Generalized Image Reconstruction over T-Algebra

Liang Liao
School of Electronics and Information
Zhongyuan University of Technology
Zhengzhou, China
liaoliangis@126.com

Xuechun Zhang
School of Electronics and Information
Zhongyuan University of Technology
Zhengzhou, China
zhangxuechun2020@126.com

Xinqiang Wang
School of Electronics and Information
Zhongyuan University of Technology
Zhengzhou, China
dqxwxq@sina.com

Sen Lin
School of Electronics and Information
Zhongyuan University of Technology
Zhengzhou, China
lins16@126.com

Xin Liu
Information Center
Yellow River Conservancy Commission
Zhengzhou, China
liuxinis@126.com

Abstract—**Principal Component Analysis (PCA) is well known for its capability of dimension reduction and data compression. However, when using PCA for compressing/reconstructing images, images need to be recast to vectors. The vectorization of images makes some correlation constraints of neighboring pixels and spatial information lost. To deal with the drawbacks of the vectorizations adopted by PCA, we used small neighborhoods of each pixel to form compoun pixels and use a tensorial version of PCA, called TPCA (Tensorial Principal Component Analysis), to compress and reconstruct a compound image of compound pixels. Our experiments on public data show that TPCA compares favorably with PCA in compressing and reconstructing images. We also show in our experiments that the performance of TPCA increases when the order of compound pixels increases.**

Index Terms—**generalized image analysis, tensorial algebra, principal component analysis, compound image**

I. Introduction

Principal component analysis (PCA), well-known for dimension reduction and information extraction, is widely used in machine intelligence, data compression, image analysis, etc.

Due to the wide-spread of PCA, many authors try to improve PCA. A PCA variant called 2DPCA (Two-Dimensional PCA) is proposed by Yang et al. with face recognition applications. The covariance matrix size for computing the eigenvectors in 2DPCA is much smaller than that of PCA [1]. Zhang et al. propose the $(2D)^2$PCA (Two-Directional Two-Dimensional PCA), which extracts information along the row direction and column direction at the same time, and its recognition rate is claimed to be higher than that of 2DPCA [2]. Mat-PCA is proposed by Chen et al. Compared with PCA, Mat-PCA computes the covariance matrix of images and calculates the feature vectors more effectively [3].

Although the above methods either reduce the computational complexity or improve computational performance, they do not particularly consider structural constraints at the pixel level. Whether one can adopt a straightforward and systematical approach to exploiting the spatial constraints of pixels? If the answer is yes, one might need a non-naïve algebraic approach. One genesis of such an approach can be the "t-product" model by M. E. Kilmer et al., which is used to generalize the SVD (Singular Value Decomposition) of a matrix to its third-order version, called t-SVD [4], [5]. Inspired by the "t-product" model, Liao and Maybank propose a novel semisimple communicative algebra, called t-algebra, via the multi-way Fourier transform [6], [7]. The elements of the t-algebra are represented by fixed-sized high-order complex arrays and generalize complex numbers, therefore, referred to as "t-scalars." With the paradigm shift from over the field of complex numbers to over the recently-proposed t-algebra, the classical principal component analysis (PCA) can be straightforwardly generalized to its higher-order version, called TPCA (Tensorial PCA) [6], [7], [8]. To demonstrate the t-matrix paradigm and the performance of TPCA, we use TPCA to compress and reconstruct image data.

As a following research of the work in [6], [7], besides demonstrating the t-matrix paradigm and TPCA's performance, we also investigate the effect of the t-scalar order and size on TPCA. We use the neighborhood strategy reported in [6, 7] to extend each image pixel to a high-order compound pixel represented by a t-scalar. Consequently, a generalized image is referred to as a compound image, which is represented by a t-matrix. Then, we adopt TPCA to analyze an obtained compound image and compare its performance, in data compression and reconstruction, with its canonical counterpart PCA.

The remainder of this paper is organized as follows. In Section II, we introduce the mechanism of TPCA and its reconstruction on compound images. In Section III, we

describe of TPCA's implementation via a decomposition mechanism in the Fourier domain and two stratgies of extend a legacy monochrome image to a compound image represented by a t-matrix. In Section IV, we give the expriment results by PCA and different TPCA variants on public images. We conclude this paper in Section V.

