Revisiting Wavelet Compression for Large-Scale Climate Data using JPEG 2000 and Ensuring Data Precision

Jonathan Woodring[1]* Susan Mniszewski[1]† Christopher Brislawn[1]‡ David DeMarle[2]§
James Ahrens[1]¶

[1]Los Alamos National Laboratory
[2]Kitware Inc.

Abstract

We revisit wavelet compression by using a standards-based method to reduce large-scale data sizes for production scientific computing. Many of the bottlenecks in visualization and analysis come from limited bandwidth in data movement, from storage to networks. The majority of the processing time for visualization and analysis is spent reading or writing large-scale data or moving data from a remote site in a distance scenario. Using wavelet compression in JPEG 2000, we provide a mechanism to vary data transfer time versus data quality, so that a domain expert can improve data transfer time while quantifying compression effects on their data. By using a standards-based method, we are able to provide scientists with the state-of-the-art wavelet compression from the signal processing and data compression community, suitable for use in a production computing environment. To quantify compression effects, we focus on measuring bit rate versus maximum error as a quality metric to provide precision guarantees for scientific analysis on remotely compressed POP (Parallel Ocean Program) data.

Index Terms: I.3.8 [Computer Graphics]: Applications E.4 [Data]: Coding and Information Theory—Data compaction and compression H.3.m [Information Storage and Retrieval]: Miscellaneous

1 Introduction

Large-scale data are a continual problem for scientific visualization and analysis. While the visualization community has been successful with parallel scaling [18], the main bottlenecks still lie in the movement of data, over networks and from storage [13, 18, 28], and in storage capacity. The leadership supercomputers are unbalanced [47] with respect to compute versus I/O. Furthermore, it is very expensive to provide high-bandwidth storage in conjunction with petascale compute-intensive facilities, e.g., Panasas and Lustre parallel file systems with many parallel high-bandwidth connections. While a large capital investment may be able to solve some of the bandwidth issues between the supercomputer and storage, it does not address bandwidth considerations for off-site data movement, distance visualization, and displays [21].

In the past, the scientific visualization community has provided various data reduction and multi-scale methods for interactive data resources for bandwidth-limited channels. One particular area of interest is wavelet decomposition and compression, due to the localized properties of the compression and the multi-scale nature of the method. Wavelets allow a structured data set to be adaptively scaled, in a focus+context manner, for different target bandwidths and response times. For production climate visualization and analysis, we utilize the collective knowledge of the signal processing and compression community by implementing wavelet compression and multi-scale visualization using JPEG 2000 technology. This allows us to reduce storage and network bandwidth requirements for large-scale scientific data using a standards-based compression technology leveraging the best practices from the compression community. We utilize compressed data in a multi-scale analysis framework implemented in a large-scale visualization tool, ParaView, to analyze remote POP (Parallel Ocean Program) data sets compressed with Kakadu, a JPEG 2000 compliant compression library.

The primary quality metric for compressed data, however, has been average image difference between compressed and uncompressed rendered data, such as root mean squared error. In contrast, our work studies the effects of data value error, rather than image error, via maximum error or the L^∞ norm. By measuring the maximum error of compressed data, we are able to provide a minimum precision guarantee so a domain scientist is assured of the data precision that a compressed source provides. We argue that the emphasis for scientific large-data management ought to be focused on error bounds, such as maximum error rather than average error, because domain scientists are always wary of the effects on their results due to visualization transformations.

By providing precise error bounds of data transformations, the transition from scientific visualization research to domain usage and practice becomes easier. This is because computational-based science is completed with quantitative domain-based analytical tools [12] that rely on data accuracy. Therefore in our opinion, the quality assessment for data transformation in visualization should primarily be focused on data-space accuracy and error metrics. Furthermore, many visualization methods, in particular isosurfacing and thresholding, would be qualitatively and quantitatively improved through precise error constraints on the data [29]. State-of-the-art standards-based compression, combined with data precision guarantees in a multi-scale visualization framework, gives us the tools to visualize and analyze large-scale data in a bandwidth-limited environment.

In Section 2, we describe the related work in compression, wavelet compression, and multi-scale visualization. Section 3 describes our use case and implementation using JPEG 2000 compression. Section 4 describes how we measure precision, visualization of compressed data, and our implementation in multi-scale ParaView. Our performance results are described in Section 5 and we conclude with Section 6.

