Adaptive lossy compression of climate model data based on hierarchical tensor with Adaptive-HGFDR

Zhaoyuan Yu[1,2], Dongshuang Li[3,4], Wen Luo[1,2], Zhengfang Zhang[1], Yuan Liu[1], Uzair Aslam Bhatti[1], Linwang Yuan[1,2,*]

[1]Key Laboratory of Virtual Geographic Environment, Ministry of Education, Nanjing Normal University, Nanjing, China,
[2]Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, Nanjing, China,
[3]Jiangsu Key Laboratory of Crop Genetics and Physiology/Jiangsu Key Laboratory of Crop Cultivation and Physiology, Agricultural College of Yangzhou University, Yangzhou, China,
[4]Jiangsu Co Innovation Center for Modern Production Technology of Grain Crops, Yangzhou University, Yangzhou, China

Correspondence to: Linwang Yuan (email: yuanlinwang@njnu.edu.cn)

Abstract. Lossy compression has been applied to large-scale experimental model data compression due to its advantages of a high compression ratio. However, few methods consider the uniform distribution of compression errors. Here we develop an adaptive lossy compression method with uniform distribution of compression error for earth system model data based on Blocked Hierarchical Geospatial Field Data Representation (Blocked-HGFDR). The original Blocked-HGFDR method is improved from the following perspectives. Firstly, the original data are divided into a series of data blocks with more balanced size. After that, from the mathematical model of the Blocked-HGFDR, the relationship between the compression parameter and compression error in Blocked-HGFDR for each data block is conducted. Finally, our method, the Adaptive-HGFDR, achieves the data compression with the uniform distribution of compression error through designing an optimal compression parameter control mechanism and a fast search method. Experiments concerning model data compression are carried out based on the Community Earth System Model (CESM) data. The results show that our method has higher compression ratio and more uniform error distributions, compared with other commonly used lossy compression methods, and can be well applied to multiple climate model variables.

1 Introduction

Earth System Model Data (ESMD), which comprehensively characterize the Earth system over space-time dimensions, are presented as multidimensional arrays of floating-point numbers (Kuhn et al., 2016; Wulder et al., 2012). With the rapid development of earth system models and growing multi-scenario earth system simulation experiments, ESMD has shown an exponential increase in data volume and data complexity (Anon, 2011; Of and Acm, 2000). Lossless compression methods, which could reduce the data volume without affecting the data quality, are commonly used in compressing ESMD data. However, their compression ratios grow much slower than the increment of data volume (Kumar et al., 2008; Tao et al., 2017c; Baker et al., 2016). Thus, the compression performance of the lossless compression will be relatively reduced. Lossy

compression methods, which focus more on saving space over preserving the data accuracy, are achieved more and more attention in ESMD compression. However, the reduction of data accuracy may lead to the deviation in calculating the long-term trend, change rate, inflection point and abrupt change, and further affect the data quality as well as the subsequent analysis of ESMD, especially for the variables with strong spatial heterogeneity like temperature or fluxes (Zabala and Pons, 2011; Feng et al., 2014; Baker et al., 2014; Berres et al., 2017). The heterogeneity of original data distribution could lead to the uneven distribution of compression errors, and make it difficult to separate the compression error from the spatial-temporal characteristics of original ESMD. Therefore, keeping the distribution of compression error as uniform as possible can improve the data quality of ESMD lossy compression.

The main idea of ESMD lossy compressions is to eliminate unnecessary or redundant information in data to reduce the data size. There are two different kinds of information that can be considered as unnecessary or redundant in ESMD: information of data descriptions and information of data features. Therefore, there are two major different types of ESMD lossy compression methods: the description-based lossy compression and feature-based lossy compression. The description-based lossy compression attempts to transform or simplify data descriptions to reduce data size. One subcategory of the description-based lossy compression is the file-based compression method, which compresses ESMD files using general file compression methods, such as GRIB2 and NetCDF (Bing et al., 2014; Hübbe et al., 2013). Another subcategory is the error truncation-based compression, which assumes the high-precision float point expression in ESMD is not necessary. Typical error truncation-based compression, such as FPZIP (Lindstrom and Isenburg, 2006) and APAX (Hübbe et al., 2013), implements the data compression through controlling the precision of a floating-point expression of the original data, and eliminating redundant floating-point precision. Recently, statistical or sparse data coding are introduced to produce better compact data descriptions (Papaioannou et al., 2011; Tao et al., 2017a; Akbudak et al., 2017). For example, vector quantization (VQ) (Vector Quantization) (Guinness and Hammerling, 2016) organize the data as a simple one-dimensional array and use a dictionary (or code table) to compose the common patterns of the data. Although the description-based lossy compression is typically easy to implement, it is not flexible enough to control the distribution of compression error. For the file-based compression method, it is difficult to arbitrarily adjust the compression parameter according to the given compression error. For the error truncation-based compression, the distribution of floating-point precision of ESMD is not uniform, which could lead to the unevenly distribution of compression errors. For VQ method, the use of a dictionary or code table enlarges the complexity of the data structure, which requires the additional operations to control the compression error. For example, when the data volume is large, the change of compression error control conditions could lead to significant time cost due to frequent compression and decompression of data (Anon, 2013; Liu et al., 2014; Mummadisetty, 2015). To summarize, it is hard to achieve flexible control of the compression ratio and errors for the description-based lossy compression methods.

As for the feature-based lossy compression, it assumes that the original data are a mixture of various features, and some features of the data can be eliminated. Based on how to deal with the feature of the original data, the feature-based lossy compression can be further classified into two subcategories: feature filtering-based compression and feature prediction-based compression. The feature filtering-based compression defines certain rules to extract and filter features from the original data and remove

unnecessary features to reduce the data volume. One example is the image-based method, which slices ESMD from different dimensions as separate images, and each image is then compressed by feature filtering with wavelet transformation or Discrete Fourier Transform (Taubman and Marcellin, 2002). As the compression is applied to single image slice, the correlations between different image slices are not always well utilized during different compression processes. Therefore, the compression error control between different image slices is not uniform (Castruccio and Genton, 2016; Guinness and Hammerling, 2016). For the feature prediction-based method, it use parametric functions to fit the data and predict the structure of data (Adhianto et al., 2010; Cui et al., 2007). Then the function parameters are used to represent the original data in a compact form, reducing the data volume. NUMARCK (Zheng et al., 2017), SSEM (Wilczyñski, 2001), SPECK (Wang and Li, 2006), and ZFP (Diffenderfer et al., 2019b) are typical methods that use the feature prediction to achieve lossy compression. It is worth mentioning that the feature prediction-based compression method is highly dependent on the parametric function and the prediction model. If the parametric function and the distribution of original data does not fit well, the performance of the compression may poor. Furthermore, the parametric function and prediction model are typically data-adaptive, which means it is difficult to adjust the compression parameters according to different constraints. To summarize, the performance of feature-based lossy compression is heavily depended on the rules of feature extraction/filtering or feature prediction. However, we have rare knowledge on how to define the rules of feature extraction/filtering or feature prediction to make the distribution of compression error uniform, which is important for ESMD (Carmel, Y.,2004; Lyre, H. (2004). Linton, O.,2007).

