International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:3, 2009

Comparison between Higher-Order SVD and Third-order Orthogonal Tensor Product Expansion

Chiharu Okuma, Jun Murakami, and Naoki Yamamoto

***Abstract*—**In digital signal processing it is important to approximate multi-dimensional data by the method called rank reduction, in which we reduce the rank of multi-dimensional data from higher to lower. For 2-dimennsional data, singular value decomposition (SVD) is one of the most known rank reduction techniques. Additional, outer product expansion expanded from SVD was proposed and implemented for multi-dimensional data, which has been widely applied to image processing and pattern recognition. However, the multi-dimensional outer product expansion has behavior of great computation complex and has not orthogonally between the expansion terms. Therefore we have proposed an alterative method, *Third-order Orthogonal Tensor Product Expansion* short for 3-OTPE. 3-OTPE uses the power method instead of nonlinear optimization method for decreasing at computing time. At the same time the group of B. D. Lathauwer proposed *Higher-Order SVD* (HOSVD) that is also developed with SVD extensions for multi-dimensional data.

3-OTPE and HOSVD are similarly on the rank reduction of multi-dimensional data. Using these two methods we can obtain computation results respectively, some ones are the same while some ones are slight different. In this paper, we compare 3-OTPE to HOSVD in accuracy of calculation and computing time of resolution, and clarify the difference between these two methods.

***Keywords*—**Singular value decomposition (SVD), higher-order SVD (HOSVD), higher-order tensor, outer product expansion, power method.

I. Introduction

THE rank reduction and approximation with low rank for given multi-dimensional data are important for digital signal processing computation. For example, in the design of a multi-dimensional digital filter, the specification of multi-dimensional design is generally reduced to a set of 1-dimensional (1-D) specification array. Then the desired multi-dimensional filter can be obtained by combining the sets of 1-D digital filters [1], [2]. The outer product expansion was proposed to decompose for multi-dimensional data with product of vectors (1-D data) [3]. The method has behavior of great computation complex, because which exploit the nonlinear optimization ordinary. Therefore we proposed an alterative method which uses the power method instead of nonlinear optimization for decreasing computing time. We also pointed out that outer product expansion has not orthogonally between the expansion term and showed definitions and calculation method of *Third-order Orthogonal Tensor Product Expansion* (3-OTPE)[4]. And we use the term *Tensor Products Expansion* (TPE) instead of outer product expansion in [3]. Additionally, we developed a calculation method of *Third-order Nonnegative Tensor Product Expansion* (3-NTPE) to design 3-D digital filter [5].

The multi-dimensional data are necessary for applications such as pattern recognition, image processing, Web retrieval, and so on [6], [7], where the application with *Higher-Order Singular Value Decomposition* (HOSVD) proposed by group of B. D. Lathauwer [8], [9] become widely more and more. HOSVD is thought as an extension of singular value decomposition (SVD) [10] for multi-dimensional data. The SVD is well known and widely used as decomposition method for matrices (2-D data) in the digital signal processing. To calculate HOSVD of multi-dimensional data, the SVD method is employed many times. The HOSVD can get the best approximation to a given tensor (multi-dimensional data) on specified dimension, and results the decomposed tensor with the product of vectors. Therefore, the resulted vectors have orthogonally each other of expansion terms.

3-TPE, 3-OTPE, and HOSVD are similarly on definitions, usages, and characters of resolution. However, the numerical calculation method is different respectively. In this paper, we compare 3-OTPE to HOSVD in accuracy of calculation and computing time of expansion, and clarify the difference between these two methods. With our conclusion we can figure out the way to the improvement of both of 3-OTPE and HOSVD.

We first treat *3*rd-order tensor (3-D data) and describe the definition of decomposition method. Next, we explain the expansion algorithm of 3-TPE and HOSVD, show the differences between the two methods, and finally analyze the properties both of them by some examples.

Manuscript received December 27, 2008.

Chiharu Okuma is with the Department of Information and Computer Sciences, Kumamoto National College of Technology, 2659-2 Suya, Koshi, Kumamoto, 861-1102, Japan (corresponding author to provide phone: +81-96-242-6387; fax: +81-96-242-6106; e-mail: chiharu@knct.ac.jp).

Jun Murakami is with the Department of Information and Computer Sciences, Kumamoto National College of Technology, Japan (e-mail: jun@knct.ac.jp).

Naoki Yamamoto is with the Department of Information and Communication Engineering, Kumamoto National College of Technology, Japan (e-mail: naoki@knct.ac.jp).

II. Third-Order Tensor Product Expansion

A. Definition of Higher-Order Tensor

In this paper, higher-order tensors are denoted by calligraphic letters such as $\mathcal{A}$ and $\mathcal{B}$, and the $(i_1, i_2, \cdots, i_N)$ -th elements of a nth-order tensor $\mathcal{A} \in \mathbf{R}^{I_1 \times I_2 \times \cdots \times I_N}$ are denoted by $a_{i_1 i_2 \cdots i_n}$ $(1 \le i_1 \le I_1, \cdots, 1 \le i_n \le I_N)$. Figure 1 shows an image of a 3rd-order tensor.

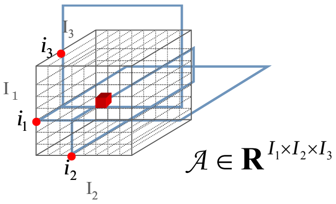

Fig. 1 Image of a 3rd-order tensor.