2 Related Work

Data movement, currently and going into exascale, is one of the top issues for large-scale simulations and analysis [13, 28]. While it is increasingly apparent that more visualization and analysis must be done on-site or *in situ* due to the data movement bottleneck, there

*woodring@lanl.gov
†smm@lanl.gov
‡brislawn@lanl.gov
§dave.demarle@kitware.com
¶ahrens@lanl.gov

IEEE Symposium on Large Data Analysis and Visualization
October 23 - 24, Providence, Rhode Island, USA
978-1-4673-0155-8/11/$26.00 ©2011 IEEE

still are use cases where off-site data analysis and distance visualization requires movement of the data through bandwidth-limited channels, and data compression can be used in these cases. For example when data are curated, shared between collaborators, or the domain scientist does not wish to spend supercomputing cycles on analysis, the data often need to be moved off-site. In another example, one of the distance visualization use cases outlined by [14, 34] moves raw data from the remote site for local processing, rather than images, for higher interactive frame rates. Furthermore, data compression, as an *in situ* data reduction method [55], can be useful to reduce simulation I/O time, post-processing time, and storage space requirements, as long as there are data and error guarantees. Domain scientists currently truncate their data from double precision to single precision for post-processing, and compression is another form of truncation, as long as the precision can be described.

Wavelet decomposition [20, 36, 56, 58] and wavelet compression [25, 26, 30, 42, 50, 54, 52] have been extensively used in the past for both data reduction and multi-scale visualization. Additionally, the scientific visualization community has devised other non-wavelet based compression [17, 33], quantization [44], and multi-scale methods [14, 31, 37] to handle large-scale data sets. In this work, we leverage the expert knowledge of the signal processing and data compression communities by using the JPEG 2000 standard for compression and multi-scale representation of scientific data, discussed in more detail in Section 2.1.

Error metrics are an important issue when dealing with compressed data [22, 24, 32, 44, 53], and data error has been used in these instances for controlling multi-scale refinement. Though, the final quality assessment usually has been evaluated as the average image difference between rendered compressed and non-compressed data. Additionally, there has been work in visualization of uncertainty and error [29, 39, 43, 55], to show the amount of variance in data, sample data, and ensembles. In our work, we measure and provide a maximum error guarantee when working with the compressed data, so that a domain scientist is assured that the data are accurate to the *x*th decimal place, where *x* is order of magnitude of the maximum error between compressed and non-compressed data. Thus, our evaluation on compressed data is that we provide a data quality guarantee for the domain scientist, because we provide a conservative precision metric up front.

2.1 Related Work in Wavelet Compression

Wavelet transforms have seen a variety of applications to scientific visualization in recent years, as described in the previous related work and the numerous references therein. Many of these studies have used wavelet transforms, usually in the form of their digital incarnations, known as *multi-rate filter banks*, to provide data reduction and multi-resolution data hierarchies for interactive rendering of high-resolution pixel or voxel data. While many authors have also noted the bandwidth crisis associated with the more mundane tasks of transmitting and storing all of this data, the focus in the scientific visualization literature has been on the difficult problems associated with interactive visualization of large-scale scientific data sets.

Some visualization researchers have gone beyond using wavelet transforms solely as data structures and have obtained significant bandwidth reductions by employing modern source coding techniques from digital communications theory [45, 40]. For example, Guthe et al. [25] and Wang et al. [52] describe schemes in which small wavelet transform coefficients are zeroed out by hard-thresholding. The thresholded data is then zero-runlength encoded, quantized if the wavelet coefficients are floating-point (as with the SPOT data discussed in [52]), and finally compressed using Huffman coding.

While this general approach is capable of providing reasonably high-quality data reconstruction at reduced bit rates (as measured in bits transmitted per pixel) for an acceptable computational burden, it suffers from an inherent shortcoming: lack of rate-versus-distortion scalability. Huffman encoder design produces a variable-length prefix-free codebook for a given IID (independent and identically distributed random variables) source that minimizes the expected number of bits (i.e, the codeword length) per symbol encoded, but Huffman coding does not enable partial or progressive decoding of a compressed bitstream. This means that all symbols must be decoded completely, at the full precision with which they were represented at encode-time, implying that the encoder rather than the end-user gets to decide data fidelity and how much bandwidth the end-user must accept. This is particularly limiting in applications like large-scale scientific data analysis, in which multiple end-users can have widely differing requirements ranging from low-precision reduced-resolution global browsing to high-fidelity visualization or floating point exploitation of small fractions of the data.