Both description-based and feature-based lossy compression methods are mostly inherited from low dimensional data compression methods (e.g. one-dimensional vector or two-dimensional images). None of these methods considers ESMD as the high dimensional data with the heterogeneous correlation between different dimensions. For some ESMD variables like temperature, solar and longwave flux, there are significant correlations between different dimensions, i.e., values in neighboring ranges tend to be numerically close to each other. As the dimension of ESMD is commonly high (e.g. even an ESMD with only one attribute and three spatial dimensions forms a four-dimensional data), the ignorance of the multidimensional correlation structure results in low compression performance in terms of efficiency (Diffenderfer et al., 2019a; Schoellhammer et al., 2004). Ignoring the control of such high dimensional correlation structure also makes it difficult to keep the error distribution uniform in different dimensions. When there are the correlation structures in higher dimensions, the uneven distribution of compression error enlarged and the impacts on the data quality becomes larger. So, it is urgent to develope high dimensional lossy compression method to avoid uneven distribution in different dimensions (Tao et al., 2017b).

Tensors are well used to represent the multidimensional data. The corresponding tensor decomposition method eliminates inconsistent, uncertain, and noisy data without destroying the intrinsic high dimensional correlation structures. With the small sets of decomposed features from tensor decomposition, tensor reconstruction can approximate the original data as accurate as possible (Li et al., 2018). Therefore, the tensor decomposition-based compression methods, with the only parameter "*rank*" controlling the compression ratio and relative compression errors, has been gradually introduced into data compression in recent years (Yuan et al., 2015). Among exiting tensor-based compression methods, the hierarchical tensor approximation,

could achieve higher quality at large compression ratios than traditional tensor methods through extracting data features level by level (Linton and Xiao, 2001; Lyre, 2004). Yuan et al (2015) designed an improved hierarchical tensor method Hierarchical Geospatial Field Data Representation (Blocked-HGFDR) to compress geospatial data in a hierarchical tree structure, showing obvious advantages in compression ratio and error distribution than traditional methods like NetCDF-based data compression. Nevertheless, Blocked-HGFDR only pays attention to global average error when assessing the compression quality. However, it is possible the global average error is relatively small when the local error is large. Therefore, we need to work out how to control the uniform distribution of local compression error of the Blocked-HGFDR method. In addition, Blocked-HGFDR is only evaluated with eight climate variables, so research on the adaptivity and universality of Blocked-HGFDR to more variables of ESMD is also required.

In this paper, we extend the study of Blocked-HGFDR by discussing the factors and constraints that affect the application of Blocked-HGFDR in ESMD compression. We study the empirical quantitative relationship between compression errors and compression parameters, and develop an Adaptive-HGFDR method based on an adaptive data block and rank adjustment mechanism under accuracy constraints. Experiments on climate simulation dataset with 22 variables are carried out to evaluate the performance and applicability of the methods in ESMD compression.

The remainder of this paper is organized as follows. Section 2 introduces the basic ideas about developing Adaptive-HGFDR. Section 3 discusses the block mechanism, the relationship between the compression parameter and compression error, and the fast search algorithm. Section 4 uses temperature data to verify that the method can obtain adaptive rank under accuracy constraint. Section 5 discusses the effectiveness and computational efficiency of the method, as well as the results.

2 Basic idea

To adaptively control the compression error of Blocked-HGFDR, there are mainly three issues to consider. The first issue is the dimensional imbalance of ESMD data. For example, the data in the temporal dimension is typically longer than that in the spatial dimension for a spatio-temporal series with long observations. This dimensional imbalance not only enlarges the overall fitting error during the data fitting process for tensor decompression, but also makes it difficult to achieve fine control of the compression ratio and error distribution. Therefore, it is better to split the original data into small local data blocks, achieving a more balanced dimension structure in each local data block. Then processing each local data block will make accurate approximation and fine control of the compression error much easier. The second issue is the heterogeneity characteristics of different variables or data blocks of ESMD. As the data distribution is different, the compression parameter of different variables or data blocks should also be different. Therefore, conducting the relations between the compression error distribution and the compression parameters with the consideration of data heterogeneity is in a high propriety. The third issue is that the compression should meet different degrees of accuracy. Therefore, a feedback control mechanism which could modify the compression parameter according to the accuracy demands should be developed.

Based on the above considerations, Our methods, Adaptive- HGFDR, is developed according to the following three procedures (Figure 1). Procedure 1: Splitting the original ESMD into small data blocks. In this procedure, the dimension to split the data and the optimized size of the data block is determined by conducting different combinations of data blocking in terms of the dimension and block counts. Procedure 2: Conducting the relationship between compression errors and compression parameters. In order to obtain a uniform distribution of the compression error for each data block, an empirical relationship between the compression error and the rank value is established, where the rank value for each data block can be adjusted at any given compression error. Procedure 3: Adaptive searching for the optimized compression parameters. A binary search method is used to search the optimal compression parameter, which is updated with a feedback control mechanism until the compression error meet the accuracy.

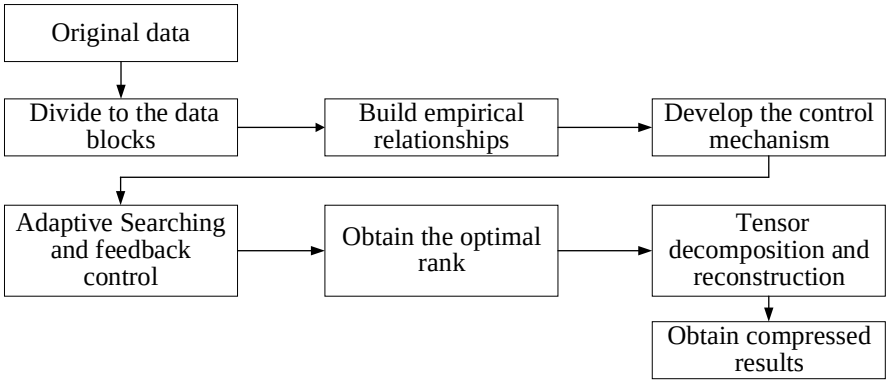

Figure 1: . Overall framework of the basic idea..

3 Method

3.1 Block hierarchical tensor compression

EMSD is a multidimensional array. It can be seen as a tensor with the spatio-temporal references and the associated attributes. Without loss of generality, a three-dimensional tensor can be defined as $Z \in \mathbb{R}^{I \times J \times K}$ (None, 1970; Suiker and Chang, 2000), where I, J, and K are values that represent the number of grids along the dimensions of longitude, latitude, and time (or height), respectively. These dimensions are always unbalanced due to the different spatial and temporal resolution. So the data block is introduced to reduce the impact of dimension unbalance on the data compression.