II. Image Reconstruction via TPCA

The notations, index protocols, symbols of this paper follow the existing ones of [7] as much as possible. For example, all indices begin from 1 rather than 0. Different symbol subscripts other than symbol fonts are used for different data types since there are many data types rather than just canonical scalars, vectors, matrices, and tensors. Interested readers are referred to [7] for more details of these symbol subscripts. To be more precise, some example notations with the t-matrix paradigm and their descriptions are given in table I.

TABLE I
Some notations and their descriptions used in this paper

Notations	Description
$C \equiv \mathbb{C}^{I_1 \times \cdots \times I_N}$	t-algebra of t-scalars
X_{vec}, X_{mat}	canonical vector and matrix over complex numbers
X_{TV}, X_{TM}	t-vector and t-matrix, i.e., vector and matrix over C
$Z_T, E_T \in C$	zero t-scalar, identity t-scalar
Z_{TM}, I_{TM}	zero t-matrix, identity t-matrix
$X_T + Y_T \in C$	t-scalar addition
$X_{TM} + Y_{TM}$	t-matrix addition
$X_T \circ Y_T \in C$	t-scalar multiplication
$X_{TM} \circ Y_{TM}$	t-matrix multiplication
$\tilde{X}_{TV} \doteq F(X_{TV})$	multi-way Fourier transform of a t-vector X_{TV}
$X_{TV} \doteq F^{-1}(\tilde{X}_{TV})$	inverse multi-way Fourier transform for a t-vector
$\tilde{X}_{TV}(i_1, \cdots, i_N)$	the $(i_1, \cdots, i_N)$-th slice of a transformed t-vector in the Fourier domain
$(X_{TM})_{1:d_1, 1:d_2}$	a t-matrix in $C^{d_1 \times d_2}$, which perserves the first d_1 and the first d_2 columns of the t-matrix X_{TM}
X_{TM}^*	conjugate tranpose of a t-matrix X_{TM}

For those notations not appearing in [7], we give their descriptions when necessary.

A. PCA

Due to image data redundancy, a raw image represented by a vector can be effectively mapped to a low-dimensional subspace. Canonical PCA seeks such low-dimensional subspace and extracts predominant image features with the help of SVD of a matrix. The specific steps can be summarized as follows.

Given K training vectors $X_{vec,1}, \cdots, X_{vec,K} \in \mathbb{R}^D$, their covariance matrix $G_{mat} \in \mathbb{R}^{D \times D}$ of these vectors is given by

$$G_{mat} = \frac{1}{K-1} \cdot \sum_{k=1}^{K} (X_{vec,k} - \bar{X}_{vec}) \cdot (X_{vec,k} - \bar{X}_{vec})^T \quad (1)$$

where

$$\bar{X}_{vec} = (1/K) \cdot \sum_{k=1}^{K} X_{vec,k} . \quad (2)$$

It is easy to verify that G_{mat} is symmetric, namely $G_{mat} = G_{mat}^T$. Let the SVD of G_{mat} be

$$G_{mat} = U_{mat} \cdot S_{mat} \cdot U_{mat}^T . \quad (3)$$

such that the matrix $U_{mat} \in \mathbb{R}^{D \times D}$ is orthogonal; namely, $U_{mat}^T \cdot U_{mat} = U_{mat} \cdot U_{mat}^T = I_{mat}$, where I_{mat} denotes the identity matrix, and the matrix $S_{mat} \in \mathbb{R}^{D \times D}$ is diagonal.

The feature vector of a test vector $Y_{vec} \in \mathbb{R}^D$ can be computed via the orthogonal matrix U_{mat} and the average vector $\bar{X}_{vec}$ of the training vectors, namely

$$Y_{vec}^{feature} = U_{mat}^T \cdot (Y_{vec} - \bar{X}_{vec}) \in \mathbb{R}^D \quad (4)$$

If one needs to reduce the dimension of the feature vector $Y_{vec}^{feature} \in \mathbb{R}^D$ from D to d, the last $(D-d)$ entries of Y_{vec} are just discarded.

The reconstructed vector $Y_{vec}^{recon} \in \mathbb{R}^D$ is computed from the feature vector $Y_{vec}^{feature}$ and the matrix U_{mat} as follows.

$$Y_{vec}^{recon} = (U_{mat})_{:,1:d} \cdot (Y_{vec}^{feature})_{1:d} + \bar{X}_{vec} \quad (5)$$

where $(U_{mat})_{:,1:d}$ denotes the sub-matrix that has the first d columns of the matrix U_{mat}, and $(Y_{vec}^{feature})_{1:d}$ denotes the sub-vector that remains the first d entries of the vector $Y_{vec}^{feature}$.