B. Definition of Tensor Product Expansion

By applying the *tensor product expansion* (TPE), a $L \times M \times N$ 3rd-order tensor $\mathcal{A}$ can be decomposed as

$$\mathcal{A} = \sum_{i=1}^{r} \sigma_i (\mathbf{u}_i \otimes \mathbf{v}_i \otimes \mathbf{w}_i), \tag{1}$$

$$\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_r,$$

where the expansion vectors $\mathbf{u}_i$, $\mathbf{v}_i$, and $\mathbf{w}_i$ correspond to the singular vectors of the SVD of a matrix, the expansion coefficients σ_i and the number of expansion terms r correspond to the singular values and the rank of a matrix similarly, and $\otimes$ denotes the outer product operation [3]. The expansion vectors are normalized as

$$\|\mathbf{u}_i\| = \sqrt{\sum_{j=1}^{L} \mathbf{u}_i(j)^2} = 1,$$
$$\|\mathbf{v}_i\| = \sqrt{\sum_{j=1}^{M} \mathbf{v}_i(j)^2} = 1, \tag{2}$$
$$\|\mathbf{w}_i\| = \sqrt{\sum_{j=1}^{N} \mathbf{w}_i(j)^2} = 1,$$

where $\mathbf{u}_i(j)$, $\mathbf{v}_i(j)$, and $\mathbf{w}_i(j)$ show the j-th element of the vector $\mathbf{u}_i$, $\mathbf{v}_i$, and $\mathbf{w}_i$ respectively.

C. Third-Order Tensor Product Expansion by the Power Method

The algorithm for calculating the *Third-order Tensor Product Expansion* (3-TPE) by the power method is described as follows [4].

Step 1. Choose the initial vectors $\mathbf{u}_n^{(p)}$, $\mathbf{v}_n^{(p)}$, and $\mathbf{w}_n^{(p)}$ arbitrarily, where these vectors must be normalized, and the subscript p and n are set to 0 and 1 respectively at the beginning of this repetitious procedure.

Step 2. The residual 3rd-order tensor $\mathcal{B}$ is obtained by subtracting sum of products of the expansion vectors $\mathbf{u}_i$, $\mathbf{v}_i$, and $\mathbf{w}_i$, which has been calculated by this time, from original tensor $\mathcal{A}$ as follows

$$\mathcal{B} = \mathcal{A} - \sum_{i=1}^{n-1} \sigma_i (\mathbf{u}_i \otimes \mathbf{v}_i \otimes \mathbf{w}_i). \tag{3}$$

Step 3. Calculate the $L \times M$ matrix F by multiplying $\mathcal{B}$ by vector $\mathbf{w}_n^{(p)}$ as

$$\mathbf{F} = \mathcal{B} \cdot \mathbf{w}_n^{(p)}. \tag{4}$$

The (i, j)-th element of the matrix F can be represented as

$$\mathbf{F}(i, j) = \sum_k \mathcal{B}(i, j, k) \mathbf{w}_n^{(p)}(k). \tag{5}$$

Next, apply the power method to the matrix F as follows:

$$\mathbf{u}_n^{(p+1)} = \mathbf{F}\mathbf{v}_n^{(p)}, \mathbf{v}_n^{(p+1)} = \mathbf{F}^T \mathbf{u}_n^{(p+1)}. \tag{6}$$

Likewise the $M \times N$ matrix G and the $N \times L$ matrix H are obtained by

$$\mathbf{G} = \mathcal{B} \cdot \mathbf{v}_n^{(p+1)},$$
$$\mathbf{w}_n^{(p+1)} = \mathbf{G}\mathbf{u}_n^{(p+1)}, \mathbf{u}_n^{(p+1)} = \mathbf{G}^{\mathbf{T}} \mathbf{w}_n^{(p+1)}, \tag{7}$$

$$\mathbf{H} = \mathcal{B} \cdot \mathbf{u}_n^{(p+1)},$$
$$\mathbf{v}_n^{(p+1)} = \mathbf{H}\mathbf{u}_n^{(p+1)}, \mathbf{w}_n^{(p+1)} = \mathbf{H}^{\mathbf{T}} \mathbf{w}_n^{(p+1)}, \tag{8}$$

where the obtained vectors $\mathbf{u}_n^{(p+1)}$, $\mathbf{v}_n^{(p+1)}$, and $\mathbf{w}_n^{(p+1)}$ must be normalized.

Repeat Step 3 until the following conditions are satisfied for sufficiently small value ε

$$\begin{cases} \left\|\mathbf{u}_n^{(p+1)} - \mathbf{u}_n^{(p)}\right\| < \varepsilon, \\ \left\|\mathbf{v}_n^{(p+1)} - \mathbf{v}_n^{(p)}\right\| < \varepsilon, \\ \left\|\mathbf{w}_n^{(p+1)} - \mathbf{w}_n^{(p)}\right\| < \varepsilon. \end{cases} \tag{9}$$

Step 4. The nth expansion vectors $\mathbf{u}_n^{(p+1)}$, $\mathbf{v}_n^{(p+1)}$, and $\mathbf{w}_n^{(p+1)}$ are obtained from Step 3. Here, these vectors are renamed as $\mathbf{u}_n$, $\mathbf{v}_n$, and $\mathbf{w}_n$.

The nth coefficient σ_n is obtained by performing an inner product operation as

$$\sigma_n = \mathcal{B}(\mathbf{u}_n \otimes \mathbf{v}_n \otimes \mathbf{w}_n). \tag{10}$$

Step 5. After increasing n and set p to 0, repeat this procedure from Step 1.