Definition 1 Data block. For the spatio-temporal data $Z \in \mathbb{R}^{I \times J \times K}$, it can be seen that it is composed of a series of local data with the same spatio-temporal reference. Here, each local data is defined as the data block as follow:

$$part(Z, n) = \{C_1, C_2, \cdots, C_n\} \quad (1)$$

Here, *part*( ) is the function that partitions the original tensor $\mathcal{Z}$ to a series of data block $\{C_i\}_{i=1}^{m}$, and each data block C_i includes local spatial and temporal information, and *n* is the number of data blocks. Compared with the original data, the dimensions of these data blocks are smaller that tend to be more balanced. For the divided data blocks, in order to construct a data compression with uniform error distribution, the key problem to be solved is how to compress each block data so that the error of each block data is close.

Definition 2 Blocked-HGFDR. Based on the divided data blocks, Yuan (Yuan et al., 2015) proposed the Blocked-HGFDR method based on the hierarchical tensor compression. In this method, the hierarchical tensor compression is applied to each block, then the hierarchical tensor compression of each data block is obtained by selecting the dominant feature component and filtering out the residual structure. This method utilizes the hierarchical structure of data features, greatly reducing data redundancy, and thereby achieving the efficient compression of the amount of spatiotemporal data (Yuan et al., 2015). The overall compression of the Blocked-HGFDR can be formulated as:

$$\begin{cases} H(A) = (U_R \otimes U_{R-1} \otimes \cdots \otimes U_1)\tilde{B}_L \tilde{B}_{L-1} \cdots \tilde{B}_1 B_{12\cdots R} + res \\ \tilde{B}_j = B_{p_{L_j}} \otimes \cdots \otimes B_{p_L} \qquad j = \{1, 2, \ldots, L\} \end{cases} \tag{2}$$

Similar to the prominent components obtained by SVD (Lathauwer et al., 2000; Springer, 2011) for two-dimensional data, the matrix U_R and the sparse transfer tensor B_R are considered to be the r-th component of a third-order tensor in each dimension, respectively, where R denotes the number of multi-domain features. The residual tensor, res, in Eq. (2) denotes information not captured by the decomposition model, and $(U_R \otimes U_{R-1} \otimes \cdots \otimes U_1)\tilde{B}_L \tilde{B}_{L-1} \cdots \tilde{B}_1 B_{12\cdots R}$ in Eq. (2) is the reconstructed r-th core tensor and feature matrix (Matrices, 2006; Oseledets and Tyrtyshnikov, 2009).

3.2 Adaptive parameter selection and solution

Since the feature structure of each divided data block is different (Hackbusch and Kühn, 2009), the key to control the stable distribution of compression error in Blocked-HGFDR is to adaptively select the compression parameter for each local data according to the given compression error. So the key step is to construct controlling mechanism based on the relationship between the compression error and compression parameter. Thus, the following terms are defined.

Definition 3 The controlling mechanism. Lars Grasedyck defines a hierarchal tensor SVD algorithm, and the approximate accuracy is determined by rank (Grasedyck, 2010). In Blocked-HGFDR, the relationship between the compression error and compression parameter (*Rank*) is given as $\varepsilon = \alpha Rank^{-\beta}$, thus the controlling mechanism to determine the compression parameter of each block data should be the rank value closest to the given error as follows:

$$\varepsilon = \alpha Rank^{-\beta} \le \varepsilon_{Given} \tag{3}$$

ε_{Given} is the given compression error that depends on different application scenarios; α, β are the coefficients depended on the structure and complexity of the data, which can be obtained by the simulation experiment for actual data.

In Blocked-HGFDR, the relationship between the compression ratio (φ) and compression parameter (*Rank*) are given as follows:

$$\varphi = \frac{datasize}{aRank^3 + bRank^2 + cRank + d} \tag{4}$$

As shown in Eqs. (2), (3), and (4), in Blocked-HGFDR, with rank decreasing, the data compression ratio of Blocked-HGFDR increases, and the compression error also increases. In Blocked-HGFDR, the rank value of different blocks is fixed, it results in the fluctuation of the compression error in specific dimension. Since the structure of each block is different, to achieve a stable error distribution of compressed data for the given compression error, the key is to select the rank for each block of data separately. We can select the optimum parameter when the minimum $Rank$ is reached to make the compression error close to the given compression error.

Definition 4 The fast search algorithm for the optimal rank.

For Blocked-HGFDR, the rank value on each data block is the main parameter, which represents the number of the feature component in each block data. So the rank value can influence the data fitting performance as well as the compression ratio for each data block. Although we have empirical observation that large rank will lead to worse compression ratio and smaller compression error. The quantitative relations of different variables and data blocks are still unknown. With the empirical relationship between the compression error and the rank value, it is possible to calculate the optimal compression parameters for each data block at any given accuracy constraints to achieve uniform distribution of the compression errors. However, the heterogeneity of real data and the constraints of the compression require dynamic search and update of the optimized compression parameters.

To find the optimal parameter for data block C_i, with the above constructed controlling mechanism, the fast search algorithm based on dichotomy is constructed. That means before adjusting the rank each time, reduce the selection interval by half of the rank, the optimal rank coressponding to the given error is constantly approached in half. The algorithm is implemented as follows:

Algorithm: optimal parameter search algorithm based on dichotomy

---

Input: data block $C_i \in \mathbb{R}^{Q \times W \times E}$; given data error std_err;

Output: the optimum rank R_Opt

Function Description: $EvalErr(C_i, r)$ is used to calculate the error of hierarchal tensor SVD of C_i at rank r based on Eqs. (4) and (6). $Round()$ is the rounding function; $Max()$ is the function which taking the maximum value

---

1: $R\_Max = Max(Q,W,E)$, $R\_Min = 0$

2: $R\_Mid = Round(\frac{R\_Max + R\_Min}{2})$

3: $err = EvalErr(C_i, R\_Mid)$

4: While ($err != std\_err$ && $R\_Max > R\_Min$)

5: If ($err > std\_err$)

6: $R\_Min = R\_Mid + 1$

7: Else

8: $R\_Max = R\_Mid - 1$

9: End If

10: $R\_Mid = Round(\frac{R\_Max + R\_Min}{2})$

11: $err = EvalErr(C_i, R\_Mid)$

12: End While

13: Return ($R\_Opt = R\_Mid$)

During the whole algorithm, the function $EvalErr(C_i, r)$ is the computing intensive function that could be the performance bottleneck. If we consider a calculation of $EvalErr(C_i, r)$ as one meta calculation, the complexity of the traditional traversal method is $O(n)$, when intorducing the dichotomy optimization, the complexity can be reduced to $O(\log n)$ (Rouillier, F. et al. 2004).

4 Case study

4.1 Data description and experimental configuration

In this paper, data produced by Community Earth System Model are used as experimental data to evaluate the compression performance of Adaptive-HGFDR, which can be obtained from Open Science Data Cloud in NetCDF (Network Common Data Form)format(http://doi.org/10.5281/zenodo.3997216). The data set includes air temperature data (T) stored as a $1024 \times 512 \times 26$ (latitude × longitude × height) tensor and 22 other attributes stored as a $1024 \times 512 \times 221$ (latitude × longitude × time) tensor from 1980/01 to 1998/05. When reading the NetCDF data, a total of 48GB memory will be occupied. Research experiments were performed by the MATLAB R2017a environment on a Windows 10 Workstation (HP Compaq Elite 8380 MT) with Intel Corei7-3770 (3.4 GHz) processors and 8 GB of RAM.