B. TPCA

TPCA is a straightforward generalization, over the t-algebra C, of the canonical PCA [6], [7] and summarized as follows.

Given the training t-vectors $X_{TV,1}, \cdots, X_{TV,K} \in C^D \equiv \mathbb{C}^{I_1 \times \cdots \times I_N \times D}$, the covariance t-matrix $G_{TM} \in C^{D \times D} \equiv \mathbb{C}^{I_1 \times \cdots \times I_N \times D \times D}$ is given by

$$\begin{cases} G_{TM} = \frac{1}{K-1} \cdot \sum_{k=1}^{K} (X_{TV,k} - \bar{X}_{TV}) \circ (X_{TV,k} - \bar{X}_{TV})^* \\ \bar{X}_{TV} = (1/K) \cdot \sum_{k=1}^{K} X_{TV,k} \end{cases} \quad (6)$$

where $\circ$ denotes the t-matrix multiplication defined via the multi-way circular convolution at the t-scalar entry level. Interested readers are referred to the following text for the relevant definitions [6], [7].

Let $G_{TM} = U_{TM} \circ S_{TM} \circ U_{TM}^*$ be the TSVD of the t-matrix $G_{TM} \in C^{D \times D} \equiv \mathbb{C}^{I_1 \times \cdots I_N \times D \times D}$ such that

$$U_{TM}^* \circ U_{TM} = I_{TM} \equiv \mathrm{diag}(E_T, \cdots, E_T)$$
$$\in C^{D \times D} \equiv \mathbb{C}^{I_1 \times \cdots \times I_N \times D \times D},$$
$$S_{TM} \doteq \mathrm{diag}(\lambda_{T,1}, \cdots, \lambda_{T,D})$$
$$\in C^{D \times D} \equiv \mathbb{C}^{I_1 \times \cdots I_N \times D \times D} . \quad (7)$$

Then, the orthogonal t-matrix $U_{TM} \in C^{D \times D}$ and the average t-vector $\bar{X}_{TV} \in C^D$ can be used to reconstruct the t-vector $Y_{TV}^{recon} \in C \equiv \mathbb{C}^{I_1 \times \cdots \times I_N \times D}$, an approximation to the input t-vector $Y_{TV} \in C \equiv \mathbb{C}^{I_1 \times \cdots \times I_N \times D}$. More precisely, one thas the following equations.

$$Y_{TV}^{feature} = U_{TM}^* \circ (Y_{TV} - \bar{X}_{TV}) \quad (8)$$

and

$$Y_{TV}^{recon} = (U_{TM})_{:,1:d} \circ (Y_{TV}^{feature})_{1:d} + \bar{X}_{TV} \quad (9)$$

where $(U_{TM})_{:,1:d} \in C^{D \times d} \equiv \mathbb{C}^{I_1 \times \cdots \times I_N \times D \times d}$ denotes the sub-t-matrix which remains the first d columns (i.e., t-vectors) of the t-matrix $U_{TM} \in C^{D \times D}$, and $(Y_{TV}^{feature})_{1:d} \in$

C^d denotes the sub-t-vector which remains the first d t-scalar entries of $Y_{TV}^{feature}$.

Using the t-matrix paradigm reported in [7], one can verify that when $I_1 = \cdots = I_N = 1$, all t-vectors and t-matrices reduce to their canonical versions, i.e., canonical vectors and matrices. Furthermore, equations (8) and (9) respectively reduce to equations (4) and (5). Namely, in this scenario, TPCA reduces to PCA.

III. TPCA Decomposed and Beyond

We describe in this section the mechanism to t-matricize a legacy image to its high-order counterpart, i.e., a compound image, and decompose TPCA to a finite number of canonical constituents.

A. Compound pixels

With the semisimple paradigm over the t-algebra in [6], [7], t-scalars are represented by fixed-sized order-N arrays in $C \equiv \mathbb{C}^{I_1 \times \cdots \times I_N}$.

A $D_1 \times D_2$ matrix over C, i.e., a $D_1 \times D_2$ matrix with t-scalar entries, is represented by an order-$(N+2)$ array in $C^{D_1 \times D_2} \equiv \mathbb{C}^{I_1 \times \cdots \times I_N \times D_1 \times D_2}$. A matrix over C is called a "t-matrix". A vector over C is called a "t-vector", which is a special case of t-matrix.

T-matrices can be scaled, added, scaled, multiplied, conjugate transposed, inverted or pseudo-inverted, in a way backward-compatible to their canonical counterparts defined over complex numbers [7].