D. Third-Order Orthogonal Tensor Product Expansion

Since the resultant expansion terms of TPE do not satisfy orthogonality, the *Third-order Orthogonal Tensor Product Expansion* (3-OTPE) is defined by

$$\mathcal{A} = \sum_{i,j,k} \sigma_{ijk} (\mathbf{u}_i \otimes \mathbf{v}_j \otimes \mathbf{w}_k), \tag{11}$$

$$\begin{aligned} &(\mathbf{u}_j \otimes \mathbf{v}_j \otimes \mathbf{w}_j)(\mathbf{u}_k \otimes \mathbf{v}_k \otimes \mathbf{w}_k) \\ &= \{\mathbf{u}_j^T \mathbf{u}_k\}\{\mathbf{v}_j^T \mathbf{v}_k\}\{\mathbf{w}_j^T \mathbf{w}_k\} = \mathbf{0}, j \ne k, \end{aligned} \tag{12}$$

where σ_{ijk} are the expansion coefficients [4]. This expansion can be calculated by introducing the Gram-Schmidt orthogonalization process into the Step 3 of the algorithm described in Section *C* as follows

Step 3.1. Along with the Gram-Schmidt process, calculate the vectors $\mathbf{u}'^{(p+1)}_n$, $\mathbf{v}'^{(p+1)}_n$, and $\mathbf{w}'^{(p+1)}_n$ by subtracting the previously obtained terms from vectors $\mathbf{u}_n^{(p+1)}$, $\mathbf{v}_n^{(p+1)}$, and $\mathbf{w}_n^{(p+1)}$ respectively as

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:3, 2009

$$\mathbf{u'}_n^{(p+1)} = \mathbf{u}_n^{(p+1)} - (\mathbf{u}_1^T \mathbf{u}_n^{(p+1)})\mathbf{u}_1 - (\mathbf{u}_2^T \mathbf{u}_n^{(p+1)})\mathbf{u}_2 - \cdots - (\mathbf{u}_{n-1}^T \mathbf{u}_n^{(p+1)})\mathbf{u}_{n-1}, \tag{13}$$

$$\mathbf{v'}_n^{(p+1)} = \mathbf{v}_n^{(p+1)} - (\mathbf{v}_1^T \mathbf{v}_n^{(p+1)})\mathbf{v}_1 - (\mathbf{v}_2^T \mathbf{v}_n^{(p+1)})\mathbf{v}_2 - \cdots - (\mathbf{v}_{n-1}^T \mathbf{v}_n^{(p+1)})\mathbf{v}_{n-1}, \tag{14}$$

$$\mathbf{w'}_n^{(p+1)} = \mathbf{w}_n^{(p+1)} - (\mathbf{w}_1^T \mathbf{w}_n^{(p+1)})\mathbf{w}_1 - (\mathbf{w}_2^T \mathbf{w}_n^{(p+1)})\mathbf{w}_2 - \cdots - (\mathbf{w}_{n-1}^T \mathbf{w}_n^{(p+1)})\mathbf{w}_{n-1}. \tag{15}$$

Normalize the vectors in these equations to obtain $\mathbf{u}_n^{(p+1)}$, $\mathbf{v}_n^{(p+1)}$, and $\mathbf{w}_n^{(p+1)}$.

Step 3.2. By performing the Step 3.1, vectors in the equation (16) are obtained in ascending order of magnitude, where m = min(L, M, N). In case that $L>m$, the remaining $(L-m)$ vectors can be calculated by using Gram-Schmidt orthogonalization process as

$$\mathbf{u}_n' = \mathbf{u}_n - (\mathbf{u}_1^T \mathbf{u}_n)\mathbf{u}_1 - (\mathbf{u}_2^T \mathbf{u}_n)\mathbf{u}_2 \cdots - (\mathbf{u}_{n-1}^T \mathbf{u}_n)\mathbf{u}_{n-1}, \quad n = m+1, \cdots, L, \tag{16}$$

where $\mathbf{u}_n$ are arbitrary chosen vectors initially and the vectors $\mathbf{u}_n'$ are to renamed as $\mathbf{u}_n$ after they are normalized. Likewise vectors $\mathbf{v}_{m+1}, \cdots, \mathbf{v}_M$ and $\mathbf{w}_{m+1}, \cdots, \mathbf{w}_N$ are calculated.

Step 3.3. For every combination of p, q and r, calculate the expansion coefficients σ_{pqr} as

$$\sigma_{pqr} = \mathcal{A}(\mathbf{u}_p \otimes \mathbf{v}_q \otimes \mathbf{w}_r), \quad (p = 1,2,\cdots L,\ q = 1,2,\cdots M,\ r = 1,2,\cdots N). \tag{17}$$

To improve in calculation time of these steps, a part of the Step 3.1 is modified. The modification is described below [5]. After the calculation of the expansion vectors $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_{m-1}$, the vector $\mathbf{u}_m$ can be calculated by

$$\mathbf{u}_m' = \mathbf{u}_m - (\mathbf{u}_1^T \mathbf{u}_m)\mathbf{u}_1 - (\mathbf{u}_2^T \mathbf{u}_m)\mathbf{u}_2 \cdots - (\mathbf{u}_{m-1}^T \mathbf{u}_m)\mathbf{u}_{m-1}, \tag{18}$$

where $\mathbf{u}_m$ is set to arbitrary value initially. The vector $\mathbf{u}_m'$ is normalized immediately, then the vector renamed as $\mathbf{u}_m$. This slight modification leads to an improvement in calculation time.

III. Higher-order SVD

A. Unfolding Matrices of Nth-Order Tensor

A higher-order tensor is represented by some matrices (2nd-order tensor), which are called unfolding matrices. By using this representation, an nth-order tensor $\mathcal{A} \in \mathbf{R}^{I_1 \times I_2 \times \cdots \times I_n}$ is unfolded n matrices $\mathbf{A}_{(n)} \in \mathbf{R}^{I_n \times (I_{n+1} I_{n+2} \cdots I_N I_1 I_2 \cdots I_{n-1})}$. Hence a 3rd-order tensor $\mathcal{A} \in \mathbf{R}^{I_1 \times I_2 \times I_3}$ has 3 unfolding matrices $\mathbf{A}_{(1)} \in \mathbf{R}^{I_1 \times (I_2 I_3)}$, $\mathbf{A}_{(2)} \in \mathbf{R}^{I_2 \times (I_3 I_1)}$, and $\mathbf{A}_{(3)} \in \mathbf{R}^{I_3 \times (I_1 I_2)}$ as illustrated in Fig. 2.