The following experiments were performed: 1) simulations with different block counts were carried out to find the optimal block counts ; 2) comparison of the compressed performance between the proposed method and the commonly used compression methods was performed; 3) to show the applicability of the method, the compressions with the multiple variables are performed. Two key indices are used to benchmark the performances: the relative compression error ratio and compression ratio. The relative compression error ratio is calculated as:

$$\varepsilon = \frac{\left\| \mathrm{T}_{\mathrm{Original}} - \mathrm{T}_{\mathrm{Reconstruction}} \right\|^2}{\left\| \mathrm{T}_{\mathrm{Original}} \right\|^2} \quad (4)$$

Here, the $\| \ \|^2$ is the F norm. $\mathrm{T}_{\mathrm{Original}}$ is the original tensor data, $\mathrm{T}_{\mathrm{Reconstruction}}$ is the compressed tensor data.

The compression ratio ϕ is calculated as：

$$\phi = \frac{D_{\mathrm{original}}}{D_{compression}} \quad (5)$$

Here, D_{original} is the memory size of original data before compression, $D_{compression}$ is the memory size of Compressed reconstructed data.

4.2 Optimal block count selection

The block counts denotes the number of block data, whcih will affect the data fitting performance, the compression ratio and the complexity of file system I/O (input/output). More blocks may achieve smaller rank for accurate fit of each block and finer control of the error distribution of overall compression but may lead to more parameters and file system I/O. On the contrary, if the block counts is small, the heterogeneity in one data block may be large, which will result in a low accuracy of data fitting and may require higher rank for the compression. Therefore, selection an optimal block counts to meet the largest compression ratio. Here, we randomly divide original data into a series of data blocks with different block counts. The data with different block counts are then compressed to meet the demand of the given compression error. By comparing the compression ratios, the optimal block count is determined at the largest compression ratio.

We demonstrate the selection of the optimal block count with the temperature data (T). Firstly, the data for each block is better to have the same size. And the block counts with a power of 2 will be best to fit as the near balanced data blocking. Therefore, a series of block counts of 4, 16, 64, and 128, 256 are generated as potential block counts. Secondly, the given compression error may affect the optimal block count, we will setup an initial given compression error of 10^{-4} to conduct the experiments.

The effect of block counts (BC) on the compression ratio is shown in Figure 2(a). Clearly, the highest compression ratio is reached when block counts equals to 16 (BC=16). Hence, the optimum block count is 16, and the corresponding block zize is

$256\times128\times26$. Interesting things can be revealed that the overall compression ratio presents a downward trend with BC in the range 16 and 64. When BC is larger than 64, the data volume of each block becomes smaller, and the number of feature components required to achieve the same compression error significantly decrease, so the data volume of each block after compression significantly decreases. Although the number of blocks is increasing (BC=128 and BC=256), the significant reduction of local block data volume makes the overall compression ratio show an upward trend. Bedsides that, the relationship between the block counts and compression ratio is related to the structure and complexity of the data itself, which is different for the data with different distribution characteristics. For the temperature data (T), the compression ratio reaches a maximum when the block count is set to 16.

Figures 2(b) show the comparison between the original data and the compressed data with different block counts. It can be seen there's no significant difference. This may because that the prominent feature components are gradually added to approximate the original data to affect the compression error, no matter how many blocks are, the proposed method can approach the given compression error by controlling the rank value to provide the accurate compression results.

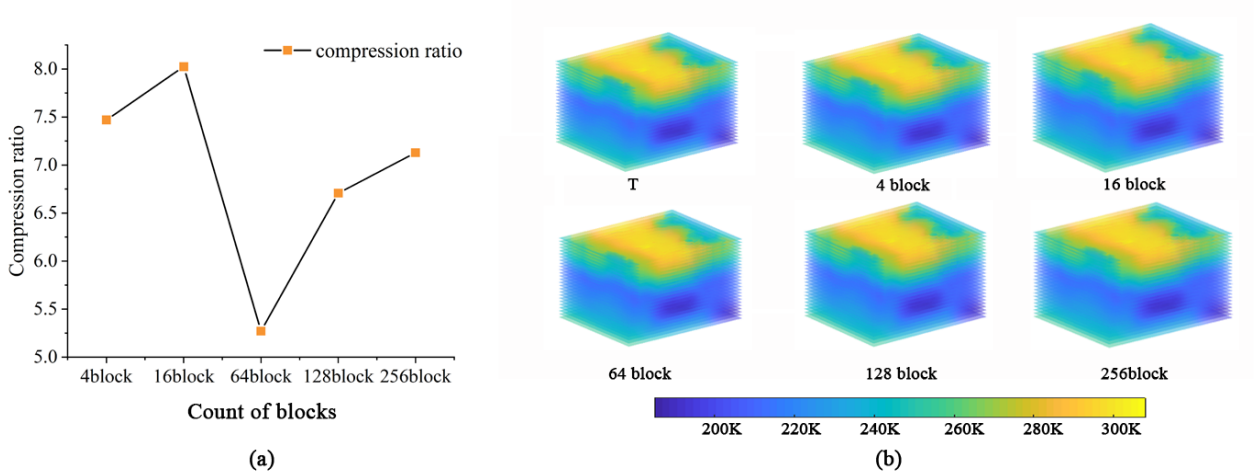

Figure 2. Original data and compression with different block counts: (a) The compression ratio change with different number of blocks; (b) original and reconstruction data of temperature (T) with different blocks

4.3 Comparison with traditional methods

To verify the advantage of the proposed compression method for ESMD, we compare Adaptive-HGFDR with the Blocked-HGFDR and the classical ZFP method. Here, the block counts in the proposed method and Blocked-HGFDR method are both set as 16, and the rank in Blocked-HGFDR is selected as the average of the adaptive rank in each divided block data. The compression error and compression ratio of the zfp algorithm are affected by the tolerance parameter, which is set to 0.5. Without loss of generality, the relative compression error ratios are set as 10^{-5}, 5×10^{-5}, 10^{-4}, 5×10^{-4}, 10^{-3} respectively. The compression ratios of different compression methods are shown in Figure 3.

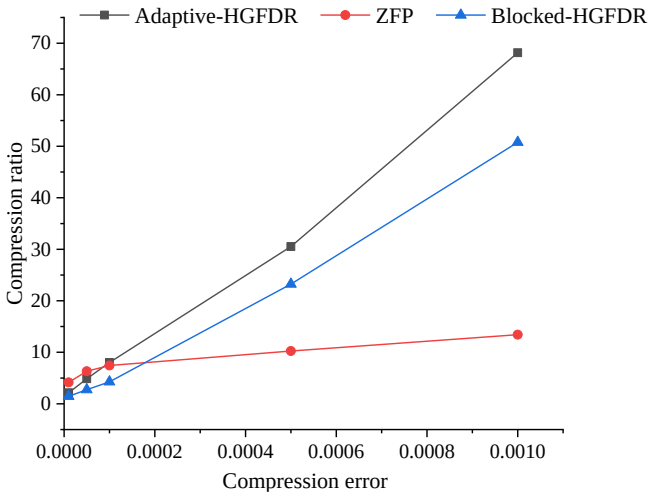

Figure 3. Compression ratio versus compression error for different methods.

