Abstract—The Higher-Order Singular Value Decomposition (HOSVD) is a possible generalization of the Singular Value Decomposition (SVD) to tensors, which have been successfully applied in various domains. Unfortunately, this decomposition is computationally demanding. Indeed, the HOSVD of a $N^{th}$-order tensor involves the computation of the SVD of $N$ matrices. Previous works have shown that it is possible to reduce the complexity of HOSVD for third-order structured tensors. These methods exploit the columns redundancy, which is present in the mode of structured tensors, especially in Hankel tensors. In this paper, we propose to extend these results to fourth order Hankel tensors. We propose two ways to extend Hankel structure to fourth order tensors. For these two types of tensors, a method to build a reordered mode is proposed, which highlights the column redundancy and we derive a fast algorithm to compute their HOSVD. Finally we show the benefit of our algorithms in terms of complexity.

I. INTRODUCTION

An increasing number of signal processing applications deal with multidimensional data like polarimetric STAP [1], multidimensional harmonic retrieval [2] or MIMO coding [3]. The multilinear algebra [4], [5] provides a good framework to exploit these data [6], [2] by conserving the multidimensional structure of the information. Nevertheless, generalizing matrix-based algorithms to the multilinear algebra framework is not a trivial task. In particular, there is no multilinear extension of the Singular Value Decomposition (SVD), having exactly the same properties as the SVD. However, two main decompositions exist: CANDECOMP/PARAFAC (CP) [7], which conserves the rank properties of SVD and the identifiability properties, and the Higher Order Singular Value Decomposition (HOSVD) [5], which keeps the orthogonality properties. HOSVD has been successfully applied in many fields such as image processing [8], sonar and seismo-acoustic [9], ESPRIT [2], ICA [10] and video compression [11].

In [5], it has been shown that the HOSVD of a $N^{th}$-order tensor involves the computation of the SVD of $N$ matrix unfoldings. As a consequence, the computational cost of this algorithm is very high. In [12], it has been shown that the complexity of this algorithm can be reduced for third order structured tensors. These methods exploit the columns redundancy, which is present in the mode of structured tensors, including Hankel tensors. In particular, they involve to determine which columns are redundant and how many times they appear by building a reordered mode.

Hankel tensors occur naturally in signal processing applications, such as the harmonic retrieval problem [13], [14] or in the context of geoscience [15]. In this paper, we propose to extend these results to fourth order Hankel tensors. We propose two ways to extend Hankel structure to fourth order tensors. For these two types of tensors, a method to build a reordered mode is proposed, which highlights the column redundancy and we derive a fast algorithm to compute their HOSVD. Finally we show the benefit of our algorithms in terms of complexity.

The following convention is adopted: scalars are denoted as italic letters, vectors as lower-case bold-face letters, matrices as bold-face capitals, and tensors are written as bold-face italic letters, vectors as lower-case bold-face letters, matrices as bold-face capitals, and tensors are written as bold-face italic letters. We use the superscripts $^H$, for Hermitian transposition and $^*$, for complex conjugation. A permutation of four elements is denoted $\pi = (\pi_1, \pi_2, \pi_3, \pi_4)$, the inverse permutation is denoted $\pi^{-1}$.

II. PRELIMINARIES IN MULTILINEAR ALGEBRA

Let us consider a fourth-order tensor $\mathcal{A} \in \mathbb{C}^{I_1 \times I_2 \times I_3 \times I_4}$ and $\mathcal{A}_{i_1,i_2,i_3,i_4}$ its elements. The 4-mode tensor slices of $\mathcal{A}$ (obtained by fixing one index) are denoted $\mathcal{A}_{\cdots,i_4} \in \mathbb{C}^{I_1 \times I_2 \times I_3}$ (in this case the fourth index is fixed to $i_4$). We recall that a fourth-order tensor can be written as the concatenation of the third-order tensors $\mathcal{A}_{\cdots,i_4}$, for $i_4 \in \{1 \ldots I_4\}$ (see figure 1).