Each unfolding matrix can be decomposed by SVD as follows

$$\mathbf{A}_{(n)} = \mathbf{U}^{(n)} \cdot \mathbf{\Sigma}^{(n)} \cdot \mathbf{V}^{(n)T}, \tag{19}$$

where the vectors $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}$ are left and right singular vectors of matrix $\mathbf{A}_{(n)}$, and matrix $\mathbf{\Sigma}^{(n)}$ is a diagonal matrix whose diagonal elements are the singular values.

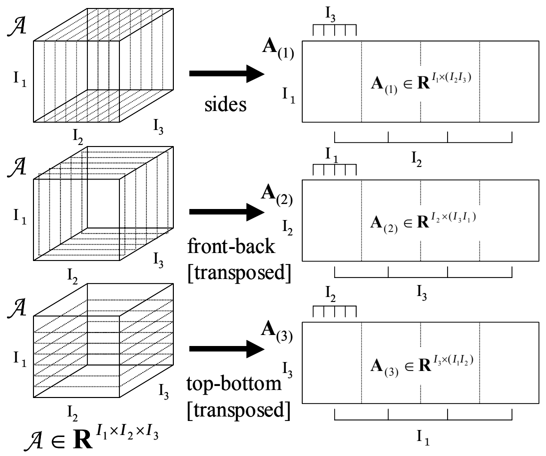

Fig. 2 Unfolding of the 3rd-order tensor $\mathcal{A} \in \mathbf{R}^{I_1 \times I_2 \times I_3}$ to matrix $\mathbf{A}_{(1)} \in \mathbf{R}^{I_1 \times (I_2 I_3)}$, $\mathbf{A}_{(2)} \in \mathbf{R}^{I_2 \times (I_3 I_1)}$, and $\mathbf{A}_{(3)} \in \mathbf{R}^{I_3 \times (I_1 I_2)}$.

B. n-Mode Product

The n-mode product of a tensor $\mathcal{A}$ by a matrix $\mathbf{U} \in \mathbf{R}^{J_n \times I_n}$ is denoted by a symbol $\times_n$ as $\mathcal{A} \times_n \mathbf{U}$. The elements of resultant tensor is defined as

$$(\mathcal{A} \times_n \mathbf{U})_{i_1 i_2 \cdots i_{n-1} j_n i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} a_{i_1 i_2 \cdots i_{n-1} i_n i_{n+1} \cdots i_N} u_{j_n i_n}. \tag{20}$$

By using this n-mode product representation, equation (19) can be written as

$$\mathbf{A}_{(n)} = \mathbf{\Sigma}^{(n)} \times_1 \mathbf{U}^{(n)} \times_2 \mathbf{V}^{(n)}. \tag{21}$$

C. HOSVD Algorithm

An nth-order tensor $\mathcal{A}$ can be denoted by n-mode product as

$$\mathcal{A} = \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \cdots \times_N \mathbf{U}^{(N)} = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} s_{i_1 i_2 \cdots i_N} \mathbf{U}_{i_1}^{(1)} \otimes \mathbf{U}_{i_2}^{(2)} \otimes \cdots \otimes \mathbf{U}_{i_n}^{(N)}, \tag{22}$$

where, the matrices $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \cdots, \mathbf{U}^{(N)}$ are the orthogonal matrices which is obtained by applying SVD to each n-mode unfolding matrix, $\mathbf{U}_{i_1}^{(1)}, \mathbf{U}_{i_2}^{(2)}, \cdots, \mathbf{U}_{i_n}^{(N)}$ are column vectors of that orthogonal matrices $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \cdots, \mathbf{U}^{(N)}$ respectively [8]. $\mathcal{S}$ is an Nth-order tensor called core tensor whose elements are denoted by $s_{i_1 i_2 \cdots i_n} (1 \le i_1 \le I_1, \cdots, 1 \le i_n \le I_N)$, and it is obtained by

$$\mathcal{S} = \mathcal{A} \times_1 \mathbf{U}^{(1)T} \times_2 \mathbf{U}^{(2)T} \cdots \times_N \mathbf{U}^{(N)T}. \quad (23)$$

As described above, we can calculate HOSVD of any higher-order tensors by exploiting the SVD technique for matrices.

D. Best rank-(R_1,R_2, … ,R_N) approximation

The rank of an n-mode unfolding matrix of an Nth-order tensor $\mathcal{A} \in \mathbf{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is called n-rank and defined as

$$\text{rank}_n(\mathcal{A}) = \text{rank}(\mathbf{A}_{(n)}). \quad (24)$$

The best approximation tensor for given tensor $\mathcal{A}$ at the specified n-mode rank rank-(R_1,R_2, ... ,R_N), where $R_1 = \text{rank}_1(\mathcal{A})$, $R_2 = \text{rank}_2(\mathcal{A})$, …, $R_N = \text{rank}_N(\mathcal{A})$, can also be obtained by HOSVD. This approximated tensor $\hat{\mathcal{A}} \in \mathbf{R}^{I_1 \times I_2 \times \cdots \times I_N}$ minimizes the squared norm $\|\mathcal{A} - \hat{\mathcal{A}}\|^2$.

The tensor $\hat{\mathcal{A}}$ is decomposed as

$$\hat{\mathcal{A}} = \mathcal{B} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \cdots \times_N \mathbf{U}^{(N)}, \quad (25)$$

where $\mathbf{U}^{(1)} \in \mathbf{R}^{I_1 \times R_1}$, $\mathbf{U}^{(2)} \in \mathbf{R}^{I_2 \times R_2}$, …, $\mathbf{U}^{(N)} \in \mathbf{R}^{I_N \times R_N}$ are the orthogonal matrices and $\mathcal{B} \in \mathbf{R}^{R_1 \times R_2 \times \cdots \times R_N}$ is approximated core tensor. The calculation algorithm of this decomposition is as follows [9]:

Step 1 Calculate orthogonal matrices $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \cdots, \mathbf{U}^{(N)}$ for a given tensor $\mathcal{A}$ by HOSVD, and rename these matrices as $\mathbf{U}_0^{(n)} (1 \le n \le N)$.