Figure 3 shows that as the relative compression error ratio grows, the compression ratio of all three methods becomes larger and larger. However, the growth rate of ZFP is much slower than that of Blocked-HGFDR and Adaptive-HGFDR. When the compression error is less than 0.0001, the compression ratio of ZFP is a little higher than that of Adaptive-HGFDR and Blocked-HGFDR. This may because the fitting of original data with high accuracy may request higher rank, which limits the improvement of compression ratio. When the compression error is 0.001, which is also acceptable for most ESMD data application, the compression ratio of Adaptive-HGFDR increase to 68.16, which means the compressed data size is 68.16 times smaller than that of original data. At the compression error of 0.001, the compression ratio of Adaptive-HGFDR, ZFP and Blocked-HGFDR are 68.16, 13.42 and 50.78, respectively. The compression ratio of Adaptive-HGFDR is 5.07 times and 1.34 times larger than that of ZFP and Blocked-HGFDR. These may because the Adaptive-HGFDR can adaptively adjust the compression parameter (rank value) according to the actual data complexity, and thus better capture data features to improve the compression ratio.

As the heterogeneity of compression error distribution is important for the data quality of lossy compression, we summarize the error distribution along the longitude dimension as well as the mean value and standard deviation of each method in Figure 4. It is clearly seen that the error distribution of both Adaptive-HGFDR and ZFP are nearly uniform among different longitude dimensions. However, the Blocked-HGFDR shown significant four segments of abrupt changes at different longitude slices. The oscillation characteristic of the three methods are different. For Adaptive-HGFDR, the error distribution are more acted as low-frequency fluctuations while ZFP method are more as higher frequency fluctuations. The Blocked-HGFDR has very different fluctuations characteristics. For the first 0-230 longitude slices, the error distribution of Blocked-HGFDR is of high frequency fluctuations with relatively high frequency, which is similar to ZFP, while in the rest three segments, it has low amplitude, which has similar fluctuations as Adaptive-HGFDR. The error distribution of Adaptive-HGFDR is nearly symmetrical above and below the mean value, while the other two methods are not. To numerically comparison of the mean

value and standard deviation of the error distribution among the three methods. The Adaptive-HGFDR has much smaller standard deviation (**6.89×10^{-6}**), compared with ZFP (**2.94×10^{-5}**) and Blocked-HGFDR (**2.80×10^{-5}**). The Blocked-HGFDR has the smallest mean compression error (**9.35×10^{-5}**), which slightly lower than Adaptive-HGFDR (**9.83×10^{-5}**). While ZFP has the largest mean compression error (**1.29×10^{-4}**).

Both Blocked-HGFDR and Adaptive-HGFDR show the small difference between the adjacent slices and the big difference among the different local block data. Due to the spatio-temporal heterogeneity, the feature distributions of each local ESMD are significantly different, but the feature distributions of adjacent slices have a small difference because of the spatio-temporal similarity. Meanwhile, since the adjacent compressed slice data have similar characteristics, the error fluctuation of these slices is small. On the contrary, the structure difference of each compressed local block data is large, the error fluctuation is also large. In Blocked-HGFDR, the compression parameters of each block are fixed, and the characteristic difference of data of each block is ignored. This weakness is improved in Adaptive-HGFDR by making each block adjust the compression parameters adaptively according to the compression error to achieve the balanced distribution of error. Although the Blocked-HGFDR performs substantially better for several slice numbers, the adaptive HGFDR shows less variations.

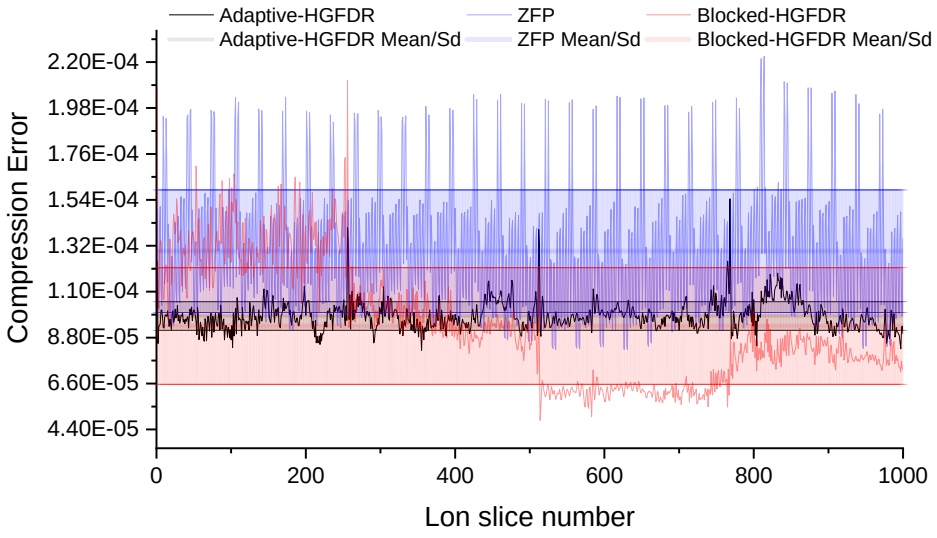

Figure 4. Compression error distribution of three compression methods on longitudinal slices(the slice means the partial data that divided along specific dimensions)

To better reveal the characteristics of the compression error distributions, three random spatial pieces (Height 2,8 and 16) of the detailed distributions of spatial error are depicted in Figure 5. From Figure 5, we can see that the spatial structure of the data is different at different height, there are both continues and abrupted structure changes at different levels. Specifically, the compression error in the Blocked-HGFDR method and the ZFP method fluctuates dramatically, forming multiple peaks and

valleys. The error distributions of ZFP suggest there are high frequency stripes. There are irregular spatial patterns for Blocked-HGFDR. The Adaptive-HGFDR method is more stable where the error distribution is nearly random. Additionally, for spatial structure of the data is different at different height, there are both continues and abrupted structure changes at different levels.