A. HOSVD

One of the extension of the SVD to the tensor case is given by the HOSVD [5]. A tensor $\mathcal{A}$ can be decomposed as follows:

$$\mathcal{A} = \mathcal{K} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)} \times_4 U^{(4)},$$

(1)
where \( \forall n, \mathbf{U}^{(n)} \in \mathbb{C}^{I_1 \times I_2 \times I_3 \times I_4} \) is an unitary matrix and where \( \mathbf{K} \in \mathbb{C}^{I_1 \times I_2 \times I_3 \times I_4} \) is the core tensor, which satisfies the all-orthogonality conditions [5]. The matrix \( \mathbf{U}^{(n)} \) is given by the left singular matrix of the \( n^{th} \)-dimension matrix unfolding (also called n-mode), \( [\mathbf{A}]_n \).

Thus, the calculation of the HOSVD of a fourth-order tensor requires the computation of the left factor in the full SVD of each \( [\mathbf{A}]_n \). However, in many applications, a truncated HOSVD is sufficient, which means that we compute only the \( r_n \) first columns of the matrix \( \mathbf{U}^{(n)} \) (where \( r_n = \text{rank}( [\mathbf{A}]_n ) \) is the \( n \) rank of \( \mathbf{A} \)). The computational cost of the full and rank-truncated HOSVD is summarized in table I.

### Table I

HOSVD for Unstructured Tensors, \( I = \frac{1}{4}(I_1 + I_2 + I_3 + I_4) \).

<table>
<thead>
<tr>
<th>Operation</th>
<th>Cost per Iteration</th>
<th>Full HOSVD</th>
<th>Truncated HOSVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVD of ([\mathbf{A}]_1)</td>
<td></td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>SVD of ([\mathbf{A}]_2)</td>
<td></td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>SVD of ([\mathbf{A}]_3)</td>
<td></td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>SVD of ([\mathbf{A}]_4)</td>
<td></td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>Global cost</td>
<td></td>
<td>16</td>
<td>16</td>
</tr>
</tbody>
</table>

### Example 2.2 (Harmonic retrieval: multi-channel case)

We consider the same case as example 2.1 with several channels \( x_n^{(q)}, q \in \{1 \ldots Q\} \)

\[
x_n^{(q)} = \sum_{p=1}^{P} a_p^{(q)} e^{j\phi_p^{(q)}} e^{(-\alpha_p + j\omega_p)t_n}
\]

Each channel can be folded as third-order tensor [14]

\[
\mathbf{X}_{1+1,i_2+i_3+1}^{(q)} = x_n^{(q)}
\]

under the constraint that \( I_1 + I_2 + I_3 + = N + 2 \). Note that \( \mathbf{X}^{(q)} \) is a Hankel tensor for all \( q \). Finally the tensor \( \mathbf{Y} \in \mathbb{C}^{I_1 \times I_2 \times I_3 \times Q} \), which contains the contribution of all channels is built concatenating the \( Q \) tensors \( \mathbf{X}^{(q)} \):

\[
\mathbf{Y}_{:,i_2,i_3,q} = \mathbf{X}^{(q)}
\]

This tensor has a block-Hankel structure. Its HOSVD is needed in order to estimate the pulsations \( \omega_p \).

### III. reordered tensor unfolding

#### A. Oblique submatrices of a tensor

In order to derive fast algorithms, it is necessary to define a reordered mode which exploits the structure of the tensors. In this way, we propose to extend the notion of oblique submatrices introduced in [17] to fourth-order tensors.