Step 2 Set the every element of the i-th column vectors of matrices $\mathbf{U}_0^{(n)} (2 \le n \le N)$ to zero, where $i > R_n$. Repeat following calculation to obtain the new matrices one after another.

$$\widetilde{\mathbf{U}}_{k+1}^{(1)} = \mathcal{A} \times_2 \mathbf{U}_k^{(2)T} \times_3 \mathbf{U}_k^{(3)T} \cdots \times_N \mathbf{U}_k^{(N)T}, \quad (26)$$

$$\widetilde{\mathbf{U}}_{k+1}^{(2)} = \mathcal{A} \times_1 \mathbf{U}_{k+1}^{(1)T} \times_3 \mathbf{U}_k^{(3)T} \cdots \times_N \mathbf{U}_k^{(N)T}, \quad (27)$$

$$\vdots$$

$$\widetilde{\mathbf{U}}_{k+1}^{(N)} = \mathcal{A} \times_1 \mathbf{U}_{k+1}^{(1)T} \times_2 \mathbf{U}_{k+1}^{(2)T} \cdots \times_{N-1} \mathbf{U}_{k+1}^{(N-1)T}, \quad (28)$$

where $(n - R_n - 1)$ column vectors of the obtained matrices must be set zero.

Repeat Step 2 until the following convergence condition is satisfied.

$$\|\mathcal{B}_k - \mathcal{B}_{k+1}\|^2 < \varepsilon, \quad (29)$$

$$\mathcal{B}_{k+1} = \mathcal{A} \times_1 \mathbf{U}_{k+1}^{(1)T} \times_2 \mathbf{U}_{k+1}^{(2)T} \cdots \times_N \mathbf{U}_{k+1}^{(N)T}. \quad (30)$$

Performing above steps, the best rank-(R_1, R_2, ... , R_N) approximation tensor $\hat{\mathcal{A}}$ can be obtained as

$$\hat{\mathcal{A}} = \mathcal{B}_{k+1} \times_1 \mathbf{U}_{k+1}^{(1)} \times_2 \mathbf{U}_{k+1}^{(2)} \cdots \times_N \mathbf{U}_{k+1}^{(N)}. \quad (31)$$

E. Best Rank-1 Approximation

As the special case of the best rank-(R_1, R_2, ... , R_N) approximation, the best rank-1 approximation tensor can be obtained by outer product of the vectors $\mathbf{U}^{(1)} \in \mathbf{R}^{I_1}$, $\mathbf{U}^{(2)} \in \mathbf{R}^{I_2}$, …, $\mathbf{U}^{(N)} \in \mathbf{R}^{I_N}$ and core tensor $\mathcal{B} \in \mathbf{R}^1$ as

$$\begin{aligned}\hat{\mathcal{A}} &= b_{111} \times_1 \mathbf{U}_{k+1}^{(1)} \times_2 \mathbf{U}_{k+1}^{(2)} \cdots \times_N \mathbf{U}_{k+1}^{(N)} \\ &= b_{111} (\mathbf{U}_{k+1}^{(1)} \otimes \mathbf{U}_{k+1}^{(2)} \cdots \otimes \mathbf{U}_{k+1}^{(N)}),\end{aligned} \quad (32)$$

where b_{111} is the (1,1,1)-th element of the core tensor $\mathcal{B}$.

IV. Comparison and Experiments

A. 3-TPE and Best Rank-1 Approximation

1) Computation Accuracy

Example 1: We consider the super symmetric 3rd-order tensor $\mathcal{A} \in R^{2 \times 2 \times 2}$ which all the elements are equal to 1 except for $a_{111} = 2$ [9]. The tensor $\mathcal{A}$ is represented by the 1-mode unfolding matrix as

$$\mathbf{A}_{(1)} = \left(\begin{array}{cc|cc} 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{array}\right). \quad (33)$$

Table I shows the resultant expansion coefficients and the expansion vectors of 3-TPE and best rank-1 approximation, where the coefficients of the latter method are the elements of the core tensor $\mathcal{S}$ which were renumbered in ascending order of magnitude.

Since the resulted coefficients and the elements of the vectors by both methods are the same except the sign of the coefficients, we see that 3-TPE by the power method and the repeated best rank-1 approximation method do quite the same expansion.

