits and Systems vol. 4, pp. 13–16 (1999)
15. Ding, C., Ye, J.: Two-dimensional singular value decomposition (2DSVD) for 2D maps and images. In: Proceedings of the SIAM International Conference on Data Mining, pp 32–43 (2005)
16. Golub, G.H., Van Loan, C.F.: Matrix Computations. The Johns Hopkins University Press, Baltimore (1996)
17. Ye, J., Janardan, R., Qi, L.: GPCA: an efficient dimension reduction scheme for image compression and retrieval. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 354–363 (2004)
18. Helmke, U., Moore, J.B.: Singular-value decomposition via gradient and self-equivalent flows. Linear Algebra Appl. **169**, 223–248 (1992)
19. Moore, J.B., Mahony, R.E., Helmke, U.: Numerical gradient algorithms for eigenvalue and singular value calculations. SIAM J. Matrix Anal. Appl. **15**, 881–902 (1994)
20. Tucker, L.: Some mathematical notes on three-mode factor analysis. Psychometrika **31**(3), 279–311 (1966)
21. Kroonenberg, P.M., Leeuw, J.: Principal component analysis of three-mode data by means of alternating least squares algorithms. Psychometrika **45**(1), 69–97 (1980)
22. Cichoki, A., Zdunek, R., Phan, A.H., Amari, S.: Nonnegative Matrix and Tensor Factorizations. Wiley, Hoboken (2009)
23. Lathauwer, L.D., De Moor, B., Vandewalle, J.: A multilinear singular value decomposition. SIAM J. Matrix Anal. Appl. **21**(4), 1253–1278 (2000)
24. Lathauwer, L.D., De Moor, B., Vandewalle, J.: On the best rank-1 and rank-(r_1, r_2, r_n) approximation of higher-order tensors. SIAM J. Matrix Anal. Appl. **21**(4), 1324–1342 (2000)
25. Inoue, K., Hara, K., Urahama, K.: Robust multilinear principal component analysis. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 591–597 (2009)
26. Lu, H., Plataniotis, K.N., Venetsanopoulos, A.N.: Uncorrelated multilinear principal component analysis for unsupervised multilinear subspace learning. IEEE Trans. Neural Netw. **20**(11), 1820–1836 (2009)
27. Allen, G.I.: Sparse higher-order principal components analysis. In: Proceedings of the International Conference on Artificial Intelligence and Statistics, pp. 27–36 (2012)
28. Itoh, H., Sakai, T., Kawamoto, K., Imiya, A.: Topology-preserving dimension-reduction methods for image pattern recognition. In: Proceedings of the Scandinavian Conference on Image Analysis, pp. 195–204 (2013)
29. Oja, E.: Subspace Methods of Pattern Recognition. Research Studies Press, Baldock (1983)
30. Hamidi, M., Pearl, J.: Comparison of the cosine and fourier transforms of Markov-1 signals. IEEE Trans. Acoust. Speech Signal Process. **24**(5), 428–429 (1976)
31. Wang, Y., Gong, S.: Tensor discriminant analysis for view-based object recognition. Proc. CVPR **3**, 33–36 (2006)
32. Hua, G., Viola, P.A., Drucker, S.M.: Face recognition using discriminatively trained orthogonal rank one tensor projections. Proc. CVPR (2007)
33. Ye, J.: Generalized low rank approximations of matrices. In: Proceedings of the ICML (2004)
34. Liang, Z., Shi, P.: An analytical algorithm for generalized low-rank approximations of matrices. Pattern Recognit. **38**, 2213–2216 (2005)
35. Georghiades, A.S., Belhumeur, P.N., Kriegman, D.J.: From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE Trans. PAMI **23**, 643–660 (2001)
36. Samaria, F., Harter, A.: Parameterisation of a stochastic model for human face identification. In: Proceedings of the IEEE Workshop on Applications of Computer Vision (1994)
37. Mobahi, H., Collobert, R., Weston, J.: Deep learning from temporal coherence in video. In: Proceedings of the 26th International Conference on Machine Learning, Montreal, Canada (2009)
38. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**, 2278–2324 (1998)
39. Saito, T., Yamada, H., Yamada, K.: On the data base ETL9 of handprinted characters in JIS Chinese characters and its analysis. IEEJ J. Ind. Appl. **68**(4), 757–764 (1985)

Hayato Itoh received his M. Eng. degree from Chiba University, Japan, in 2012. He is a research scholar pursuing his Ph. D in theoretical aspects of pattern recognition and computer vision. His research interests include metric-space theory and its application to image pattern recognition and image registration.

Atsushi Imiya is Professor of Super Computing Division at IMIT, Chiba University. He has served as a PC member of DGCI, IWCIA, and SSVM conferences for many years. He is an editorial member of "Pattern Recognition (Journal)" and a co-editor of seminar publications for "Digital and Image Geometry" held at Schloss Dagstuhl in 2000, and MLDM2007 (Machine Learning and Data Mining in Pattern Recognition), proceedings of which were published from Springer-Verlag. He was a general co-chair of S+SSPR (Statistical, and Synthetic and Structural Pattern Recognition) 2012 (Proceedings were published LNCS7626 from Springer). He is participating in a government-funded project titled: "Multidisciplinary computational anatomy and its application to highly intelligent diagnosis and therapy" as an applied mathematician from 2013. He was elected as a Fellow of IAPR in 2008 for contributions to randomized model fitting approaches in pattern recognition and computer vision, and to discrete geometry and a Fellow of IEICE, Japan in 2015 for contributions in to mathematical image analysis and its applications. He also serves as a review committee member of the research projects internationally.

Tomoya Sakai is currently an Associate Professor at the Graduate School of Engineering, Nagasaki University, Japan. He received his M. Eng. degree and Ph.D Engineering from Chiba University in 1998 and 2001, respectively. He served as an Assistant Professor at IMIT, Chiba University from 2001 to 2010. He is a member of IEEE, IPSJ and IEICE. He currently serves as an Associate Editor of the IEICE Transactions on Fundamentals of Electronics. His research interests range from signal processing to machine learning for computer vision and pattern recognition with a particular focus on sparsity-aware approaches to high-dimensional data analysis.