International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:4, No:2, 2010

An Improved Algorithm for Calculation of the Third-order Orthogonal Tensor Product Expansion by Using Singular Value Decomposition

Chiharu Okuma, Naoki Yamamoto, Jun Murakami

Abstract**—**As a method of expanding a higher-order tensor data to tensor products of vectors we have proposed the *Third-order Orthogonal Tensor Product Expansion* (3OTPE) that did similar expansion as *Higher-Order Singular Value Decomposition* (HOSVD). In this paper we provide a computation algorithm to improve our previous method, in which SVD is applied to the matrix that constituted by the contraction of original tensor data and one of the expansion vector obtained. The residual of the improved method is smaller than the previous method, truncating the expanding tensor products to the same number of terms. Moreover, the residual is smaller than HOSVD when applying to color image data. It is able to be confirmed that the computing time of improved method is the same as the previous method and considerably better than HOSVD.

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

I. Introduction

OUR *Third-order Orthogonal Tensor Product Expansion* (3OTPE) [1], [2] and *Higher-Order Singular Value Decomposition* (HOSVD) [3] are both proposed as the expansion technique for multidimensional data to the sum of low rank data, and are the method to do the same kind of expansion. These methods express the multidimensional data as a tensor, and expand that data into the sum of tensor products of vectors, where expansion terms should satisfy the orthogonally each other. The volume of the original multidimensional data can be reduced by truncating this expansion equation in a suitable number of terms. It means a complicated data can be made of concise form.

In addition, each expansion can be expressed in simple form to tensor product of one dimensional data. These properties are the reasons why both expansion methods are effective in fields of the pattern recognition and the digital signal processing, etc [4], [5].

Manuscript received December 28, 2009

Chiharu Okuma is with the Department of Human-Oriented Information Systems Engineering, 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@kumamoto-nct.ac.jp).

Naoki Yamamoto is with the Department of Human-Oriented Information Systems Engineering, Kumamoto National College of Technology, Japan (e-mail: naoki@kumamoto-nct.ac.jp).

Jun Murakami is with the Department of Human-Oriented Information Systems Engineering, Kumamoto National College of Technology, Japan (e-mail: jun@kumamoto-nct.ac.jp).

There are our research results concerning the computation accuracy and the computing time for third-order tensors about these two expansion method [1]. According to these results, superiority or inferiority is not indiscriminately decided for the accuracy of calculation though our method is superior at the computing time. Also in each expansion method, it agrees to the original tensor data by adding $L \cdot M \cdot N$ expansion terms finally when the size of the third order tensor is $L \times M \times N$.

The expansion equation of 3OTPE is approximated by truncating after the expansion terms are permuted in descending order of norm in practical use. Comparing the number of terms that the approximated equation converges to small enough value, 3OTPE is better than HOSVD. However, 3OTPE does not necessarily show a good tendency about the decreasing tendency of the residual before reaching the converging term.

The power method well known as a numerical calculation method of eigenvectors of a matrix [6] is repeatedly used for the calculation of 3OTPE. The 3OTPE algorithm is improved by using SVD [7] besides the power method together. In addition, after calculating min (L, M, N) terms by the algorithm that uses the power method, the improved algorithm uses SVD in order to obtain good computation accuracy. As a result, it is able to be confirmed that the computation accuracy of the proposed method is improved more than previous method. On the other hand, the improvement concerning the computing time is not able to be expected. Especially, we know that since the proposed method dose not require such an additional calculation that is necessary to calculate the remaining expansion vectors by using Gram-Schmidt orthgonalization algorithm [7] after the calculation of min(L, M, N) terms when the size of the tensor is not L=M=N, the method improves accuracy better than the previous method.

II. Third-order Orthogonal Tensor Product Expansion and Higher-Order SVD

A. Representation 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 third-order tensor.

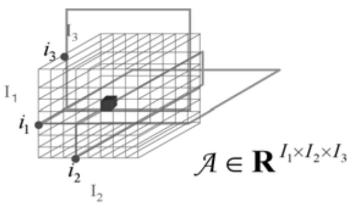


Fig. 1 Image of a third-order tensor

B. Third-order Orthogonal Tensor Product Expansion by the Power Method (3OTPE)

1) Representation of 3OTPE

By applying the Third-order Orthogonal Tensor Product Expansion (3OTPE), a $L \times M \times N$ third-order tensor $\mathcal{A}$ can be expanded as

$$\mathcal{A} = \sum_{i,j,k} \sigma_{ijk} (\mathbf{u}_i \otimes \mathbf{v}_j \otimes \mathbf{w}_k), \tag{1}$$

$$(i = 1,2,\cdots L,\ j = 1,2,\cdots M,\ k = 1,2,\cdots N),$$

where, $\otimes$ denotes the outer product operation, the expansion vectors $\mathbf{u}_i$, $\mathbf{v}_j$, and $\mathbf{w}_k$ correspond to the singular vectors of the SVD for matrices [2]. Each of the expansion vectors is normalized so that its norm is 1 and has a following relation mutually.

$$\begin{aligned}&(\mathbf{u}_j \otimes \mathbf{v}_j \otimes \mathbf{w}_j)(\mathbf{u}_k \otimes \mathbf{v}_k \otimes \mathbf{w}_k) \\ &= (\mathbf{u}_j^T \mathbf{u}_k)(\mathbf{v}_j^T \mathbf{v}_k)(\mathbf{w}_j^T \mathbf{w}_k) = 0,\ j \neq k.\end{aligned} \tag{2}$$

The expansion coefficients σ_{ijk} correspond to the singular value, and they can be calculated for every combination of *i*, *j* and *k* as

$$\sigma_{ijk} = \mathcal{A}(\mathbf{u}_i \otimes \mathbf{v}_j \otimes \mathbf{w}_k), \tag{3}$$

$$(i = 1,2,\cdots L,\ j = 1,2,\cdots M,\ k = 1,2,\cdots N).$$

2) Calculation Algorithm for 3OTPE

When the size of a tensor $\mathcal{A}$ is assumed to $L \times M \times N$, where *m*= $\min(L, M, N)$, the expansion vectors $\mathbf{u}_1, \cdots, \mathbf{u}_m$, $\mathbf{v}_1, \cdots, \mathbf{v}_m$ and $\mathbf{w}_1, \cdots, \mathbf{w}_m$ are obtained by the Algorithm I that is shown later. Because the algorithm uses the Gram-Schmidt orthogonalization process these vectors have the orthogonally mutually.