TABLE I
EXPANSION COEFFICIENTS AND EXPANSION VECTORS OF 3-TPE AND BEST RANK-1 APPROXIMATION

	3-TPE		Best rank-1	
i	Expansion term σ_i	Expansion vectors $\mathbf{u}_i$	Expansion term λ_i	Expansion vectors $\mathbf{U}_i^{(1)}$
1	+3.2560	$\begin{pmatrix}+0.7981\\+0.6025\end{pmatrix}$	-3.2560	$\begin{pmatrix}-0.7981\\-0.6025\end{pmatrix}$
2	+0.5234	$\begin{pmatrix}+0.9186\\-0.3952\end{pmatrix}$	-0.5234	$\begin{pmatrix}-0.9186\\+0.3952\end{pmatrix}$
3	+0.3212	$\begin{pmatrix}-0.0392\\+0.9992\end{pmatrix}$	+0.3212	$\begin{pmatrix}-0.0392\\+0.9992\end{pmatrix}$
4	+0.1286	$\begin{pmatrix}-0.8733\\-0.4872\end{pmatrix}$	-0.1286	$\begin{pmatrix}+0.8733\\+0.4872\end{pmatrix}$
5	+0.0596	$\begin{pmatrix}+0.8252\\-0.5649\end{pmatrix}$	-0.0596	$\begin{pmatrix}-0.8252\\+0.5649\end{pmatrix}$
6	+0.0263	$\begin{pmatrix}+0.1355\\+0.9908\end{pmatrix}$	-0.0263	$\begin{pmatrix}-0.1355\\-0.9908\end{pmatrix}$
7	+0.0118	$\begin{pmatrix}-0.9469\\-0.3217\end{pmatrix}$	-0.0118	$\begin{pmatrix}+0.9469\\+0.3217\end{pmatrix}$
8	+0.0052	$\begin{pmatrix}-0.7112\\+0.7030\end{pmatrix}$	-0.0052	$\begin{pmatrix}+0.7112\\-0.7030\end{pmatrix}$
9	+0.0023	$\begin{pmatrix}+0.3107\\+0.9505\end{pmatrix}$	-0.0023	$\begin{pmatrix}-0.3107\\-0.9505\end{pmatrix}$
10	+0.0011	$\begin{pmatrix}-0.9891\\-0.1471\end{pmatrix}$	-0.0011	$\begin{pmatrix}+0.9891\\+0.1471\end{pmatrix}$

2) Computation Time

Example 2: We consider the following magnitude specification $\mathbf{h}_d(x_i, y_j, z_k)$ of a 3-D digital filter design problem [2].

$$\mathbf{h}_d(x_i, y_j, z_k) = \begin{cases} 1, & (0 \le r \le 0.4) \\ \dfrac{(0.6-r)}{0.2}, & (0.4 \le r \le 0.6) \\ 0, & (r \ge 0.6), \end{cases} \tag{34}$$

where,

$$r = \frac{1}{\pi}\sqrt{x_i^2 + y_j^2 + z_k^2}, \qquad x_i = \frac{i\pi}{L'-1}, (0 \le i \le L'-1),$$
$$y_j = \frac{j\pi}{M'-1}, (0 \le j \le M'-1), \quad z_k = \frac{k\pi}{N'-1}, (0 \le k \le N'-1).$$

The elements a_{ijk} of a 3rd-order tensor $\mathcal{A}$ is given by

$$a_{ijk} = \mathbf{h}_d(x_i, y_j, z_k). \tag{35}$$

Since the magnitude specification $\mathbf{h}_d(x_i, y_j, z_k)$ is zero when $r \ge 0.6$, the size of $\mathcal{A}$ can be reduced to $L \times M \times N$, where $L = L' \times 0.6,\ M = M' \times 0.6,\ N = N' \times 0.6.$

In Fig. 3 the computation time of both methods to calculate 5 terms are plotted when L=M=N=2, 3, … , 13, 18, 24. From the figure, we see that 3-TPE can reduce the computation time considerably as compared with the best rank-1 method. Because the best rank-1 approximation which repeats SVD many times takes a lot of time.

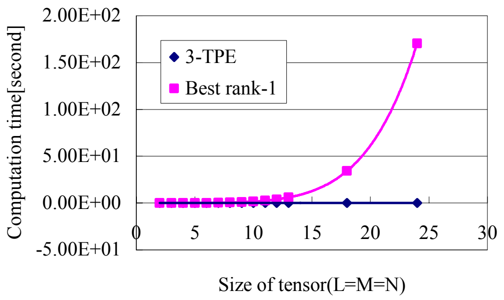

Fig. 3 Computation time of 3-TPE and best rank-1

B. 3-OTPE and HOSVD

1) Computation Accuracy

The expansion coefficients and the expansion vectors are calculated for the example 2 by 3-OTPE and HOSVD. The size of the tensor $\mathcal{A}$ is fixed to $L=M=N=3$ ($L'=M'=N'=5$).

In the Table II, i-th column vectors of $\mathbf{U}$ and $\mathbf{U}^{(1)T}$ are denoted by $\mathbf{u}_i$ and $\mathbf{U}_i^{(1)T}$ respectively. From the table we see that although both methods have the orthogonally regulation, the expansion vectors of 3-OTPE are different to the column vectors of the matrices of HOSVD.

TABLE II
EXPANSION VECTORS BY 3-OTPE AND COLUMN VECTORS OF THE ORTHOGONAL MATRICES BY HOSVD

i	$\mathbf{u}_i$	$\mathbf{U}_i^{(1)T}$
1	$\begin{pmatrix} +6.869E-01 \\ +6.223E-01 \\ +3.754E-01 \end{pmatrix}$	$\begin{pmatrix} -6.908E-01 \\ -6.248E-01 \\ -3.639E-01 \end{pmatrix}$
2	$\begin{pmatrix} +3.221E-01 \\ +2.023E-01 \\ -9.248E-01 \end{pmatrix}$	$\begin{pmatrix} -4.529E-01 \\ -1.836E-02 \\ +8.913E-01 \end{pmatrix}$
3	$\begin{pmatrix} +6.515E-01 \\ -7.562E-01 \\ +6.149E-01 \end{pmatrix}$	$\begin{pmatrix} +5.636E-01 \\ -7.805E-01 \\ +2.703E-01 \end{pmatrix}$

In this case the orthogonal matrices $\mathbf{V}$ and $\mathbf{W}$ by 3-OTPE are obtained as

$$\mathbf{U} = (\mathbf{u}_1 \quad \mathbf{u}_2 \quad \mathbf{u}_3),$$
$$\mathbf{V} = (\mathbf{u}_1 \quad -\mathbf{u}_2 \quad \mathbf{u}_3), \mathbf{W} = (\mathbf{u}_1 \quad \mathbf{u}_2 \quad -\mathbf{u}_3).$$

Similarly the orthogonal matrices $\mathbf{U}^{(2)}$ and $\mathbf{U}^{(3)}$ by HOSVD are obtained as

$$\mathbf{U}^{(1)} = \begin{pmatrix} \mathbf{U}_1^{(1)} \\ \mathbf{U}_2^{(1)} \\ \mathbf{U}_3^{(1)} \end{pmatrix} = \mathbf{U}^{(2)} = \mathbf{U}^{(3)}.$$