To use the t-matrix paradigm for general image analysis, one needs a mechanism to extend a pixel, represented by a scalar, to its high-order version, i.e., a compound pixel represented by a t-scalar in the form of an order-N array.

Such an extension from one scalar (pixel) to a t-scalar (compound pixel) can be realized using a fixed-sized neighborhood of each pixel. An $D_1 \times D_2$ image with compound pixel is called a "compound image" and represented by a t-matrix in $C^{D_1 \times D_2} \equiv \mathbb{C}^{I_1 \times \cdots \times I_N \times D_1 \times D_2}$.

We use two pixel neighborhood strategies to extend each pixel to its high-order version, i.e., a compound pixel.
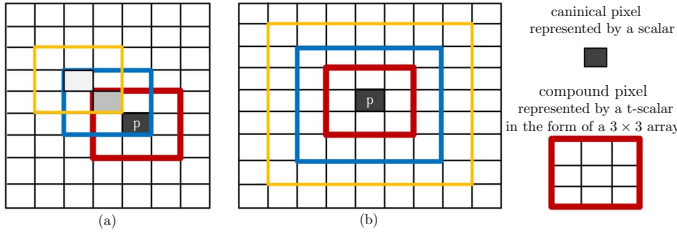

Fig. 1. Two strategies of using pixel neighborhoods to extend each pixel to a compound pixel

Compound pixel strategy 1. The first strategy is shown in figure 1(a). Given a canonical pixel p, one can find a 3×3 neighborhood centered at p. A red-line box in the subfigure highlights the 3×3 neighborhood.

Now, one extends a scalar/pixel to an order-two t-scalar, i.e., a 3×3 array.

One can reuse the 3×3 neighborhood strategy to increase the order of t-scalars further.Each pixel (scalar) in the red-line box has a 3×3 neighborhood. For example, the top-left pixel (scalar) in the red-line box has a 3×3 neighborhood highlighted by a blue-line box.

When one can substitutes each pixel (scalar) in the red box, one can have an order-four array of size $3 \times 3 \times 3 \times 3$. Similarly, after a third reusing the 3×3 neighborhood strategy, one can extend a pixel (scalar) p to its order-six version, i.e., a compound pixel (t-scalar) in the form of a $3 \times 3 \times 3 \times 3 \times 3 \times 3$ array.

Compound pixel strategy 2. The second strategy is shown in figure 1(b). This strategy increases the size of the array rather than its order. The t-scalar extension for the pixel p is always in the form of an order-two array. However, one can use a 3×3 neighborhood, 5×5 neighborhood, 7×7 neighborhood or 9×9 neighborhood, etc., for the pixel p.

If a scalar (pixel) is located at the image border, one can pad with 0 when necessary to have a neighborhood. No matter using which strategy, when the size of t-scalars is determined, it is fixed and should not be changed anymore.

Analogous to the process of recasting an image to a vector, one can recast an obtained compound image to a t-vector.

Using the multi-way Fourier transform described in [6] and [7], one can decompose any t-vector/t-matrix operation to a finite number of canonical vector/matrix counterparts carried out in the Fourier domain.

TPCA can be efficiently implemented in this way using the multi-way Fourier transform on input data. More precisely, TPCA is given by Algorithm 1.

Fourier slice. In Algorithm 1 we use the so-called Fourier slice mechanism to decompose a fourier-transformed t-matrix $\tilde{X}_{TM} \in C^{D_1 \times D_2} \equiv \mathbb{C}^{I_1 \times \cdots \times I_N \times D_1 \times D_2}$ to K^{slice} complex matrices $\tilde{X}_{TM}(i_1, \cdots, i_N) \in \mathbb{C}^{D_1 \times D_2}$ for all $(i_1, \cdots, i_N) \in [I_1] \times \cdots \times [I_N]$, where $K^{slice} \doteq I_1 \cdots I_N$ such that the following condition holds for all $(d_1, d_2) \in [D_1] \times [D_2]$,

$$\left(\tilde{X}_{TM}(i_1, \cdots, i_N) \right)_{d_1, d_2} = \left((\tilde{X}_{TM})_{d_1, d_2} \right)_{i_1, \cdots, i_N} \in \mathbb{C} . \quad (10)$$

where $(\tilde{X}_{TM})_{d_1, d_2} \in C \equiv \mathbb{C}^{I_1 \times \cdots \times I_N}$.