In case that *L*, *M*, and *N* are same, we can obtain a whole $L \cdot M \cdot N$ of orthonormal vectors for the $L \times M \times N$ tensor. Since the numbers of vectors for one or two dimension are larger than *m* in another case that *L*, *M*, and *N* are different, we need to use Gram-Schmidt process again based on the previously obtained vectors from Algorithm I to calculate the remaining vectors.

For example, in case of *L*>*m*, to calculate the remaining vectors $\mathbf{u}_{m+1}, \mathbf{u}_{m+2}, \cdots, \mathbf{u}_L$, *n* in equation (15) is increased from *m+1* to *L* one by one and the Gram-Schmidt process described later is carried out, where the initial vector $\mathbf{u}_n$ is given arbitrarily. Likewise, other remaining vectors $\mathbf{v}_{m+1}, \mathbf{v}_{m+2}, \cdots, \mathbf{v}_M$ and $\mathbf{w}_{m+1}, \mathbf{w}_{m+2}, \cdots, \mathbf{w}_N$ are calculated.

Algorithm I: 3OTPE (by the power method)

IN: tensor $\mathcal{A}$ ($L \times M \times N$), *m* = min(*L*, *M*, *N*)

OUT: vectors $\mathbf{u}_1, \cdots, \mathbf{u}_m$, $\mathbf{v}_1, \cdots, \mathbf{v}_m$, $\mathbf{w}_1, \cdots, \mathbf{w}_m$

p=0, *n*=1.

Step1. Initial values: normalized vectors $\mathbf{u}_n^{(p)}$, $\mathbf{v}_n^{(p)}$, $\mathbf{w}_n^{(p)}$.

Step2. Residual Tensor $\mathcal{B}_n$ is obtained by

$$\mathcal{B}_n = \mathcal{A} - \sum_{i=1}^{n-1} \sigma_i (\mathbf{u}_i \otimes \mathbf{v}_i \otimes \mathbf{w}_i). \tag{4}$$

Step3. Iterate the power method as follows.

- ♦ Contraction of tensor: $\mathbf{F} = \mathcal{B}_n \mathbf{w}_n^{(p)}$ (5)

  $\mathbf{F}$ is $L \times M$ matrix and its (i, j) th element are denoted as

  $$\mathbf{F}(i, j) = \sum_k \mathcal{B}_n(i, j, k) \mathbf{w}_n^{(p)}(k)\ . \tag{6}$$

  $$\text{Set } \mathbf{u}_n^{(p+1)} = \mathbf{F}\mathbf{v}_n^{(p)}, \mathbf{v}_n^{(p+1)} = \mathbf{F}^T \mathbf{u}_n^{(p+1)}. \tag{7}$$

  Gram-Schmidt process: $\mathbf{u}_n^{(p+1)}$, $\mathbf{v}_n^{(p+1)}$

  $$\begin{aligned}\mathbf{u'}_n^{(p+1)} &= \mathbf{u}_n^{(p+1)} - (\mathbf{u}_1^T \mathbf{u}_n^{(p+1)})\mathbf{u}_1 - \\ &(\mathbf{u}_2^T \mathbf{u}_n^{(p+1)})\mathbf{u}_2 - \cdots - (\mathbf{u}_{n-1}^T \mathbf{u}_n^{(p+1)})\mathbf{u}_{n-1},\end{aligned} \tag{8}$$

  $$\mathbf{u}_n^{(p+1)} = \left\| \mathbf{u'}_n^{(p+1)} \right\|.$$

- ♦ Contraction of tensor: $\mathbf{G} = \mathcal{B}_n \mathbf{v}_n^{(p+1)}$ (9)

  ($M \times N$ matrix)

  $$\text{Set } \mathbf{w}_n^{(p+1)} = \mathbf{G}\mathbf{u}_n^{(p+1)}, \mathbf{u}_n^{(p+1)} = \mathbf{G}^T \mathbf{w}_n^{(p+1)}. \tag{10}$$

  Gram-Schmidt process: $\mathbf{w}_n^{(p+1)}$, $\mathbf{u}_n^{(p+1)}$

- ♦ Contraction of tensor: $\mathbf{H} = \mathcal{B}_n \mathbf{u}_n^{(p+1)}$ (11)

  ($N \times L$ matrix)

  $$\text{Set } \mathbf{v}_n^{(p+1)} = \mathbf{H}\mathbf{u}_n^{(p+1)}, \mathbf{w}_n^{(p+1)} = \mathbf{H}^T \mathbf{w}_n^{(p+1)}. \tag{12}$$

  Gram-Schmidt process: $\mathbf{v}_n^{(p+1)}$, $\mathbf{w}_n^{(p+1)}$

  Convergence conditions:

  $$\begin{cases} \left\| \mathbf{u}_n^{(p+1)} - \mathbf{u}_n^{(p)} \right\| < \varepsilon, \\ \left\| \mathbf{v}_n^{(p+1)} - \mathbf{v}_n^{(p)} \right\| < \varepsilon, \\ \left\| \mathbf{w}_n^{(p+1)} - \mathbf{w}_n^{(p)} \right\| < \varepsilon, \end{cases} \tag{13}$$

  where, ε is a small enough value.

  p=*p*+1, return to step3.

Step4. $\mathbf{u}_n = \mathbf{u}_n^{(p+1)}$, $\mathbf{v}_n = \mathbf{v}_n^{(p+1)}$, $\mathbf{w}_n = \mathbf{w}_n^{(p+1)}$, which are called *n*th expansion vectors.

Step5. *n*th expansion coefficient σ_n is obtained by performing an inner product operation as

$$\sigma_n = \mathcal{B}_n (\mathbf{u}_n \otimes \mathbf{v}_n \otimes \mathbf{w}_n). \tag{14}$$

n=*n*+1, *p*=0, return to step1.

Gram-Schmidt process:

Along with the Gram-Schmidt orthgonalization algorithm, calculate the vector $\mathbf{u}_n^{'}$ by subtracting the previously obtained terms from vector $\mathbf{u}_n$ as,

$$\begin{aligned}\mathbf{u}_n^{'} &= \mathbf{u}_n - (\mathbf{u}_1^T\mathbf{u}_n)\mathbf{u}_1 - (\mathbf{u}_2^T\mathbf{u}_n)\mathbf{u}_2 \cdots \\ &\quad - (\mathbf{u}_{n-1}^T\mathbf{u}_n)\mathbf{u}_{n-1}.\end{aligned} \tag{15}$$

Set $\mathbf{u}_n = \left\|\mathbf{u}_n^{'}\right\|$.