In response to these limitations, one goal of our research is to bridge this gap between the current state of visualization research and the need for standards-based, computationally feasible, highly scalable scientific source coding methods for production scientific computing. Fortunately, highly scalable source coding methods have evolved over the past few decades that address at least some of these issues. Transform-based image coding techniques [27, 57] emerged in the 1970's and 80's as computer capabilities evolved to the point where PC-scale devices could handle the computational requirements, leading to the publication of the ISO/IEC JPEG standard [5, 38] and the FBI Wavelet/Scalar Quantization fingerprint image compression specification [4, 15] in the early 1990's. These developments, particularly wavelet transform and sub-band coding techniques, quantization theory [23], binary embedded arithmetic bitplane coding, and highly scalable code-stream architectures [48] led in turn to the latest generation of image source coding technology, the ISO/IEC JPEG 2000 family of standards [35, 19, 51, 41, 16]. The JPEG 2000 books [49, 11] provide extensive details about the theory behind JPEG 2000 and the baseline (Part 1) standard [6].

The main benefit from using a compression standard is that it leverages the collective knowledge of the compression community and, through multiple implementations, it is more likely that support for JPEG 2000 compressed data will exist in the future, which is important for production-based scientific computing. There exist several open-source JPEG 2000 implementations, most notably the JasPer implementation [1, 7], which comprises the ISO reference implementation of JPEG 2000 Part 1 [6], and the OpenJPEG project [2]. For our research, however, we chose the commercial Kakadu implementation [3] because of its extensive support for JPEG 2000 Part 2 extensions [8], the Part 9 interactive client-server protocol (JPIP) [9], and other features described in Section 3.1. Even though Kakadu supports Part 9, we did not utilize it in this research and used standard methods for data transfer of remote compressed data instead.

3 Use Case and Implementation

At Los Alamos National Laboratory (LANL), the climate scientists run daily Parallel Ocean Program (POP) [46] simulations on Oak Ridge National Laboratory (ORNL) supercomputers. In addition to using the remote supercomputer for distance visualization via images and for batch movie generation, there is also a need to move the data to LANL for local analysis by collaborators and team members. Due to the large size of the data (1.4 GB per field per time slice, 3600 × 2400 × 42 single precision floats) and the low single-link bandwidth between LANL and ORNL (measured at approximately 1MB/s for a serial link), it takes approximately 23 minutes to copy one field for one time slice.

Considering there are typically four fields that the climate sci-

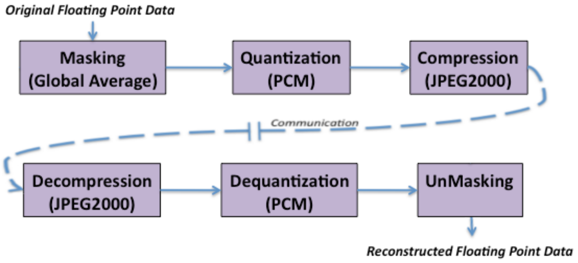

Figure 1: The compression-decompression pipeline for our target use case.

entists are interested in (salinity, temperature, east-west velocity, north-south velocity), that brings the total up to 5.6 GB per time slice, and a transfer time of 72 minutes per time slice, over a serial link. A normal sized set of POP time slices, that can have from tens to thousands of time steps, brings the total data size up to 56 GB for ten time steps or 5.6 TB for a thousand time steps. The serial transfer time in this case would take 12 hours for the former and 50 days for the latter case. At this point, it would be more efficient to utilize the high-bandwidth, high-latency "station wagon" transfer (or possibly RFC 1149) to send terabyte hard drives via the mail.

Although parallel file transfers can moderately speed up the amortized transfer time between ORNL and LANL, we explore the complementary technique of using compression to reduce the data sizes and thereby speed up transfer times. Figure 1 describes our compressed data management pipeline from the remote site to the local site. Assuming we have raw floating point data generated by the POP simulation, we perform a series of steps to compress and decompress the data for transfer from ORNL to LANL.