When $D_2 = 1$, the fourier-transformed t-matrix $\tilde{X}_{TM}$ reduces to a fourier-transformed t-vector $\tilde{X}_{TV}$. Therefore, one has the fourier slice $\tilde{X}_{TV}(i_1, \cdots, i_N)$ for all $(i_1, \cdots, i_N) \in [I_1] \times \cdots \times [I_N]$.

IV. Experimental Results and Analysis

We use PCA and TPCA on the public MNIST handwritten digit dataset for image reconstruction experiments. Each image of the MNIST dataset is a monochrome image of the size 28×28. There are ten classes in the dataset, i.e., the classes respectively for the digits $0, \cdots, 9$. We randomly choose 60 images from each class as training images and 10 images from each class as query images.

Algorithm 1 Principal features extraction and data reconstruction via TPCA for t-vectors

---

Input: training t-vectors $X_{TV,1}, \cdots, X_{TV,K}$ and a query t-vector Y_{TV} in $C^D \equiv \mathbb{C}^{I_1 \times \cdots \times I_N \times D}$

Output: feature t-vector $Y_{TV}^{feature} \in C^D \equiv \mathbb{C}^{I_1 \times \cdots \times I_N \times D}$ as in equation (8) and reconstructed t-vector $Y_{TV}^{recon} \in C^D \equiv C^{I_1 \times \cdots \times I_N \times D}$ as in equation (9).

1: Compute the following Fourier transforms

$$\tilde{X}_{TV,k} \leftarrow F(X_{TV\,k}), \; \forall k \in [K], \; \tilde{Y}_{TV} \leftarrow F(Y_{TV}).$$

2: **for all** $(i_1, \cdots, i_N) \in [I_1] \times \cdots \times [I_N]$ **do**

3:　Compute the following covariance matrix $G_{mat} \in \mathbb{C}^{D \times D}$.

$$G_{mat} = \frac{1}{K-1} \cdot \sum_{k=1}^{K} \big(\tilde{X}_{TV}(i_1, \cdots, i_N) - \bar{X}_{vec} \big) \cdot \big(\tilde{X}_{TV}(i_1, \cdots, i_N) - \bar{X}_{vec} \big)^H$$

　　where $\bar{X}_{vec} = \frac{1}{K} \cdot \sum_{k=1}^{K} \tilde{X}_{TV}(i_1, \cdots, i_N) \in \mathbb{C}^D$.

4:　Compute the canonical SVD of the hermitian matrix $G_{mat} \in \mathbb{C}^{D \times D}$ such that

$$G_{mat} = U_{mat} \cdot S_{mat} \cdot U_{mat}^H$$

　　where $U_{mat} \in \mathbb{C}^{D \times D}$, $S_{mat} \in \mathbb{C}^{D \times D}$.

5:　Compute the $(i_1, \cdots, i_N)$-th fourier slice of the fourier-transformed t-vectors $\tilde{Y}_{TV}^{feature}, \tilde{Y}_{TV}^{recon} \in \mathbb{C}^{I_1 \times \cdots I_N \times D}$, more precisely,

$$\tilde{Y}_{TV}^{feature}(i_1, \cdots, i_N) \leftarrow U_{mat}^H \cdot \big(\tilde{Y}_{TV}(i_1, \cdots, i_N) - \bar{X}_{vec} \big)$$

$$\tilde{Y}_{TV}^{recon}(i_1, \cdots, i_N) \leftarrow (U_{mat})_{:,1:d} \cdot \big(\tilde{Y}_{TV}^{feature}(i_1, \cdots, i_N) \big)_{1:d} + \bar{X}_{vec}$$

6: **end for**

7: **return** $Y_{TV}^{feature} \in C^D \equiv \mathbb{C}^{I_1 \times \cdots \times I_N \times D}$ and $Y_{TV}^{recon} \in C^D \equiv \mathbb{C}^{I_1 \times \cdots \times I_N \times D}$ give by $Y_{TV}^{feature} \leftarrow F^{-1}(\tilde{Y}_{TV}^{feature})$ and $Y_{TV}^{recon} \leftarrow F^{-1}(\tilde{Y}_{TV}^{recon})$.