C. Higher-Order SVD(HOSVD)

1) Unfolding Matrices of Nth-Order Tensor

A higher-order tensor is represented by some matrices (second-order tensor) which are called unfolding matrices [3]. By using this representation, an nth-order tensor $\mathcal{A} \in \mathbf{R}^{I_x \times I_2 \times \cdots \times I_n}$ is unfolded to 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 third-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 Figure 2.

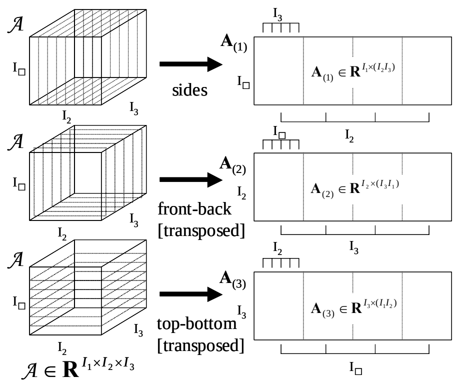


Fig. 2 Unfolding of the third-order tensor $\mathcal{A} \in \mathbf{R}^{I_1 \times I_2 \times I_3}$ to matrices $\mathbf{A}_{(1)} \in \mathbf{R}^{I_1 \times (I_2 I_3)}$, $\mathbf{A}_{(2)} \in \mathbf{R}^{I_2 \times (I_3 I_1)}$, and $\mathbf{A}_{(3)} \in \mathbf{R}^{I_3 \times (I_1 I_2)}$.

2) n-Mode Product

Each unfolding matrix can be decomposed by SVD as follows

$$\mathbf{A}_{(n)} = \mathbf{U}^{(n)} \mathbf{\Sigma}^{(n)} \mathbf{V}^{(n)T}, \tag{16}$$

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

The n-mode product of a tensor $\mathcal{A}$ and 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 matrix is represented as

$$(\mathcal{A} \times_n \mathbf{U})_{i_1 i_2 \cdots i_{n-1} j_n i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} a_{i_1 i_2 \cdots i_{n-1} i_n i_{n+1} \cdots i_N} u_{j_n i_n}. \tag{17}$$

By using this n-mode product representation, equation (16) can be rewritten as

$$\mathbf{A}_{(n)} = \mathbf{\Sigma}^{(n)} \times_1 \mathbf{U}^{(n)} \times_2 \mathbf{V}^{(n)}. \tag{18}$$

3) Calculation Algorithm for HOSVD

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

$$\begin{aligned}\mathcal{A} &= \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \cdots \times_N \mathbf{U}^{(N)} \\ &= \sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} s_{i_1 i_2 \cdots i_N} \mathbf{U}_{i_1}^{(1)} \otimes \mathbf{U}_{i_2}^{(2)} \otimes \cdots \otimes \mathbf{U}_{i_n}^{(N)},\end{aligned} \tag{19}$$

where the matrices $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \cdots, \mathbf{U}^{(N)}$ are orthogonal matrices obtained by applying SVD to the n-mode unfolding matrices, $\mathbf{U}_{i_1}^{(1)}, \mathbf{U}_{i_2}^{(2)}, \cdots, \mathbf{U}_{i_n}^{(N)}$ are the column vectors $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \cdots, \mathbf{U}^{(N)}$ respectively. The tensor $\mathcal{S}$ in this equation 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

$$\mathcal{S} = \mathcal{A} \times_1 \mathbf{U}^{(1)T} \times_2 \mathbf{U}^{(2)T} \cdots \times_N \mathbf{U}^{(N)T}. \tag{20}$$

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

The calculation algorithm for HOSVD is shown in Algorithm II [3].

Algorithm II: HOSVD

IN: tensor $\mathcal{A} \in \mathbf{R}^{I_x \times I_2 \times \cdots \times I_n}$

OUT: vectors $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \cdots, \mathbf{U}^{(N)}$

Step1. Matrix unfolding

$$\mathbf{A}_{(n)} \in \mathbf{R}^{I_n \times (I_{n+1} I_{n+2} \cdots I_N I_1 I_2 \cdots I_{n-1})} \tag{21}$$

Step2. Matrix SVD: $\mathbf{A}_{(n)}$

$$\mathbf{A}_{(n)} = \mathbf{U}^{(n)} \mathbf{\Sigma}^{(n)} \mathbf{V}^{(n)T}, \tag{22}$$

where $\mathbf{U}^{(n)}$ and $\mathbf{V}^{(n)}$ are left and right singular vectors obtained by applying SVD to unfolding matrix $\mathbf{A}_{(n)}$, and $\mathbf{\Sigma}^{(n)}$ is a diagonal matrix which has singular values as the diagonal elements.

Step3. Calculation of core tensor: $\mathcal{S}$

III. Improved Third-order Orthogonal Tensor Product Expansion (Improved 3OTPE)

A. The problem of our previous algorithm

In the calculation algorithm for 3OTPE described subsection *B*, the nth expansion vectors $\mathbf{u}_n$, $\mathbf{v}_n$ and $\mathbf{w}_n$ can be calculated by applying the power method to the matrices that obtained from contraction operation between the tensor $\mathcal{A}$ and iteration vectors $\mathbf{u}_n^{(p)}$, $\mathbf{v}_n^{(p)}$ and $\mathbf{w}_n^{(p)}$. Since the calculation is repeated just m times, where $m = \min(L, M, N)$, additional calculation process namely Gram-Schmidt orthogonalization process is to be needed for the remaining vectors when even one of L, M, and N is larger than m. The Gram-Schmidt process is well known algorithm [7], which gives a set of orthogonal vectors on the vector space for the given set of linear independent vectors. We use the algorithm to calculate the remaining expansion vectors from m orthonormal vector in the vector space of a dimension that is larger than m. While the remaining vector is calculated uniquely when L-m=1, the vectors are not unique when L-m>1 because these vectors depends on the given initial vectors.

For this reason, the problem is that the approximating expansion for the original tensor cannot give the smallest residual during the expanding process by our previous method.