**Definition 3.1 (Oblique submatrices of a tensor):** For any permutation \( \pi \) the oblique submatrices of a tensor \( \mathbf{A} \) are defined as follows. For all \( k \in \{0, \ldots, I_\pi + I_\pi - 2\} \) let

\[
J(\pi)(k) = \min(I_{\pi_1}, I_{\pi_2}, I_{\pi_3}, 1 + k, I_{\pi_2} + I_{\pi_3} - 1 - k).
\]

For all \( i_{\pi_4} \in \{1 \ldots I_{\pi_4}\} \), the coefficients of the \( I_{\pi_1} \times J(\pi)(k) \) oblique submatrix of \( \mathbf{A} \) are

\[
\mathbf{R}_{k,i_{\pi_4}}^{(\pi)}(i,j) = \mathbf{A}_{\pi-1(i,\max(k-I_{\pi_3},1) + j - 1, \min(k,I_{\pi_3}) - j + 1,i_{\pi_4})}
\]

An example of these matrices is shown on figure 2.

This definition is linked to the type-2 submatrices of a 3rd-order tensor introduced in [17]: for each \( i_{\pi_4} \) the matrices \( \mathbf{R}_{k,i_{\pi_4}}^{(\pi)} \) are the oblique submatrices of the 3rd-order tensor

\[
\mathbf{A}_{\pi-1(\pi_1,i_{\pi_4})}^{\pi_2,i_{\pi_3}} \in \mathbb{C}^{I_1 \times I_2 \times I_3}.
\]

**Proposition 3.1:** If \( \mathbf{A} \) is a Hankel or a block Hankel tensor, then all columns of each oblique submatrix \( \mathbf{R}_{k,i_{\pi_4}}^{(\pi)} \) are the same.

**Proof:** Hence \( \mathbf{A} \) is Hankel (or block Hankel), \( \mathbf{A}_{\pi_1,i_{\pi_2},i_{\pi_3}}^{\pi_2,i_{\pi_3}} \) of the form \( \mathbf{A}_{i_1,i_2,i_3} = a_{i_1+i_2+i_3} \) (or \( \mathbf{A}_{i_1,i_2,i_3} = a_{i_1+i_2+i_3} \)). Then \( \mathbf{R}_{k,i_{\pi_4}}^{(\pi)}(i,j) = a_{i_1+\max(k-I_{\pi_3},1)+\min(k,I_{\pi_3})+i_{\pi_4}} \) (or \( \mathbf{R}_{k,i_{\pi_4}}^{(\pi)}(i,j) = a_{i_1+\max(k-I_{\pi_3},1)+\min(k,I_{\pi_3})+i_{\pi_4}} \) which does not depend on \( j \).

#### B. reordered tensor unfolding

Using the previous definition of oblique submatrices, it is then possible to define a reordered tensor unfolding, which highlights the columns redundancy of Hankel (and block Hankel) tensors.

**Definition 3.2 (Reordered tensor unfolding):** For a tensor \( \mathbf{A} \), the reordered tensor unfolding along the \( \pi_1 \) dimension,
denoted \([A]_{\pi}^{n}\) is defined by concatenating all the oblique submatrices for a given permutation \(\pi\). In other words, the matrix \([A]_{\pi}^{n}\) contains all matrices \(R_{k,i}^{(n)}\) for all \(k \in \{0, \ldots, I_{\pi_{2}} + I_{\pi_{3}} - 2\}\), \(i_{\pi} \in \{1 \ldots I_{\pi_{1}}\}\).  

For each value of \(i_{\pi}\), the set of \(R_{k,i}^{(n)}\) (for all \(k \in \{0, \ldots, I_{\pi_{2}} + I_{\pi_{3}} - 2\}\)) contains all the columns of \([A]_{\pi}^{n}\) (see [17] for details). Then it is clear that \([A]_{\pi}^{n}\) contains the same columns as \([A]_{\pi}^{\pi}\). Therefore the left singular factor in the SVD of \([A]_{\pi}^{n}\) is the same as in the SVD of \([A]_{\pi}^{\pi}\).