TABLE II
TPCA with different shapes of t-scalars

TPCA variant	t-scalar shape (i.e., $I_1 \times \cdots \times I_N$)	t-scalar order	neighborhood strategy
PCA	1	0	N/A
TPCA	3×3	2	compound pixel strategy 1
TPCA-A	$3 \times 3 \times 3 \times 3$	4	
TPCA-B	$3 \times 3 \times 3 \times 3 \times 3 \times 3$	6	
TPCA-X	5×5	2	compound pixel strategy 2
TPCA-Y	7×7	2	
TPCA-Z	9×9	2	

We compare the reconstruction performances of PCA and TPCA with different t-scalar shapes.

Using or reusing the neighborhood of each pixel described as in strategies 1 and 2, one can have different t-scalar shapes. In this section, TPCA with different t-scalar shapes is respectively named as TPCA, TPCA-A, TPCA-B (TPCA variants using compound pixel strategy 1) and TPCA-X, TPCA-Y, TPCA-Z (TPCA variants using compound pixel strategy 2).

When each t-scalar only contains one scalar entry (in other words, t-scalars are order-zero), TPCA reduces to PCA. Namely, PCA is a special TPCA with order-zero t-scalars. The detailed t-scalar settings are given in table II.

The reconstruxtion quality is measured in PSNR (Peak Signal Noise Ratio). More precisely, given an array X and its approximation version X^{recon}, their PSNR is given as follows.

$$PSNR = 20 \log_{10} \frac{MAX \cdot \sqrt{D}}{\|X - X^{recon}\|_F} \quad (11)$$

where MAX denotes the largest possible value of the entries in X and D denotes the number of entry of X, and $\| \cdot \|_F$ stands for the Frobenius norm of an array. In this paper, the parameter MAX is a constant, i.e., $MAX = 255$.

Each raw MNIST image is a 28×28 array, i.e., with 784 pixels. When using PCA, each raw or reconstructed MNIST image is recast to a vector in $\mathbb{R}^{784}$. Hence, one can organize 100 raw or reconstructed vectors, as columns, in an array in $\mathbb{R}^{784 \times 100}$ As a consequence, 100 raw images form an array $X \in \mathbb{R}^{784 \times 100}$ and 100 recontructed vectors form an array X^{recon} of the same size.

By substituting the arrays X and X^{recon} to equation (11) with $D = 78400 \equiv 784 \times 100$, one can have the PSNR of X annd X^{recon}.

Note that TPCA with different shapes of t-scalars yields t-vectors rather than vectors. Each t-vector represents a compound image and is in the form of a high-order array in $C^D \equiv \mathbb{C}^{I_1 \times \cdots \times I_N \times D}$. Our experiment setting is given by table II with the shape of t-scalars being $I_1 \times \cdots \times I_N$.

To give a fair comparison of the performances of PCA and TPCA, we extract the cental spatial slices of the input and output t-vectors yielded by TPCA for computing PSNRs.

A spatial slice is analogous to a Fourier slice described in Section III but extracted in the spatial domain rather than the Fourier domain.

The $(i_1, \cdots, i_N)$-th spatial slice of a t-vector $X_{TV} \in C^D \equiv \mathbb{C}^{I_1 \times \cdots \times I_N \times D}$ is denoted by $X_{TV}(i_1, \cdots, i_N) \in \mathbb{C}^D$ such that, for all $(i, \cdots, i_N, d) \in [I_1] \times \cdots \times [I_N] \times [D]$, the following equality holds

$$\big(X_{TV}(i_1, \cdots, i_N) \big)_d = \big((X_{TV})_d \big)_{i_1, \cdots, i_N} \in \mathbb{C}. \quad (12)$$

where $(X_{TV})_d \in C \equiv \mathbb{C}^{I_1 \times \cdots \times I_N}$.

In our experiments, the shape of t-scalars is given by $I_1 \times \cdots \times I_N$ where $I_1, \cdots, I_N$ are odd positive integers. Therefore, the so-called central spatial slice of a t-vector $X_{TV} \in C^D \equiv \mathbb{C}^{I_1 \times \cdots \times I_N \times D}$ is given by $X_{TV}(\frac{I_1+1}{2}, \cdots, \frac{I_N+1}{2}) \in \mathbb{C}^D$ with $D = 784$ in our experiments.

The input t-vector slices and output reconstructed t-vector slices are respectively organized in two 784×100 arrays X and X^{recon}. The two arrays are substituted in equation (11) with $D = 78400$ for computing PSNRs.

Some quantitative PSNR comparsion of PCA and different TPCA variants is given by table III. A visual comparison of the PSNRs by PCA and different TPCA variants is given by figure 2.