B. Calculation Algorithm for Improved 3OTPE

We improve the previous calculation algorithm for 3OTPE as follows.

First, the expansion vectors $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_m$, $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_m$ and $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_m$ of a given $L \times M \times N$ third-order tensor $\mathcal{A}$ are calculated by Algorithm I that uses the power method as well as the previous method, where $m = \min(L, M, N)$. So the set of vectors $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_N$ can be obtained, assuming that N=m.

Then, the $L \times M$ matrix $\hat{\mathbf{F}}$ are calculated by a contraction operation of the given tensor $\mathcal{A}$ and the expansion vector $\mathbf{w}_1$ as

$$\hat{\mathbf{F}} = \mathcal{A}\mathbf{w}_1. \tag{23}$$

The matrix $\hat{\mathbf{F}}$ is decomposed by the SVD as

$$\hat{\mathbf{F}} = \hat{\mathbf{U}}\Sigma\hat{\mathbf{V}}^T. \tag{24}$$

In this equation, each set of the column vectors of the matrix $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$, namely $\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \cdots, \hat{\mathbf{u}}_L$ and $\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \cdots, \hat{\mathbf{v}}_M$, are the orthogonal set respectively. Finally, we can obtain the expansion vectors of the improved 3OTPE method by renaming those vectors as $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_L$ and $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_M$. There are following relations for the expansion vectors between the improved method and the previous one,

$$\mathbf{u}_1 = \hat{\mathbf{u}}_1, \mathbf{u}_i \neq \hat{\mathbf{u}}_i, i = 2, \cdots, m, \tag{25}$$
$$\mathbf{v}_1 = \hat{\mathbf{v}}_1, \mathbf{v}_i \neq \hat{\mathbf{v}}_i, i = 2, \cdots, m.$$

These relations show that though the vectors in the first term by the improved method are identical with those by the previous one, the vectors in other term are different. We show the improved calculation algorithm for 3OTPE in Algorithm III.

IV. Comparison and Experiments

A. Example of Tensors

To compare the improved method with previous one we treat $L \times M \times N$ tensor classified in the shape depending on the size of tensor as Figure 3. The tensor is classified to cubic type tensor (a) when the size of L and M are all equal to N, otherwise it is classified to non-cubic type tensor (b).

Algorithm III: Improved 3OTPE

IN: tensor $\mathcal{A} \in \mathbf{R}^{I_1 \times I_2 \times I_3}$, $I_1 = L, I_2 = M, I_3 = N$.

OUT: vectors $\mathbf{u}_1, \cdots, \mathbf{u}_L$, $\mathbf{v}_1, \cdots, \mathbf{v}_M$, $\mathbf{w}_1, \cdots, \mathbf{w}_N$.

Step1. 3OTPE by Algorithm I:

IN: $\mathcal{A} \in \mathbf{R}^{I_1 \times I_2 \times I_3}$, $m = \min(L, M, N)$

OUT: m=L: $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_L$,

m=M: $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_M$,

m=N: $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_N$.

Step2. Contraction of tensor : $\hat{\mathbf{F}}$

$$\hat{\mathbf{F}} = \mathcal{A}\mathbf{w}_1, \tag{26}$$

$\mathbf{w}_1$: first expansion vector obtained by step 1, if m=N.

$\hat{\mathbf{F}}$: $L \times M$ matrix, whose (i, j) th element are denoted as

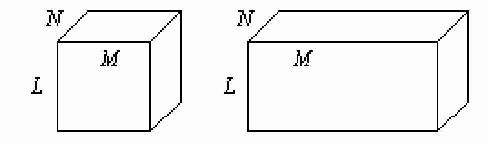
$$\hat{\mathbf{F}}(i, j) = \sum_k \mathcal{A}(i, j, k)\mathbf{w}_1(k). \tag{27}$$

Step3. Matrix SVD: $\hat{\mathbf{F}} = \hat{\mathbf{U}}\Sigma\hat{\mathbf{V}}^T$ (28)

New vectors : $\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \cdots, \hat{\mathbf{u}}_L$, $\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \cdots, \hat{\mathbf{v}}_M$

Set $\mathbf{u}_i = \hat{\mathbf{u}}_i$, $i = 1, 2, \cdots, L$,

$\mathbf{v}_i = \hat{\mathbf{v}}_i$, $i = 1, 2, \cdots, M$.



(a) Cubic type (L=M=N) (b) Non-cubic type (not L=M=N)

Fig. 3 Classification based on the shape of tensors.

As an example, we consider the following magnitude specification $\mathbf{h}_d(x_i, y_j, z_k)$ of a 3-D digital filter design problem [8], [9].

$$\mathbf{h}_d(x_i, y_j, z_k) = \begin{cases} 1, & (0 \le r \le 0.4), \\ \dfrac{(0.6 - r)}{0.2}, & (0.4 \le r \le 0.6), \\ 0, & (r \ge 0.6), \end{cases} \tag{29}$$

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 (i,j,k)th elements a_{ijk} of a third-order tensor $\mathcal{A}$ is given by

$$a_{ijk} = \mathbf{h}_d(x_i, y_j, z_k). \tag{30}$$

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 $L = L' \times 0.6$, $M = M' \times 0.6$, and $N = N' \times 0.6$.

B. *Cubic Type Tensor*

The calculation results both improved method and previous one are shown about a cubic type tensor which the size in three directions are identical.

1) *Expansion Vectors*

Table I shows results of calculated expansion vectors $\mathbf{u}_i$, $\mathbf{v}_i$, and $\mathbf{w}_i$, where i=1, 2, 3, by the both methods, when the size of tensor is L=M=N=3. Both of the value of expansion vectors $\mathbf{v}_i$ and $\mathbf{w}_i$ are equal to the value of $\mathbf{u}_i$ of previous method from the symmetry of the original 3-D data.

The improved method uses the vector $\mathbf{w}_i$ obtained by the previous method as it is to perform a contraction operation to given tensor, the value of that vector is equal to $\mathbf{u}_i$ obtained by the previous method.