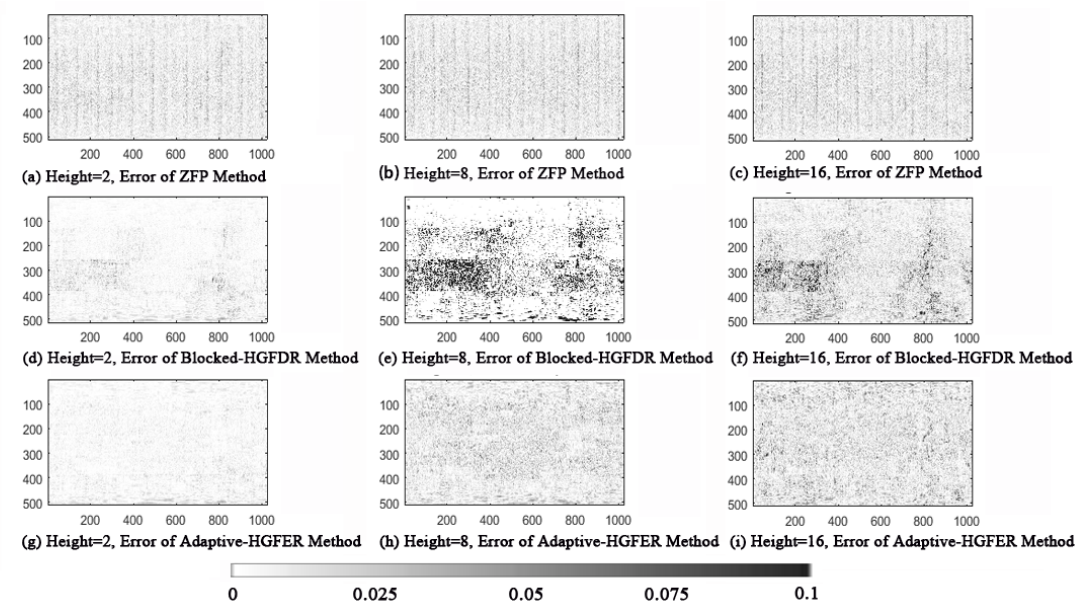

Figure 5. Spatial compression distribution of compression error for three compression methods.

4.4 Evaluation with multiple variables

For a comprehensive comparison of the different methods, 22 monthly climate model data were used as experimental data. Here, we focus on the variables with flux information and fast changing. Among these variables, there are variables with weak spatio-temporal heterogeneity such as the temperature, and the variables with strong spatio-temporal heterogeneity, which will help to better demonstrate the applicability of the method. The dimension of experimental data is 1024×512×221. Here, considering that the compression error and compression performance of each variable can be comparable, the compression error should be not too big or too small for all the 22 variables, the given error is 0.01, the block counts is 256×128×26, and the block count is 144. A detailed description of the variables is shown in Table 1.

Table 1: 22 Descriptions of climate model data variables.

Variable name	Variable description	Variable name	Variable description
FLDS	Downwelling longwave flux at the surface	PCONVT	Convection top pressure
FLDSC	Clearsky downwelling longwave flux at surface	RHREFHT	Reference height relative humidity
FLNSC	Clearsky net longwave flux at surface	SOLIN	Solar insolation
FLNT	Net longwave flux at top of model	SRFRAD	Net radiative flux at surface
FLNTC	Clearsky net longwave flux at top of model	TMQ	Total (vertically integrated) precipitable water

FLUT	Upwelling longwave flux at top of model	TREFHT	Reference height temperature
FLUTC	Clearsky upwelling longwave flux at top of model	TREFMNAV	Average of TREFHT daily minimum
FSDSC	Clearsky downwelling solar flux at surface	TREFMXAV	Average of TREFHT daily maximum
FSNSC	Clearsky net solar flux at surface	TS	Surface temperature (radiative)
FSNTC	Clearsky net solar flux at top of model	TSMN	Minimum surface temperature over output period
FSNTOAC	Clearsky net solar flux at top of atmosphere	TSMX	Maximum surface temperature over output period

The Adaptive-HGFDR, Blocked-HGFDR, and ZFP method were applied to the 22 variables. The compression ratio, time, and standard deviation of the slice error were calculated and shown in Figure 6. Form Figure 6(a), it can be seen that, compared with the other two methods, the compression ratio of Adaptive-HGFDR is the largest. This may because Adaptive-HGFDR considers the coupling relationship among the spatial-temporal dimensions and search for optimal compression parameters at each data blocks. This not only makes the number of features required by each data block is small, but also makes the effect of data heterogeneity on the compression ratio least. Adaptive-HGFDR captures the data features more accurate than the other two methods. The adaptive adjustment of parameters makes Adaptive-HGFDR yield the uniform error distribution for the multiple variables shown in Figure 6(c). In summary, Adaptive-HGFDR provides good adaptability for ESMD. Additionally, the figure 6(a) also shows that the tensor-based compression methods (Adaptive-HGFDR, Blocked-HGFDR) have the high compression ratios for some variables, it may because for tensor-based compression, the relationship between data volume and dimensions is transformed from exponential growth to nearly linear growth by defining the tensor product of tensors, which is essentially the displacement of space by calculating time, so the compression ratio is very high. Also, we can see that with the given compression error, the compression rates of different variables are significant different. It may because different climate model variables have different distribution features. Generally speaking, for the variables with weak spatio-temporal heterogeneity, a small number of feature components can well achieve the accurate approximation that have the high compression rate. While, the variables with strong spatio-temporal heterogeneity may need a large number of feature components that have the low compression rate. Due to the continuous adjustment of compression parameters to search for the optimal rank, Adaptive-HGFDR is the most time consuming [Figure 6 (b)]. Despite this, some optimization strategies, such as the spatiotemporal indexes and the unbalanced block split, can help improving the efficiency of Adaptive-HGFDR.

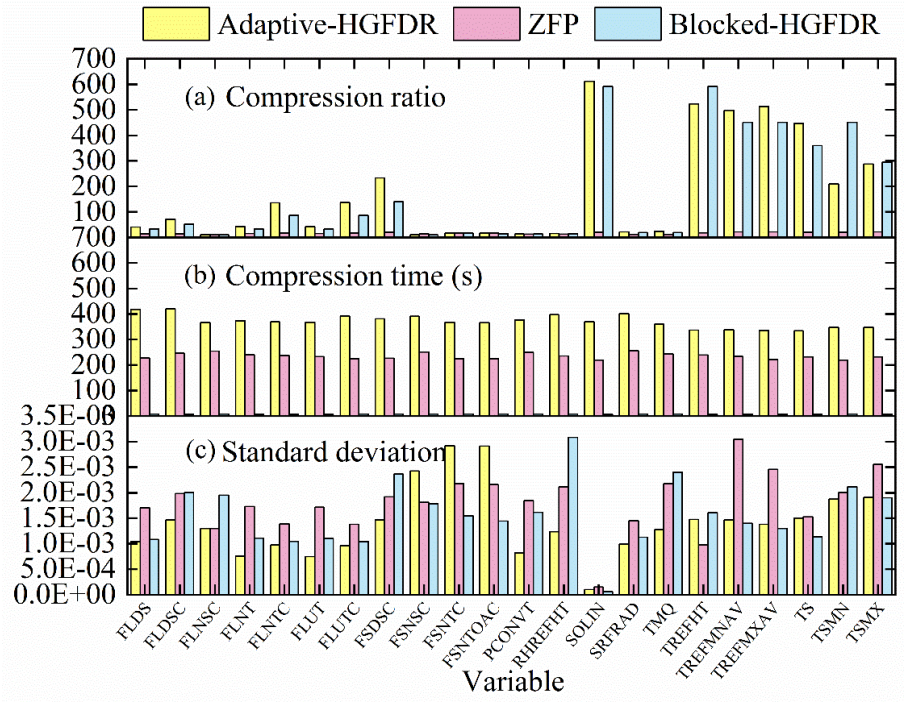

Figure 6. Comparison of the compression results of three compression methods for 22 variables: (a) Compression ratio comparison; (b) compression time comparison; (c) standard deviation comparison of slice error.