The PSNRs in table III and figure 2 are outputs of

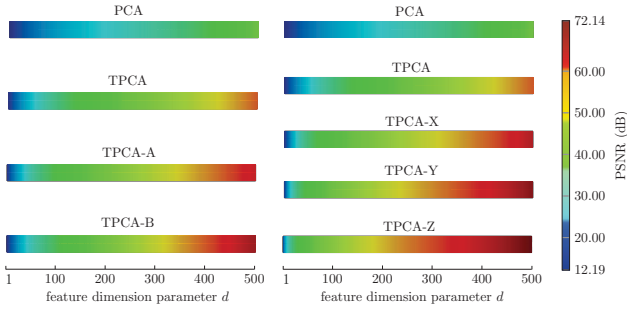PCA and TPCA with the feature dimension parameter $d = 50, 100, \cdots, 500$.

Fig. 2. A heat-map comparison of PSNRs by PCA and different TPCA variants

It shows from table III and figure 2 that when the feature dimension parameter d increases, the reconstruction qualities in terms of PSNRs of PCA and different TPCA variants all increase.

Another observation from table III and figure 2 is that TPCA always outperforms its canonical counterpart PCA. When $d = 500$, TPCA outperforms PCA for at least 17.296 dB (i.e., 55.263 dB- 37.967 dB).

A third observation from the table and figure is that the size of t-scalars effect TPCA's performance, sometimes significantly. For example, a complicated t-scalar shape, such as a $3 \times 3 \times 3 \times 3$ t-scalar shape adopted by TPCA-A, affects the performance favorably than the simple 3×3 t-scalar shape adopted by TPCA.

The results in table III and figure 2 also show that, in the experiment of compound image approximations, TPCA using larger pixel neighborhoods might be more effective than TPCA using higher-order pixel neighborhoods.

For example, TPCA-A and TPCA-Z have the same amount of fourier slices since $3 \times 3 \times 3 \times 3 = 9 \times 9$. However, with the same parameter d, TPCA-Z yields more favorable results than TPCA-A.

One might be interested in the reconstruction performance of PCA and TPCA on each image. Figures 3 and 4 show the PSNR curves of PCA and TPCA over each image.

The variables of the horizontal axes of figures 3 and 4 are image indices, which are sorted to make a PSNR curve of PCA an increasing function of the image index.

Figure 3 corresponds to the setting of the feature dimension $d = 250$. Figure 4 corresponds to the setting of the feature dimension $d = 500$. Both figures corroborate the observations found in table III and figure 2. Namely, TPCA outperforms PCA in terms of reconstruction quality.

V. Conclusion

The performance of generalized image reconstruction, using TPCA (Tensorial Principal Component Analysis),

over a recently-reported commutative algebra, called t-algebra (tensorial algebra) [6], [7] is evaluated.

The t-algebra generalizes the field of complex numbers and contains the generalized complex numbers referred to as t-scalars. Over the t-algebra, one can have a novel matrix paradigm that is backward-compatible to the canonical matrix paradigm. This paradigm shift might bring new insights into general data analytics, particularly into visual information analysis, including but not limited to image reconstruction.

To validate this claim, we conduct some numerical experiments on public images using TPCA for image reconstruction. The classical PCA is a special case of TPCA. The experiments show that TPCA with different shapes of t-scalars consistently outperforms its canonical counterpart PCA. This paper also reports the effect of the shape of t-scalars on the performance of TPCA.

For a more rigorous introduction of the t-algebra and the matrix paradigm over the algebra, interested readers are referred to [6], [7] for more details.

Open Source

A MATLAB repository on the t-algebra, t-vectors, and t-matrices is open-sourced at the following URL.

www. github/liaoliang2020/talgebra.

Interested readers are referred to it for more details.

References

[1] J. Yang, D. Zhang, A. F. Frangi, and J. Yang, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 1, pp. 131–137, 2004.

[2] D. Zhang and Z.-H. Zhou, "(2D)²PCA: two-directional two-dimensional PCA for efficient face representation and recognition," *Neurocomputing*, vol. 69, no. 1-3, pp. 224–231, 2005.