2) Expansion Coefficients

By using the orthogonal vectors obtained above, the expansion coefficients are calculated. In the table III and table IV the resulted coefficients $\sigma_{i_1 i_2 i_3}$ by 3-OTPE and $s_{i_1 i_2 i_3}$ by HOSVD are listed in the ascending order of magnitude. The residuals are defined by $\| \mathcal{A} - \mathcal{A}_j \|$ where $\mathcal{A}_j$ is a following j-th expansion term.

$$\mathcal{A}_j = \sum_{i=1}^{j} \sigma_{i_1 i_2 i_3} (\mathbf{u}_{i_1} \otimes \mathbf{v}_{i_2} \otimes \mathbf{w}_{i_3}). \tag{36}$$

These residuals are listed in the tables together with the coefficients.

TABLE III
EXPANSION COEFFICIENTS AND RESIDUALS BY 3-OTPE

3-OTPE			
j	$\sigma_{i_1 i_2 i_3}$	residual	index (i_1, i_2, i_3)
1	3.8007E+00	2.6073E-01	σ(1,1,1)
2	-5.0980E-01	2.2630E-01	σ(1,2,1)
3	-5.0980E-01	1.8559E-01	σ(1,3,1)
4	-5.0980E-01	1.3295E-01	σ(2,1,1)
5	3.4057E-01	1.0096E-01	σ(2,2,1)
6	-1.6093E-01	9.2313E-02	σ(2,3,1)
7	-1.6093E-01	8.2770E-02	σ(3,1,1)
8	-1.6093E-01	7.1972E-02	σ(3,2,1)
9	-1.6093E-01	5.9238E-02	σ(3,3,1)
10	-1.6093E-01	4.2875E-02	σ(1,1,2)
11	-1.6093E-01	1.2937E-02	σ(1,2,2)
12	2.7371E-02	1.0910E-02	σ(1,3,2)
13	2.7371E-02	8.4075E-03	σ(2,1,2)
14	2.7371E-02	4.7275E-03	σ(2,2,2)
15	-1.0658E-02	3.8755E-03	σ(2,3,2)

TABLE IV
EXPANSION COEFFICIENTS AND RESIDUALS BY HOSVD

HOSVD			
j	$s_{i_1 i_2 i_3}$	residual	index (i_1, i_2, i_3)
1	-3.7996E+00	2.6177E-01	σ(1,1,1)
2	5.5640E-01	2.2034E-01	σ(2,1,2)
3	5.5640E-01	1.6904E-01	σ(1,2,2)
4	5.5640E-01	9.2739E-02	σ(2,2,1)
5	2.9316E-01	5.5276E-02	σ(2,2,2)
6	7.6487E-02	5.1749E-02	σ(3,2,2)
7	7.6487E-02	4.7963E-02	σ(2,3,2)
8	7.6487E-02	4.3852E-02	σ(2,2,3)
9	-7.0743E-02	4.0001E-02	σ(1,3,3)
10	-7.0743E-02	3.5738E-02	σ(3,3,1)
11	-7.0743E-02	3.0892E-02	σ(3,1,3)
12	5.8833E-02	2.7037E-02	σ(1,2,1)
13	5.8833E-02	2.2532E-02	σ(2,1,1)
14	5.8833E-02	1.6863E-02	σ(1,1,2)
15	-2.5177E-02	1.5603E-02	σ(3,2,1)

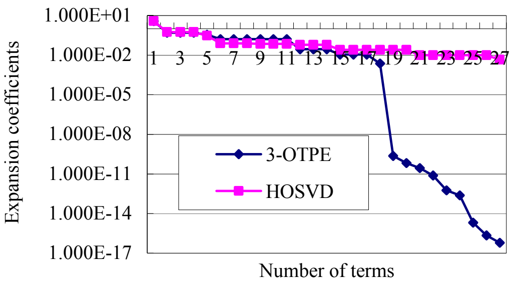

Fig. 4 Expansion coefficients by 3-OTPE and HOSVD

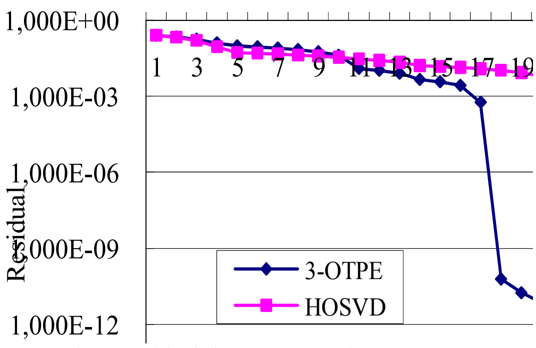

Fig. 5 Residuals by 3-OTPE and HOSVD

All of the coefficients and residuals are plotted in Fig. 4 and Fig. 5. We compared the accuracy of calculation and the computing time for decomposition both of methods. Because the decomposition definition with the product of the vectors is common, both methods can expand a given tensor to a sum of $L \times M \times N$ rank-1 tensors without residual. Fig. 5 shows this fact apparently. We see also that when breaks off expansion calculation on the way, the residuals of 3-OTPE are smaller than HOSVD and that the convergence of 3-OTPE is earlier than HOSVD.

It is guaranteed to obtain the best approximation by the first term of 3-OTPE for given 3rd-order tensor thoroughly same to TPE. Since second term, 3-OTPE obtains the better approximation of the residual tensor when we impose an orthogonal condition. But HOSVD uses all 27 term to approximate for a given data, and does not consider the residuals of expansion on the way.

3) *Computation Time*

Fig. 6 shows the computation time of 3-OTPE and HOSVD when the size of tensor L=M=N=6, 12, 18, 24, 30, 36, 42, 48 for example 2.