TABLE I EXPANSION VECTORS

i	Improved 3OTPE $\mathbf{u}_i$	Previous 3OTPE $\mathbf{u}_i$	HOSVD $\mathbf{U}_i^{(1)T}$
1	$\begin{pmatrix}-6.869E-01\\-6.223E-01\\+3.754E-01\end{pmatrix}$	$\begin{pmatrix}+6.869E-01\\+6.223E-01\\+3.754E-01\end{pmatrix}$	$\begin{pmatrix}-6.908E-01\\-6.248E-01\\-3.639E-01\end{pmatrix}$
2	$\begin{pmatrix}+4.841E-01\\-6.665E-01\\-8.750E-01\end{pmatrix}$	$\begin{pmatrix}+3.221E-01\\+2.023E-01\\-9.248E-01\end{pmatrix}$	$\begin{pmatrix}-4.529E-01\\-1.836E-02\\+8.913E-01\end{pmatrix}$
3	$\begin{pmatrix}+5.420E-01\\-7.828E-01\\+3.059E-01\end{pmatrix}$	$\begin{pmatrix}+6.515E-01\\-7.562E-01\\+6.149E-01\end{pmatrix}$	$\begin{pmatrix}+5.636E-01\\-7.805E-01\\+2.703E-01\end{pmatrix}$

For the comparison of our two methods, we calculated the difference vectors between improved 3OTPE and HOSVD, previous 3OTPE and HOSVD respectively. The relative norms defined by following equation of these difference vectors are shown in Table II.

$$d(\mathbf{u}_i) = \frac{\left\| \mathbf{u}_i - \mathbf{U}_i^{(1)T} \right\|}{\left\| \mathbf{U}_i^{(1)T} \right\|} \tag{31}$$

It is obviously that these values of the first term of the expansion equation by both methods are the same, and that the values by improved method are closer to those by HOSVD in the order of a digit than the previous one after the second term. The vectors $\mathbf{w}_i$ in both methods are equal for the reason mentioned above.

The averaged difference between each set of whole expansion vectors in both our methods and the other set of those vectors in HOSVD is calculated by following equation respectively, and it is listed in the Table II.

$$\frac{1}{L+M+N}\left(\sum_{i=1}^{L} d(\mathbf{u}_i) + \sum_{i=1}^{M} d(\mathbf{v}_i) + \sum_{i=1}^{N} d(\mathbf{w}_i)\right) \tag{32}$$

From the table, we see the difference between the improved method and HOSVD is about 50% smaller than the previous method.

TABLE II THE DIFFERENCE OF EXPANSION VECTORS BETWEEN OUR METHODS AND HOSVD (3,3,3)

Vectors	i	Improved 3OTPE	Previous 3OTPE
$\mathbf{u}_i$	1	1.236E-02	1.236E-02
	2	4.322E-02	2.282E-01
	3	4.166E-02	2.279E-01
$\mathbf{v}_i$	1	1.236E-02	1.236E-02
	2	4.322E-02	2.282E-01
	3	4.166E-02	2.279E-01
$\mathbf{w}_i$	1	1.236E-02	1.236E-02
	2	2.282E-01	2.282E-01
	3	2.279E-01	2.279E-01
Average		7.365E-02	1.561E-01

2) *Residuals*

After all the expansion coefficients are permuted in the descending order of absolute value of magnitude, and ith coefficient is expressed as σ_i. The tensor $\mathcal{A}_n$ obtained by using n coefficients is calculated as follows,

$$\mathcal{A}_n = \sum_{i=1}^{n} \sigma_i (\mathbf{u}_i \otimes \mathbf{v}_i \otimes \mathbf{w}_i), \tag{33}$$
$$\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_i \ge \cdots \ge \sigma_{L \cdot M \cdot N}.$$

The residuals of the expansion calculation in the previous method and the improved one by equation (34) are plotted in the Figure 4. The results by HOSVD are also plotted for the comparison to our methods.

$$\left\| \mathcal{A} - \mathcal{A}_n \right\| = \sqrt{\sum_{i_1,i_2,i_3} \left(\mathcal{A}(i_1,i_2,i_3) - \mathcal{A}_n(i_1,i_2,i_3) \right)^2} \tag{34}$$

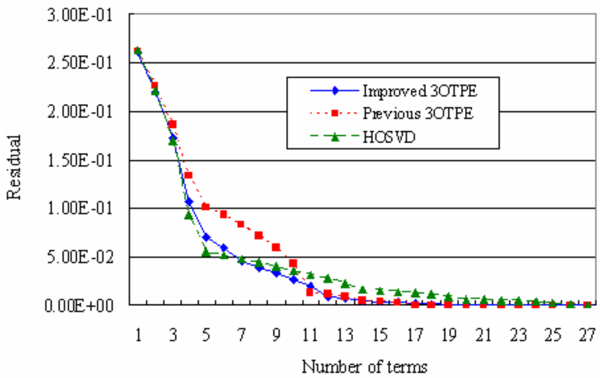


Fig. 4 Residuals of tensor (3, 3, 3)

In each method, the residuals become 0 when the whole of 27 terms in expansion equation are added. The difference appears from the 2nd term, and the residuals of the improved method are smaller as compared with previous method and the improved one. The difference becomes gradually grows, then a remarkable 3.0×10^{-2} or more difference is shown from *5*th term to the 8th term, and it becomes small gradually again at the end.

The comparison between HOSVD and the previous method has already been done [1]. The paper showed that HOSVD was good for first ten terms, and the resultant value by the previous method is smaller after that concerning to the residuals. From above result it can be seen that the residuals approach with these by HOSVD before first some terms, and become smaller than HOSVD after *7*th term by using improved method.

To make the convergence characteristics of the residuals more comprehensible, these residual values are plotted by the logarithm in Figure 5. We see that the residual is almost 0 in the improved method in *19*th terms from this figure. Though the residual by the previous method becomes almost 0 in *18*th terms, the improved method shows smaller residual after that term. We also see that the residual converge to 0 earlier in both our methods than HOSVD.