Example 3.1: Consider the \(2 \times 2 \times 2 \times 2\) Hankel tensor given by \(A_{1,2,3,4}\). The classic 1-mode is given by:

\[
[\mathbf{A}]_{1} = [\mathbf{A}_{1,\ldots,1}]_{1} [\mathbf{A}_{1,\ldots,2}]_{1}
\]

with

\[
[\mathbf{A}_{1,\ldots,1}]_{1} = \begin{bmatrix}
  4 & 5 & 6 & 5 & 6 & 7 & 6 & 7 & 8 \\
  5 & 6 & 7 & 6 & 7 & 8 & 7 & 8 & 9
\end{bmatrix}
\]

and

\[
[\mathbf{A}_{1,\ldots,2}]_{1} = \begin{bmatrix}
  5 & 6 & 7 & 6 & 7 & 8 & 7 & 8 & 9 \\
  6 & 7 & 8 & 7 & 8 & 9 & 8 & 9 & 10
\end{bmatrix}.
\]

The reordered 1-mode unfolding is equal to:

\[
[\mathbf{A}]_{1} = \begin{bmatrix}
  4 & 5 & 6 & 5 & 6 & 7 & 6 & 7 & 8 \\
  5 & 6 & 7 & 6 & 7 & 8 & 7 & 8 & 9 \\
  6 & 6 & 7 & 7 & 7 & 7 & 8 & 8 & 8 \\
  7 & 7 & 7 & 7 & 8 & 8 & 8 & 9 & 9 \\
  8 & 8 & 8 & 8 & 9 & 9 & 9 & 10 & 10
\end{bmatrix}
\]

IV. FAST MULTILINEAR SVD FOR 4TH ORDER HANKEL TENSORS

A. Algorithms exploiting column-redundancy

The fast multilinear SVD relies on algorithms exploiting column-redundancy as it was shown in [17]. Let us consider the unfolding of a tensor \(\mathbf{A} \in \mathbb{C}^{I_{1} \times I_{2} \times I_{3} \times I_{4}}\) in the \(n^{th}\) dimension, \([\mathbf{A}]_{n} \in \mathbb{C}^{I_{n} \times \prod_{k \neq n} I_{k}}\). We assume \([\mathbf{A}]_{n}\) contains column-redundancy.

We define the \(I_{n} \times J_{n}\) matrix \(\mathbf{H}_{n}\) as the matrix obtained by removing the repeated columns in the \(n^{th}\) mode \((J_{n} \leq \prod_{k \neq n} I_{k})\), and we denote \(d_{\pi}^{(n)}\) the number of occurrences of the \(k^{th}\) column of \(\mathbf{H}_{n}\) in the \(n^{th}\) mode. It is clear that:

\[
[\mathbf{A}]_{n} [\mathbf{A}]_{n}^{H} = \mathbf{H}_{n} \mathbf{D}_{n}^{2} \mathbf{H}_{n}^{H}
\]

where \(\mathbf{D}_{n} = \text{diag}(\sqrt{d_{\pi}^{(n)}}, \ldots, \sqrt{d_{\pi}^{(n)}})\). This equality allows to prove that \(\mathbf{H}_{n} \mathbf{D}_{n}\) and \([\mathbf{A}]_{n}\) have the same left factors. Thanks to the smaller dimensions of \(\mathbf{H}_{n} \mathbf{D}_{n}\) it is then possible to derive fast multilinear SVD algorithms for structured tensors. To take advantage of this method, a method need to be provided to compute \(\mathbf{H}_{n}\) and \(\mathbf{D}_{n}\).

B. Hankel tensor

For all \(i_{\pi} \in \{1 \ldots I_{\pi}\}\), \(k \in \{0, \ldots, I_{\pi_{2}} + I_{\pi_{3}} - 2\}\) and for all permutation \(\pi\), the proposition 3.1 shows that all columns of \(\mathbf{R}_{k,i}^{(\pi)}\) are equal. Using this property, it is possible to derive the nonredundant matrix \(\mathbf{H}_{\pi}\) and the weighting factors \(d_{k}^{(\pi)}\) for the unfolding tensor \([\mathbf{A}]_{\pi}\).