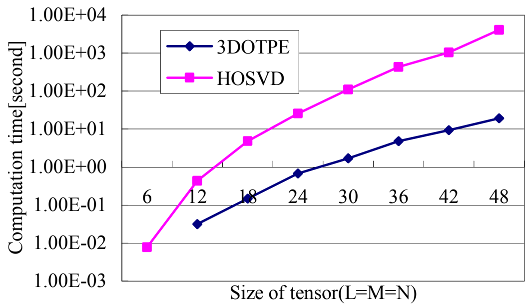

Fig. 6 Computation time of 3-OTPE and HOSVD

From this figure we see that HOSVD requires a lot of computation time compared with 3-OTPE. Since the SVD which spends much computation time [11] is performed for 3 unfolding matrices of the tensor, the difference of computation time comes out.

V. CONCLUSION

We compared the method 3-OTPE to HOSVD for a given 3rd-order tensor. It was confirmed that the computing of OTPE was greatly fast and the accuracy of decomposition on the case of 3rd-order tensor is the same as HOSVD or better.

HOSVD can approximate the low level data very well for the specified rank by the best rank-(R_1, R_2, …, R_N) approximation. Recently OTPE can not get results as good as HOSVD, we will improve OTPE in our future work. Moreover, another one of our future work is to make the OTPE calculation toward higher dimension, which will easily achieve, we think.

We confirmed that the calculation speed of TPE by using the power method is much higher than HOSVD by best rank-1 approximation. In addition, we think our method has advantages for application because we can add the condition to decompose by the product of the vectors which have nonnegative values.

ACKNOWLEDGMENT

The authors would like to thank Dr. Sun Ningping for her generous support.

REFERENCES

[1] Tian bo Deng and Masayuki Kawamata: Design of Two-Dimensional Recursive Digital Filters Based on the Iterative Singular Value Decomposition, Transactions of the Institute of Electronics, Information and Communication Engineers, Vol.E 73, No.6, pp.882-892, 1990.
[2] Makoto Ohki and Masayuki Kawamata: Design of Three-Dimensional Digital Filters Based on the Outer Product Expansion, IEEE Transactions on circuits and Systems, Vol.CAS-37, No.9, pp.1164-1167, 1990.
[3] Takahiro Saitoh, Takashi Komatsu, Hiroshi Harashima, and Hiroshi Miyakawa: Still Picture Coding by Multi-Dimensional Outer Product Expansion (in Japanese), Transactions of the Institute of Electronics, Information and Communication Engineers, Vol.J68-B, No.4, pp.547-548, 1985.
[4] Jun Murakami, Naoki Yamamoto, and Yoshiaki Tadokoro: High-Speed Computation of 3D Tensor Product Expansion by the Power Method, Electronics and Communications in Japan, Part 3, Vol.85, pp.63-72, 2002.
[5] Chiharu Okuma, Jun Murakami, and Naoki Yamamoto: Calculation of 3-D Nonnegative Outer Product Expansion by the Power Method and Its Application to Digital Signal Processing, Proceeding of 12th International Symposium on Artificial Life and Robotics, GS14-4, 2007.
[6] Manolis G. Vozalis and Konstantinos G. Margaritis: Applying SVD on Generalized Item-based Filtering, International Journal of Computer Science & Applications, Vol.3, Issue 3, pp.27-51, 2006.
[7] Berkant Savas and Lars Eldén: Handwritten Digit Classification using Higher order Singular Value Decomposition, Pattern Recognition, Vol.40, pp.993-1003, 2007.
[8] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle: A Multilinear Singular Value Decomposition, SIAM Journal on Matrix Analysis and Applications, Vol.21, No.4, pp.1253- 1278, 2000.
[9] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle: On the Best Rank-1 And Rank-(R1,R2, … ,RN) Approximation Of Higher-Order Tensors, SIAM Journal on Matrix Analysis and Applications., Vol.21, No4, pp.1324-1342,2000.
[10] William H. Press, William T. Vetterling, Saul A. Teukolsky, and Brian P. Flannery: Numerical Recipes in C, Cambridge University Press, 1988.
[11] Michael W. Berry, Susan T. Dumais, and Gavin W. O'Brien: Using Linear Algebra for Intelligent Information Retrieval, SIAM Review, Vol.37, No.4, pp.573-595, 1995.

Chiharu OKUMA received the BE degree in Electrical Engineering from Kyushu Institute of Technology, JAPAN, in 1994, and she is currently PhD candidate in Faculty of Sciences, Kumamoto University.

Since 1994 she is working as Assistant Professor of Department of Information and Computer Sciences, Kumamoto National College of Technology. And her research area of interest is Numerical Calculation and Digital Signal Processing.

Ms. Okuma is a student member of the Institute of Electronics, Information and Communication Engineers (IEICE).

Jun MURAKAMI received the BE and ME degrees in Information Engineering, both from Toyohashi University of Technology, Japan, in 1982 and 1984. And he received the PhD degree in Engineering from Toyohashi University of Technology, in 2000.

He is Professor and the dean of the department of Information and Computer Sciences, Kumamoto National College of Technology. And his research area of interest is Numerical Calculation, Digital Signal Processing, and Human Interface.

Dr. Murakami is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) and the Information Processing Society of Japan (IPSJ).

Naoki YAMAMOTO received the BE and ME degrees in Information Engineering, both from Toyohashi University of Technology, Japan, in 1991 and 1993. And he received the PhD degree in Engineering from Kyushu Institute of Technology, in 2001.

He is working as Associate Professor of the department of Information and Communication Engineering, Kumamoto National College of Technology. And his research area of interest is Numerical Calculation, Digital Signal Processing, and Image Processing.

Dr. Yamamoto is a member of the Institute of Electronics, Information and Communication Engineers (IEICE).