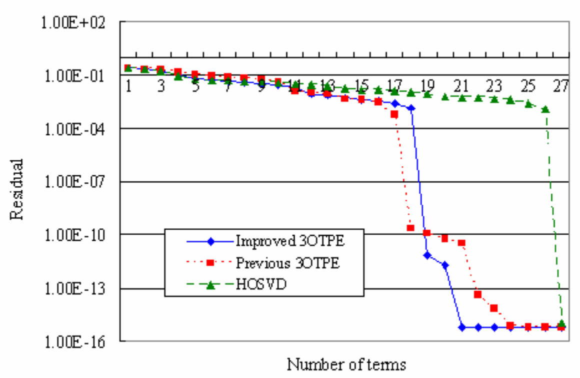


Fig. 5 Residuals of tensor (3, 3, 3) in logarithm

Figure 6 shows the differences of each pair of the best residual and other two residual among the three methods. The above-mentioned results are confirmed from this figure. The averages of differences are 1.85×10^{-3} for improved method, 9.88×10^{-3} for previous one, and 6.44×10^{-3} for HOSVD, and the maximum values are 1.47×10^{-2}, 4.57×10^{-2}, and 1.83×10^{-2}, respectively. The improved method has the best result in three methods about these values.

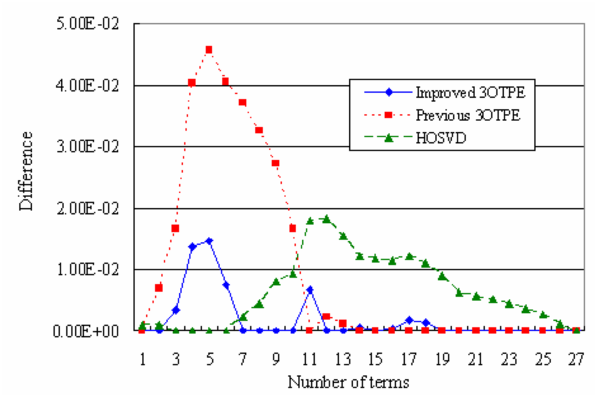


Fig. 6 The comparison with the best residual of Fig. 4.

C. *Non-Cubic Type Tensor*

When the tensor data are not a cubic type pointed out as a problem of previous method, the expansion is calculated about the 3-D digital filter design specification matrix of equation (30) as well as the case of a cubic type tensor data, where the size of the tensor is assumed to be $L=6$, $M=5$, and $N=3$.

1) *Expansion Vectors*

Table III shows the results of comparison between the vectors obtained by the previous method and the improved one with that of HOSVD as well as Table II. From the table, b oth methods give same values about all vectors of the first term of the expansion equation and about the vectors $\mathbf{w}_i$, i=1, 2, 3 as well as the case of the cubic type tensor data. The improved method gives 1 digit smaller values than the previous one in the *2*nd term for the vectors $\mathbf{u}_i$ and $\mathbf{v}_i$, and these values are closer to those by HOSVD. However it is impossible to say which method gives closer results to HOSVD indiscriminately as for the *3*rd or more term, it can be said that the results by improved method nearer to those by HOSVD because of the terms with small number influences more greatly to the expansion equation. Considering the averages of the difference vectors of each expansion vectors of our methods and those of SVD, the improved method shorted the distance of vectors about 8%.

TABLE III The differences of expansion vectors between our methods and HOSVD (6, 5, 3)

Vectors	i	Improved 3OTPE	Previous 3OTPE
$\mathbf{u}_i$	1	1.026E-02	1.026E-02
	2	5.024E-02	1.583E-01
	3	4.313E-01	3.661E-01
	4	4.384E-01	3.114E-01
	5	2.977E-01	5.325E-01
	6	2.030E-01	4.451E-01
$\mathbf{v}_i$	1	1.053E-02	1.053E-02
	2	2.817E-02	1.831E-01
	3	1.086E-01	2.792E-01
	4	7.697E-01	7.162E-01
	5	7.671E-01	6.952E-01
$\mathbf{w}_i$	1	1.000E+00	1.000E+00
	2	1.407E+00	1.407E+00
	3	1.300E+00	1.300E+00
Average		4.873E-01	5.297E-01

2) *Residuals*

Figure 7 shows the residuals between the calculation results of the expansion and the original tensor when the expansion equation is truncated as well as the case of the cubic type tensor data. In this case it can be confirmed that the residuals become 0 when the whole expansion terms are added in any method that are taken up here including HOSVD. The decreasing tendency of the residuals is smoother than the Figure 4, and the mutual differences of the tendency of three methods are not large. The residuals are plotted even by the logarithm in Figure 8. The differences of three methods begin to appear from about 60th term, and the improved method converges earliest, the previous method succeeds, and HOSVD converges lastly to residual 0 without changing in the decreasing tendency of the Figure 4.

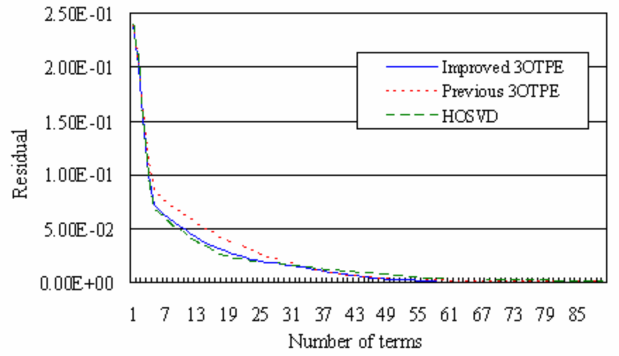


Fig. 7 Residuals of tensor (6, 5, 3)

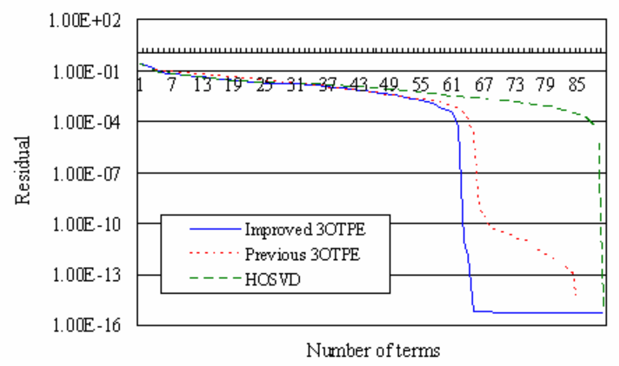


Fig. 8 Residuals of tensor (6, 5, 3) in logarithm

The differences of each pair of the best residual and other two residual among the three methods are plotted in Figure 9 as well as Figure 6. The difference curve of improved method becomes small more than HOSVD from *31*st terms, and it converges after that as the curve changing places into that of the previous one. The average of the differences of the improved method, the previous method, and HOSVD are 1.09×10^{-3}, 4.09×10^{-3}, and 1.30×10^{-3} respectively, so the improved method is the best. Since the maximum value of the differences are 6.06×10^{-3}, 1.76×10^{-2}, and 3.58×10^{-3} respectively in the same order as those, we see that the improved method decreases to about 1/3 compared with the previous one though it is not better than HOSVD.