First, it is possible to calculate the column redundancy for each matrix \([\mathbf{A}]_{\pi}^{n}\) (which is the 1-mode of the 4-mode tensor slice \(\mathbf{A}_{\pi}^{n}\)).

- The nonredundant matrix, denoted \(\mathbf{G}_{\pi}^{(n)} \in \mathbb{C}^{I_{\pi_{2}} \times J^{(n)}}\) with \(J^{(n)} = I_{\pi_{2}} + I_{\pi_{3}} - 1\) is obtained by concatenating the first column of each matrix \(\mathbf{R}_{k,i}^{(\pi)}\) for all \(k \in \{0, \ldots, I_{\pi_{2}} + I_{\pi_{3}} - 2\}\) and for \(i_{\pi}\) set.
- The weighting factors \(g_{k,i_{\pi}}^{(\pi)}\) are the number of columns of the matrices \(\mathbf{G}_{\pi}^{(n)}\), which is equal to \(J^{(n)}(k)\) for each \(k \in \{0, \ldots, I_{\pi_{2}} + I_{\pi_{3}} - 2\}\). It can be rewritten as follows (see also figure 3):

\[
g_{k,i_{\pi}}^{(\pi)} = \begin{cases}
  1 + k & \text{if } 1 \leq 1 + k < \min(I_{\pi_{2}}, I_{\pi_{3}}) \\
  \min(I_{\pi_{2}}, I_{\pi_{3}}) & \text{if } \min(I_{\pi_{2}}, I_{\pi_{3}}) \leq 1 + k \leq \max(I_{\pi_{2}}, I_{\pi_{3}}) \\
  J^{(n)} - k & \text{if } \max(I_{\pi_{2}}, I_{\pi_{3}}) \leq 1 + k \leq J^{(n)}
\end{cases}
\]

Hence, since \(J^{(n)}\) does not depend on \(i_{\pi}\), the coefficient \(g_{k,i_{\pi}}^{(\pi)} = g_{k}^{(\pi)}\) are equal for all values of \(i_{\pi}\). These remarks are true for both Hankel and block Hankel tensors. However, the rest of the derivation will differ.

1) Block Hankel tensors: In this case, the matrices \(\mathbf{G}_{\pi}^{(n)}\) are different for each values of \(i_{\pi}\) (especially they have no columns in common) since \(\mathbf{G}_{\pi}^{(n)}(i, k + 1) = \delta(i_{\pi}, k)\). Then
the matrix $H_{\pi_1}$ and the weighting factors $d_k^{(\pi)}$ are obtained by concatenating the matrices $G_{i\pi_4}^{(\pi)}$ and the weighting factors $g_{k,i\pi_4}$ for all $i\pi_4 \in \{1 \ldots I_{\pi_4}\}$. Finally the dimensions of the matrix $H_{\pi_1}$ are $I_{\pi_1} \times I_{\pi_4} J^{(\pi)}$.

2) Hankel tensors: In this case, the matrices $G_{i\pi_4}^{(\pi)}$ have some columns in common. First, the number of different columns in $[A]_{\pi_1}$ is equal to $L^{(\pi)} = I_{\pi_2} + I_{\pi_3} + I_{\pi_4} - 2$ since the values of $i\pi_2 + i\pi_3 + i\pi_4$ are between 3 and $I_{\pi_2} + I_{\pi_3} + I_{\pi_4}$. The dimensions of $H_{\pi_4}$ are then equal to $I_{\pi_1} \times I_{\pi_4} J^{(\pi)}$.