TABLE III
PSNR comparison of PCA and different TPCA variants

| feature dimension parameter d	PCA	TPCA (3×3 t-scalars)	TPCA-A ($3 \times 3 \times 3 \times 3$ t-scalars)	TPCA-B ($3 \times 3 \times 3 \times 3 \times 3 \times 3$ t-scalars)	TPCA-X (5×5 t-scalars)	TPCA-Y (7×7 t-scalars)	TPCA-Z (9×9 t-scalars)
50	18.610	23.449	26.304	25.241	29.070	32.561	35.061
100	21.269	28.646	31.846	30.685	34.408	38.193	41.196
150	22.970	32.456	35.694	34.804	38.263	42.255	45.628
200	24.353	35.366	38.933	38.817	41.519	45.878	49.581
250	25.756	38.102	42.192	42.755	44.651	49.229	52.963
300	27.261	40.832	45.441	46.761	47.593	52.286	55.991
350	29.077	43.578	48.756	51.027	50.588	55.579	59.165
400	31.223	46.712	52.165	55.236	54.071	58.898	62.574
450	34.125	50.463	56.270	60.081	58.045	63.078	66.727
500	37.967	55.263	61.268)	66.067	63.726	68.643	72.141
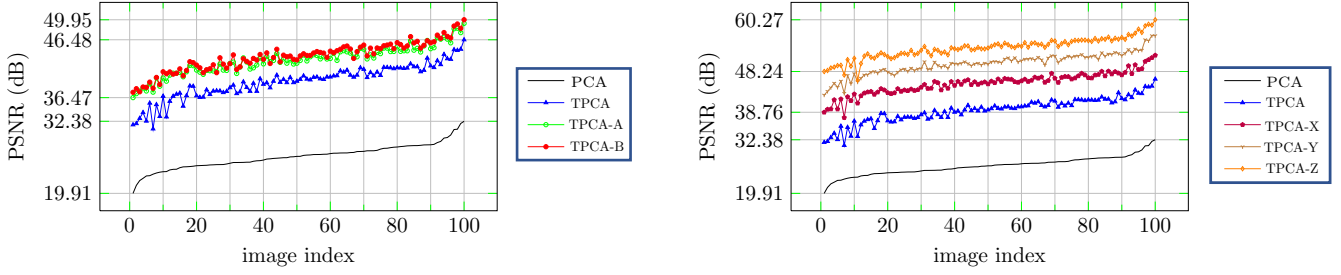
Fig. 3. PSNR comparison on each image reconstructed by PCA and different TPCA variants, feature dimension $d = 250$
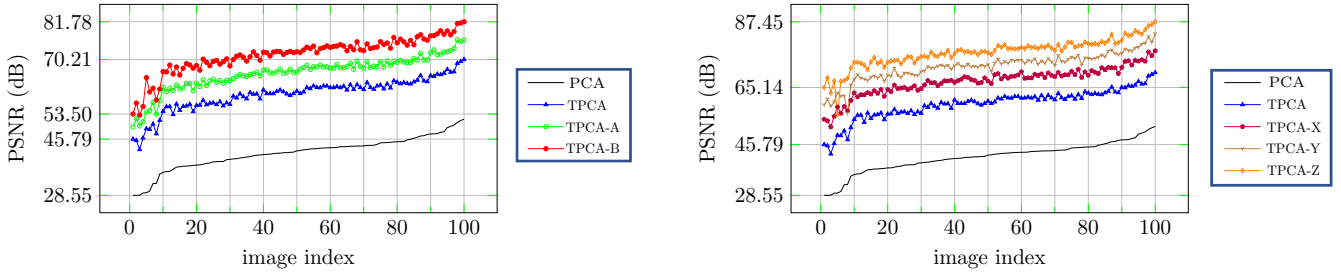

Fig. 4. PSNR comparison on each image reconstructed by PCA and different TPCA variants, feature dimension $d = 500$

[3] S. Chen, Y. Zhu, D. Zhang, and J. Yang, "Feature extraction approaches based on matrix pattern: MatPCA and MatFLDA," *Pattern recognition letters*, vol. 26, no. 8, pp. 1157–1167, 2005.

[4] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra and its Applications*, vol. 435, no. 3, pp. 641–658, 2011.

[5] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 1, pp. 148–172, 2013.

[6] L. Liao and S. J. Maybank, "Generalized visual information analysis via tensorial algebra," *Journal of Mathematical Imaging and Vision*, vol. 62, pp. 560–584, 2020.

[7] ——, "General data analytics with applications to visual information analysis: A provable backward-compatible semisimple paradigm over t-algebra," *arXiv preprint arXiv:2011.00307*, pp. 1–53, 2020.

[8] Y. Ren, L. Liao, S. J. Maybank, Y. Zhang, and X. Liu, "Hyperspectral image spectral-spatial feature extraction via tensor principal component analysis," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 9, pp. 1431–1435, 2017.