The average and the maximum value of each method are smaller than the results for the cubic type tensor data in three methods, because the size of the tensor used here is more than that of the example of cubic type tensor. That is to say, more numbers of sample points are given to the example of equation (30) in this case.

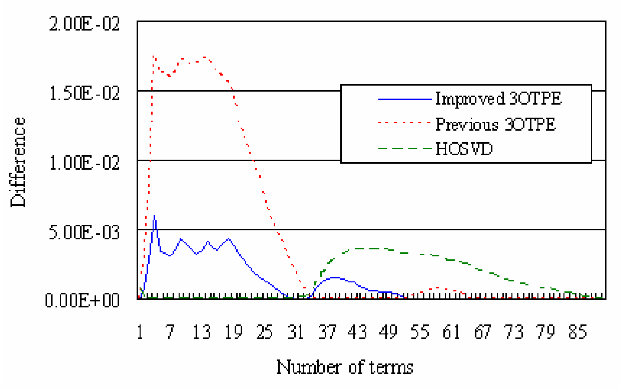


Fig. 9 The comparison with the best residual of Fig. 7

D. Computation Time

Figure 10 shows the computing time required to carry out the expansion by improved method, previous method, and HOSVD changing *L*, *M*, and *N* between from 6 to 60 in case of the cubic type tensor data. Because the algorithm of the improved method needs contraction operations and SVD calculation in addition to the algorithm of previous one, improved method takes the computing time more than the previous one. However, it can be understood that there is only little difference between these two methods, and the improved method can calculate at a higher speed compared with HOSVD. Since the HOSVD is required to calculate SVD for matrices of which the size of a side becomes the order of the square, SVD takes large time to carry out the expansion [10]. It was available to calculate in near 0[sec] when the size of tensor $L=M=N=12$ in our both methods and the size is $L=M=N=6$ in HOSVD. The similar results are yielded for the non-cubic type tensor as well as we described here. In this paper, DELL POWEREDGE 840 is used to carry out numerical calculation.

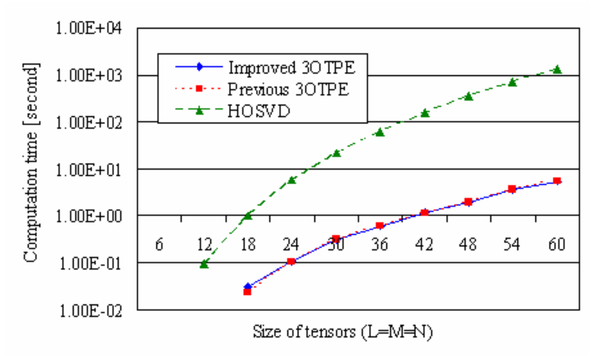


Fig. 10 Computation time

E. Application to Color Image Data

In order to compare the tendency of results with the case of cubic type tensor, the 3-D digital filter specification is used as the non-cubic type tensor data as mentioned above. Here, the three methods are applied to the color image data which is often

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:4, No:2, 2010

used as an example of 3-D data processing. Because a color image consists of RGB images, we treat it as a tensor data whose depth size is 3. Famous standard test image "Lena" [11] which is shown in Figure 11 is used as the example image to be calculated, and the sizes of images are fixed to 16×16, 32×32, and 64×64.

Fig. 11 Standard test image "Lena"

Figure 12 shows the residuals between the original $16 \times 16 \times 3$ tensor and the reconstructed tensor by the three methods when the expansion equation is truncated in progress on the calculation. The improved method is obviously excellent compared to the case of the digital filter of Figure 7, and the method gives the smallest residual from first to last over the whole of expansion. Moreover, it can be confirmed that the improved method converges earliest from the Figure 13, where the residuals are shown by logarithm.

The differences between the smallest residuals and that of each method are plotted in Figure 14. Because the residuals of the improved method are the smallest in all the area, this figure shows the differences with the improved method. In this case, the average of the differences of the previous method is 4.09×10^{-3}, and that of HOSVD is 1.30×10^{-3}. Similarly, the maximum difference of the previous method is 1.76×10^{-2}, and that of HOSVD is 3.58×10^{-3}. So, we see that the residuals of other two methods are large compared with the improved method.

We can see the improved method shows the better results for application to the color image than those for the digital filter design by following reason. The size of the depth direction is fixed with 3 in the color image data, and that is from a few tenth to a few hundredth of the size of the length or that of the width, so that the number of terms calculated by using the Gram-Schmidt orthogonalization process increases in the improved method. The influence of the problem pointed out by in the chapter III appears more greatly for these reasons.

Table IV shows the number of terms where the residual of the three methods becomes 5% or less. It is understood that the improved method becomes residual 5% by the least number of terms in case of any image size. The tensor is constructed by truncating the expansion at this number of terms of improved method here. After each element value of the tensors is made an integral value form 0 to 255, the color images are reconstructed and shown in Table V. The *signal to noise ratio* (*SNR*) of each reconstructed images is indicated in the table. The *SNR* is defined as follows,

$$SNR = 20 \times \log_{10} \frac{Ps}{Pn} \,[\mathrm{dB}], \tag{35}$$

$$Ps = \sum_{i=1}^{L} \sum_{j=1}^{M} \sum_{k=1}^{3} \mathcal{A}(i,j,k)^2,$$

$$Pn = \sum_{i=1}^{L} \sum_{j=1}^{M} \sum_{k=1}^{3} (\mathcal{A}(i,j,k) - \mathcal{A}_n(i,j,k))^2,$$

$\mathcal{A}$: Original image, $\mathcal{A}_n$: Reconstructed image,

where the third-order tensors $\mathcal{A}$ and $\mathcal{A}_n$ represent the original image and the reconstructed image respectively.

From the images shown in the Table V, the improved method has the highest *SNR*, and its *SNR* value is 50[dB] level about the image of all the sizes. We can see that the improved method gives good reconstruction images than other two methods.