Then the redundancy between the matrices $G_{i\pi_4}^{(\pi)}$ can be calculated, using $G_{i\pi_4}^{(\pi)}(i,k) = a_k+i\pi_4+1 = G_{i\pi_4}(i,k+1)$. In other words, it means the matrices $G_{i\pi_4+1}$ and $G_{i\pi_4}$ can be written as follows:

$$G_{i\pi_4} = [g_1, \ldots, g_{J^{(\pi)}}] \quad (15)$$

Thanks to this remark, it is possible to build the non redundant matrix $H_{\pi_1}$ for the mode $[A]_{\pi_1}$ by concatenating the matrix $G_{1}^{(\pi)}$ and the last columns of the matrices $G_{i\pi_4}^{(\pi)}$ for $i\pi_4 \in \{2 \ldots I_{\pi_4}\}$

$$H_{\pi_1}(i,l) = \begin{cases} G_{i}^{(\pi)}(i,l) & \text{if } l \leq J^{(\pi)} \\ G_{i-J^{(\pi)}+1}^{(\pi)}(i,l) & \text{if } J^{(\pi)} < l \leq I_{\pi_4} \end{cases} \quad (17)$$

We can also determine in how many matrices $G_{i\pi_4}$ appears each column of $H_{\pi_1}$:

$$r_1 = \begin{cases} 1 & \text{if } l \leq \min(I_{\pi_4}, J^{(\pi)}) \\ \min(I_{\pi_4}, J^{(\pi)}) & \text{if } \min(I_{\pi_4}, J^{(\pi)}) \leq l \leq L^{(\pi)} - \min(I_{\pi_4}, J^{(\pi)}) \\ L^{(\pi)} - l & \text{else} \end{cases} \quad (18)$$

Finally the factors $d_k^{(\pi)}$ are obtained by combining equations (18) and (14):

$$d_k^{(\pi)} = \sum_{j=\min(I_{\pi_4}, J^{(\pi)})}^{\min(I_{\pi_4}, J^{(\pi)})-r_1} g_j^{(\pi)} \quad (19)$$

3) Comparison of the complexities: Thanks to the previous results, the computation of the HOSVD of a Hankel or block Hankel relies on the SVD of $H_2D_1$, $H_3D_2$, $H_4D_3$ and $H_2D_1$. The costs of these fast algorithms are summarized in table II. We notice that this method allows a reduction of the complexity of one order of magnitude for Hankel block tensors and two orders of magnitude for Hankel tensors compared to the classic algorithm presented in table I.

### Table II

**Fast HOSVD for Hankel and Block Hankel Tensors.**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Cost per iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hankel</td>
</tr>
<tr>
<td>SVD of $[A]_{\pi_1}$</td>
<td>$4\pi_1 I_{\pi_2} + I_{\pi_3} + I_{\pi_4}$</td>
</tr>
<tr>
<td>SVD of $[A]_{\pi_2}$</td>
<td>$I_{\pi_1} I_{\pi_3} I_{\pi_4}$</td>
</tr>
<tr>
<td>SVD of $[A]_{\pi_3}$</td>
<td>$I_{\pi_1} I_{\pi_2} I_{\pi_4}$</td>
</tr>
<tr>
<td>SVD of $[A]_{\pi_4}$</td>
<td>$I_{\pi_1} I_{\pi_2} I_{\pi_3}$</td>
</tr>
<tr>
<td>Global cost for cube tensor</td>
<td>$48RT^2$</td>
</tr>
</tbody>
</table>

V. Conclusion

In this paper, we extended the results on third-order Hankel tensors to fourth order Hankel and block Hankel tensors. The reordered tensor unfolding, which highlights the column redundancy, has been detailed for these two types of tensors. Then a fast algorithm to compute their HOSVD has been proposed. Finally the benefit in terms of complexity of our algorithms has been evaluated: the reduction of the complexity is one order of magnitude for Hankel block tensors and two orders of magnitude for Hankel tensors. This results shows the interest of our approach. It could also be interesting to consider other fourth-order structured tensors like Hermitian or Toeplitz tensors.

### References