5 Conclusion

In this paper, the Adaptive-HGFDR method is developed for ESMD compression. Adaptive-HGFDR achieves the uniform distribution of compression error through blocking data, conducting empirical relationship and searching optimal compression parameters. With the adaptive adjustment of the local compression parameters, our proposed method has better compression performance compared with the traditional methods. Specially, Adaptive-HGFDR can not only maintain a larger compression ratio under the same overall compression error, but can also make the compression error distribution more uniform in all dimensions of the data. Our method is also evaluated with multiple variables. The results show that Adaptive-HGFDR performs well for all 22 variables, indicating the method has large potential to be a universal compression method for high-dimensional data. Additionally, the tensor method can also suitable for in situ observation data, even the data is sparse (Li, D. et al. 2020a), and the data fusion and data synthesis (Wang, P. et al. 2019; Akl, A. et al. 2015). The proposed method can be also applied to sensor data time series. With any data that can be represented as a tensor, it can be compressed with this method. We have also tried to extend the method to be fit for the irregular data that has arbiter boundaries or sparse data (Li, D. et al. 2019). From the perspective of mathematical foundation, tensor can not only support the multidimensional structure but also detect the multidimensional coupling feature. Besides that, the tensor can support many kinds of unstructured multidimensional data with a strict mathematical theory (Li, D. et al. 2020). The current main problem is how to construct the concise and efficient

algorithm, and found, validation, and solve the core science and technology problem of tensor compression in the practical application. we believe that tension-based spatiotemporal data compression must be an important research direction in the future data management of earth system models.

Code and data availability. The Adaptive-HGFDR lossy compression algorithm proposed in this paper was conducted out in MATLAB R2017a. The exact version of Adaptive-HGFDR and experimental data used in this paper is archived on Zenodo (AndyWZJ, 2020; Zhengfang Zhang, 2020). Climate model data [Data set]. Zenodo. http://doi.org/10.5281/zenodo.3997216). The experimental data are Large-scale Data Analysis and Visualization Symposium Data obtained from (OSDC) Open Science Data Cloud. This data set consists of files from a series of global climate dynamics simulations run on the Titan supercomputer at Oak Ridge National Laboratory in 2013 by postdoctoral researcher Abigail Gaddis, Ph.D. The simulations were performed at approximately 1/3-degree spatial resolution, or a mesh size of 1024x512 for 2D. We downloaded this simulation data in the common NetCDF (network Common Data Form) format in 2016 from https://www.opensciencedatacloud.org/.

Author contribution. Zhaoyuan Yu, Linwang Yuan and Wen Luo designed the paper's ideas and methods. Zhengfang Zhang and Yuan Liu implemented the method of the paper with code. Zhaoyuan Yu, Zhengfang Zhang and Dongshuang Li wrote the paper with considerable input from Linwang Yuan. Uzair Aslam Bhatti revised and checked the language of the paper.

Funding. This work was financially supported by the National Natural Science Foundation of China[41625004 41971404] and the National Key R&D Program of China[2017YFB0503500].

Competing interests. The authors declare that they have no conflict of interest.

References

Kuhn, M., Kunkel, J. and Ludwig, T.: Data compression for climate data, Supercomput. Front. Innov., 3(1), 75–94, https://doi.org/10.14529/jsfi160105, 2016.

Wulder, M. A., Masek, J. G., Cohen, W. B., Loveland, T. R. and Woodcock, C. E.: Opening the archive: How free data has enabled the science and monitoring promise of Landsat, Remote Sens. Environ.,122(Complete), 2–10, https://doi.org/10.1016/j.rse.2012.01.010, 2012.

Tilton, H. C., Manohar, M. and Newcomer, J. A.: Earth science data compression issues and activities, Remote Sens. Rev., 9(4), 271-298, https://doi.org/10.1080/02757259409532232, 1994.

Newman, H. B., Ellisman, M. H. and Orcutt, J. A.:Data-intensive e-science frontier research, Commun. ACM, 46(10), 68-77, https://doi.org/10.1145/948383.948411, 2003.

Liang, X., Di, S., Li, S., Tao, D., Nicolae, B., Chen, Z., Cappelo, F., Significantly improving lossy compression quality based on an optimized hybrid prediction model, in the International Conference for High Performance Computing, Networking, Storage and Analysis, Denver Colorado, 17-22 November 2019, 2019.

Santhosh Kumar, V., Nanjundiah, R., Thazhuthaveetil, M. J. and Govindarajan, R.: Impact of message compression on the scalability of an atmospheric modeling application on clusters, Parallel Comput., 34(1), 1–16, https://doi.org/10.1016/j.parco.2007.10.002, 2008.

Tao, D., Di, S., Guo, H., Chen, Z. and Cappello, F.: Z-checker: A Framework for Assessing Lossy Compression of Scientific Data, Int. J. High Perform. Comput. Appl., 33(12), 1-19, https://doi.org/10.1177/1094342017737147, 2017c.

Baker, A. H., Hammerling, D. M., Mickelson, S. A., Xu, H. and Lindstrom, P.: Evaluating Lossy Data Compression on Climate Simulation Data within a Large Ensemble, Geosci. Model Dev.,9(12), 4381-4403, https://doi.org/10.5194/gmd-9-4381-2016, 2016.

Li, B., Zhang, L., Shang, Z. and Dong, Q.: Implementation of LZMA compression algorithm on FPGA, Electron. Lett., 50(21), 1522-1524, https://doi.org/10.1049/el.2014.1734, 2014.

Nathanael Hübbe, Wegener, A., Kunkel, J. M., Ling, Y. and Ludwig, T.: Evaluating lossy compression on climate data, Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), LNCS 7905, 343–356, https://doi.org/10.1007/978-3-642-38750-0_26, 2013.

Lindstrom, P. and Isenburg, M.: Fast and efficient compression of floating-point data, IEEE Trans. Vis. Comput. Graph., 12(5), 1245–1250, https://doi.org/10.1109/TVCG.2006.143, 2006.

Papaioannou, T. G., Riahi, M. and Aberer, K.: Towards Online Multi-model Approximation of Time Series, in 2011 IEEE 12th International Conference on Mobile Data Management, Luleå, Sweden, 6-9 June 2011, 2011.

Tao, D., Di, S., Chen, Z. Z., and Cappello, F.: Exploration of Pattern-Matching Techniques for Lossy Compression on Cosmology Simulation Data Sets, in International Conference on High Performance Computing, Bangkok, Thailand, 18-20 December 2017, 2017a.

Guinness, J. and Hammerling, D.: Compression and Conditional Emulation of Climate Model Output, J. AM. Stat. Assoc.,113(521), 56-67, https://doi.org/10.1080/01621459.2017.1395339, 2016.

Akbudak, K., Ltaief, H., Mikhalev, A. and Keyes, D.: Tile Low Rank Cholesky Factorization for Climate/Weather Modeling Applications on Manycore Architectures, in International Conference on High Performance Computing, Chicago American, 14-16 June 2017, 2017.

Liu, S., Huang, X., Fu, H., Yang, G. and Song, Z.: Data Reduction Analysis for Climate Data Sets, Int. J. Parallel Program., 43, 508-527, https://doi.org/10.1007/s10766-013-0287-0, 2013.