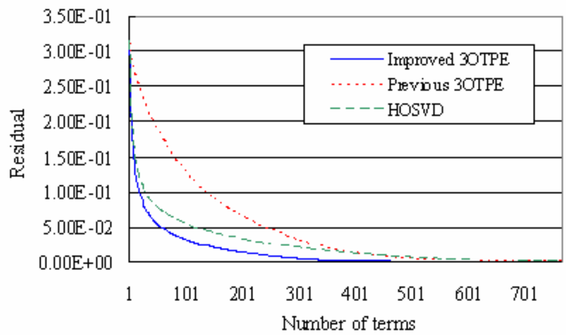


Fig. 12 Residuals of tensor (16, 16, 3)

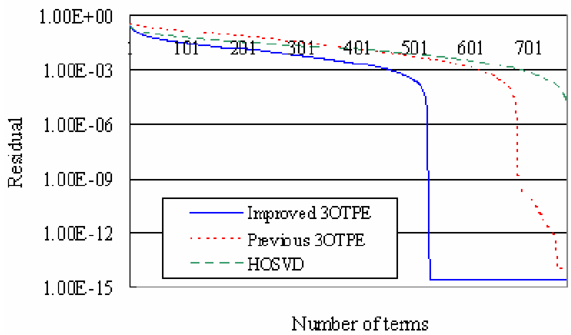


Fig. 13 Residuals of tensor (16, 16, 3) in logarithm

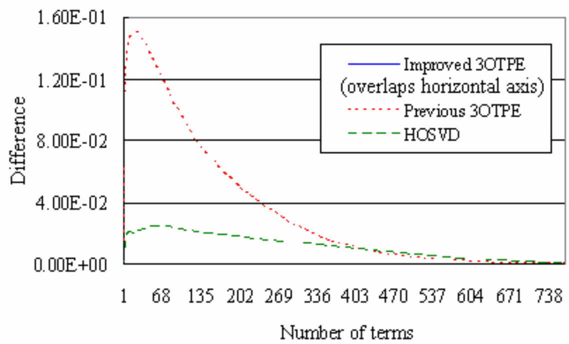
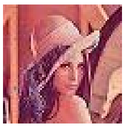
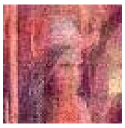
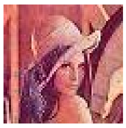


Fig. 14 The comparison with the best residual of Fig. 12

TABLE IV THE NUMBER OF TERMS THAT BECOMES RESIDUAL 5% OR LESS

Size	Improved 3OTPE	Previous 3OTPE	HOSVD
16×16	56	239	113
32×32	180	1108	326
64×64	345	4698	832

TABLE V RECONSTRUCTED IMAGES

Size ($size \times 3$)	Terms	Improved 3OTPE	Previous 3OTPE	HOSVD
16×16 (768)	56	 52.36[dB]	 29.85[dB]	 45.66[dB]
32×32 (3072)	180	 52.33[dB]	 28.77[dB]	 46.85[dB]
64×64 (12288)	345	 52.29[dB]	 26.81[dB]	 45.08[dB]

V. CONCLUSION

Our calculation algorithm of 3OTPE has been improved to overcome the problem concerning the data of non-cubic type tensor mainly in the following ways comparing to previous method and HOSVD.

(1) In the evaluation of a residual, the first term in the evaluation of a residual has almost the same value in the three methods when the expansion equation is truncated or not. Therefore, the expansion vectors $\mathbf{u}_1$, $\mathbf{v}_1$, and $\mathbf{w}_1$ have the almost the same element values. It is different from HOSVD that improved method and previous one have almost equal vectors of $\mathbf{w}_1$ for the non-cubic type tensor. Especially, for the small number of terms which influence on the entire equation, the improved method calculates vectors near to those of HOSVD about the vector since second term.

(2) Considering the residuals, the HOSVD converges earlier than previous method and the number of terms necessary to 0 residual is more than previous method. The improved method converges earlier than HOSVD and the number of terms necessary to 0 residual is less than previous method.

(3) It is confirmed that the above results are similarly in the case of a cubic type tensor and that of a non-cubic type.

(4) When the size of length and width are larger than the size of the depth in a color image, the improved method shows the best results in the three methods.

(5) The improved method almost takes same computing time as previous method, therefore that can could calculate the expansion higher-speed than HOSVD.

Our new 3OTPE method has improved the expansion precision of previous one further and kept the original feature of previous one left. From above results for cubic type tensors and non-cubic ones, we can say that the improved method is the most excellent and effective expansion method in the accuracy of calculation and the computing time for the application of treating 3-D data, that is, the color image processing etc.

Golub-Reinsch algorithm [7], which has been commonly used in numerical calculation method of SVD, is used to compare the characteristics of our method and HOSVD in this paper. It is known that the technique for the speed-up of the computing time of SVD calculation is variously researched [12].

ACKNOWLEDGMENT

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

REFERENCES

[1] Chiharu Okuma, Jun Murakami, and Naoki Yamamoto: Comparison between Higher-order SVD and Third-order Orthogonal Tensor Product Expansion, International Journal Electronics, Communications and Computer Engineering, vol.1, no.2, pp.131-137, 2009.

[2] 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.

[3] 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.

[4] 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.

[5] Berkant Savas and Lars Eldén: Handwritten Digit Classification using Higher order Singular Value Decomposition, Pattern Recognition, Vol.40, pp.993-1003, 2007.

[6] J.H. Wilkinson: The Algebraic Eigenvalue Problem, Oxford Science Publications, 1965.

[7] Gene Howard Golub, Christian Reinsch; Singular Value Decomposition and Least Squares Solutions, Numerische Mathematik14, pp.403-420, 1970.

[8] 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.

[9] 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.

[10] R. L. Johnston: Numerical Methods, John Wiley & Sons, 1982.

[11] Jamie Hutchinson: Culture *Communication, and an Information Age Madonna* IEEE Professional Communication Society Newsletter, Volume 45, No 3, pp. 1, 5–7, 2001.

[12] Taro Konda, Masami Takata, Masashi Iwasaki, and Yoshimasa Nakamura: A new singular value decomposition algorithm suited to parallelization and preliminary results, Proceedings of the 2nd IASTED international conference on Advances in computer science and technology, Puerto Vallarta, Mexico, pp.79-84, 2006.

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 Human-Oriented Information Systems Engineering, 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).

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 Human-Oriented Information Systems 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).

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 of the department of Human-Oriented Information Systems Engineering, 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), the Information Processing Society of Japan (IPSJ), and the Human Interface Society (HIS).