Liu, S., Huang, X., Ni, Y., Fu, H. and Yang, G.: A high performance compression method for climate data, in 2014 IEEE International Symposium on Parallel and Distributed Processing with Applications, Milan, Italy, 26-28 August 2014,2014.

Mummadisetty, B. C.: Performance Analysis of Hybrid Algorithms For Lossless Compression of Climate Data, UNLV Theses, Dissertations, Professional Papers, and Capstones, American, 2015.

Baker, A. H., Xu, H., Dennis, J. M., Levy, M. N., Nychaka, D., Mickelson, S. A., Edwards, J., Vertenstein, M. and Wegener, A.: A methodology for evaluating the impact of data compression on climate simulation data, in HPDC'14: The 23rd International Symposium on High-Performance Parallel and Distributed Computing, Vancouver, BC, Canada, June 2014, 2014.

Berres, A. S., Turton, T. L., Petersen, M., Rogers, D. H. and Ahrens, J. P.: Video Compression for Ocean Simulation Image Databases, in EnvirVis '17: The Workshop on Visualisation in Environmental Sciences, Barcelona, Spain, 2017.

Feng, J., Wu, Z. and Liu, G.: Fast Multidimensional Ensemble Empirical Mode Decomposition Using a Data Compression Technique, J. Clim., 27(10), 3492–3504, https://doi.org/10.1175/JCLI-D-13-00746.1, 2014.

Zabala, A. and Pons, X.: Effects of lossy compression on remote sensing image classification of forest areas, Int. J. Appl. Earth Obs. Geoinf., 13(1), 43–51, https://doi.org/10.1016/j.jag.2010.06.005, 2011.

Taubman, D. and Marcellin, M.: JPEG2000: Image Compression Fundamentals, Standards and Practice, Image Compression Fundamentals, Standards and Practice, 2002.

Castruccio, S. and Genton, M. G.: Compressing an Ensemble With Statistical Models: An Algorithm for Global 3D Spatio-Temporal Temperature, Technometrics, 58(3), 319–328, https://doi.org/10.1080/00401706.2015.1027068, 2016.

Adhianto, L., Banerjee, S., Fagan, M., Krentel, M., Marin, G., Mellor-Crummey, J. and Tallent, N. R.: HPCTOOLKIT: Tools for performance analysis of optimized parallel programs, CONCURR. COMP.-PRACT. E., 22(6), 685–701, https://doi.org/10.1002/cpe.1553, 2010.

Cui, X.-N., Kim, J.-W., Choi, J.-U. and Kim, H.-I.: Reversible Watermarking in JPEG Compression Domain, J. Korea Inst. Inf. Secur. Cryptol., 17(6), 121–130, 2007.

Zheng, Y., Hendrix, W., Son, S. W., Federrath, C., Agrawal, A., Liao, W. K. and Choudhary, A.: Parallel Implementation of Lossy Data Compression for Temporal Data Sets, in 2016 IEEE 23rd International Conference on High Performance Computing, Hyderabad, India, 19-22, December 2017, 2017.

Wilczyñski, K.: SSEM: a computer model for a polymer single-screw extrusion, J. Mater. Process. Tech., 109(3), 308–313, https://doi.org/10.1016/S0924-0136(00)00821-9, 2001.

Wang, N. and Li, X.: A New Low Memory Set Partitioned Embedded Block Coder, Acta Electron. Sin., 34(11), 2068–2071, https://doi.org/10.1198/108571106X99751, 2006.

Diffenderfer, J., Fox, A., Hittinger, J., Sanders, G. and Lindstrom, P.: Error Analysis of ZFP Compression for Floating-Point Data, SIAM J. Sci. Comput., 41, A1867–A1898, https://10.1137/18M1168832, 2019.

Lyre, H.: Holism and structuralism in U(1) gauge theory, Stud.Hist.Phil.Sci.B, 35(4), 643–670, https://doi.org/10.1016/j.shpsb.2004.07.004, 2004.

Schoellhammer, T., Greenstein, B., Osterweil, E., Wimbrow, M. and Estrin, D.: Lightweight temporal compression of microclimate datasets [wireless sensor networks], in 29th Annual IEEE International Conference on Local Computer Networks, Tampa, FL, USA, 16-18 November 2014, 2004.

Tao, D., Di, S. and Cappello, F.: Significantly Improving Lossy Compression for Scientific Data Sets Based on Multidimensional Prediction and Error-Controlled Quantization, in 2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS), Orlando, FL, USA, 29 May-2 June 2017, 2017b.

Li, D., Yang, L., Yu, Z., Hu, Y. and Yuan, L.: A tensor-based interpolation method for sparse spatio-temporal field data, J. Spat. Sci., 65(2), 307-325, https://doi.org/10.1080/14498596.2018.1509740, 2020a.

Yuan, L., Yu, Z., Luo, W., Hu, Y., Feng, L. and Zhu, A. X.: A hierarchical tensor-based approach to compressing, updating and querying geospatial data, IEEE T. Data En. 27(2), 312–325, https://doi.org/10.1109/TKDE.2014.2330829, 2015.

Linton, O. B. and Xiao, Z.: A nonparametric regression estimator that adapts to error distribution of unknown form, Economet. Theor., 23(3), 371-413, https://doi.org/10.1017/S026646660707017X, 2001.

Grasedyck, L.: Hierarchical Singular Value Decomposition of Tensors, SIAM J. Matrix Anal. A., 31(4), 2029-2054, https://doi.org/10.1137/090764189, 2010.

Oseledets, I. V. and Tyrtyshnikov, E. E.: Breaking the Curse of Dimensionality, Or How to Use SVD in Many Dimensions, SIAM J. Sci. Comput., 31(5), 3744-3759, https://doi.org/10.1137/090748330, 2009.

Rouillier, F., Zimmermann, P.: Efficient Isolation of Polynomial's Peal Roots. J. Comput. Appl. Math, 162(1), 33–50, https://doi.org/10.1016/J.CAM.2003.08.015, 2004.

Wang, P., Yang, L. T., Li, J., Chen, J., Hu, S.: Data Fusion in Cyber-physical-social Systems: State-of-the-art and Perspectives. Inform. Fusion., 51, 42–57, https://doi.org/10.1016/J.INFFUS.2018.11.002, 2019.

Akl, A., Yaacoub, C., Donias, M., Costa, J.-P. D., Germain, C.: Texture Synthesis Using the Structure Tensor. IEEE T. Image Process, 24(11), 4082–4095, https://doi.org/10.1109/TIP.2015.2458701, 2015.

Li, D., Gao, H., Luo, W., Yu, Z., & Yuan, L.: Multidimensional Feature Explorer for Unbalanced Spatiotemporal Data. Earth Space Sci., 6(5), 716–729, https://doi.org/10.1029/2018EA000514, 2019.

Li, D., Yu, Z., Wu, F., Luo, W., Hu, Y., & Yuan, L.: The Tensor-based Feature Analysis of Spatiotemporal Field Data With Heterogeneity. Earth Space Sci., 7(2), https://doi.org/10.1029/2019EA001037, 2020b.