3.1 Wavelet Compression by JPEG 2000

For our wavelet compression, we used the Kakadu implementation of the JPEG 2000 image compression, described earlier in Section 2.1. The main benefits of Kakadu are that the software is based on the JPEG 2000 standard, it is a supported software package suitable for production computing, and it is the fastest known implementation of JPEG 2000 as of this writing. Kakadu also supports codestream packet-ordering optimization for multiple decoding bit rates, which in our research was usually chosen to be powers-of-two ranging from 0.25 to 16 bits/pixel. This means that not only can the end-user decode the codestream for arbitrary bit rates, multiple (power-of-two) spatial resolutions, user-selected components, and user-selected regions of interest, but that the data will be optimal at each of the preset rates, even though all reconstructions are obtained from the same compressed file (i.e., no need for separate encodings at each preset bit rate).

We have to apply several preconditioning steps to prepare POP data for the wavelet compression. Although the Kakadu implementation supports the JPEG 2000 Part 2 capabilities for three-dimensional data transformations, in these experiments we treated three-dimensional data as a stack of 2D slices with no z-axis decorrelating transforms. The reason is that we focused on 2D slices is because there is little to no effect on compression rates and data error when including the z-axis for POP data (the amount of effort for the relative gain is lost in the noise). Therefore, we individually compressed z-slices of the POP data as 2D images, by field and time slice treating each z-slice as a JPEG 2000 image component.

3.1.1 Adjusting Masked Data

A limitation is that JPEG 2000 expects logically rectangular data. The POP data have masked areas (the landmasses) in the simulation output, as the valid data are only calculated in the ocean areas. The masked areas in the stored output cause large edge discontinuities between the valid data in the oceans and the masked data on the land, because the masked area is an arbitrary constant value (-10^{34}). Since JPEG 2000 compresses the entire 2D area and due to the value discontinuities at the mask edge, it causes inefficient wavelet compression at the edges and is a large source of errors between compressed and non-compressed data.

To correct this, we interpolate the data in the masked areas to reduce the discontinuity between the valid data and the masked data. We smooth the masked areas for more efficient compression by replacing the constant value with the value average over the entire field and time slice, prior to quantization. We have considered using other interpolations, such as a blurring filter at the edges of the mask, but they have not been implemented as of this writing. Since the mask is fixed across time steps, it is assumed that it is transmitted prior to compression, such that both the remote and local side have access to the mask.

3.1.2 Quantizing Floating Point Data

The capability most lacking in JPEG 2000 from the standpoint of scientific visualization is support for floating point input. While floating point support was originally proposed for inclusion in JPEG 2000 Part 10 [10], there was not sufficient international involvement to complete the development of floating point extensions. Consequently, JPEG 2000 is currently limited to fixed-precision input and can be as high as 32 bits/pixel. In practice, implementations like Kakadu can handle less input precision, and the experiments reported below involved pre-quantization of the floating point data (typically to around 25 bits per data point) before sending it into Kakadu.

This quantization was done using adaptive PCM (pulse code modulation, also known as uniform scalar quantization) designed for each input component, a relatively low-complexity preprocessing step. The bin ranges for the quantization are the minimum to the maximum floating point values for a field and time slice. The floating point values are normalized and binned to the nearest integer between 0 and $N-1$, where $N = 2^R$ is the number of bins and R is the number of bits for quantization. The achieved entropy after quantization is less than or equal to R. A point to note for the data ranges in the POP data is that the 25 bit fixed point quantization should be enough to capture most of the dynamic range with little to no loss in precision (23 bit mantissa for single floating point precision). We do not quantize with fewer bits and rely on bit rate variation inherent in JPEG 2000 to reduce data sizes. This is because in our studies we saw that the resulting data errors were worse at the same data sizes when using smaller bit quantization than without.

4 Measuring Precision and Visualization of Compressed Data

Assessing performance from a signal processing and visualization perspective generally relies on signal-to-noise ratio (SNR), root mean squared error (RMSE), or the L^2 norm, which are all analogous measures for compression quality. While they are good for measuring average performance of the compression, it is not conservative enough to use as a data quality metric for the scientific analysis of compressed data. These average metrics can "hide" high pointwise errors in the compressed data. While the overall average error may be good and only a few points have high error, those specific points that can create problems for local analysis because the error is not bounded nor described to a domain scientist.

For localized scientific analysis, like scientific feature finding, it is more useful to overspecify and provide the worst case, maximum pointwise error (L^∞ norm), rather than average error (L^2 norm). Unbounded errors, like average error, can have unknown effects on localized scientific features. By providing an upper bound on the

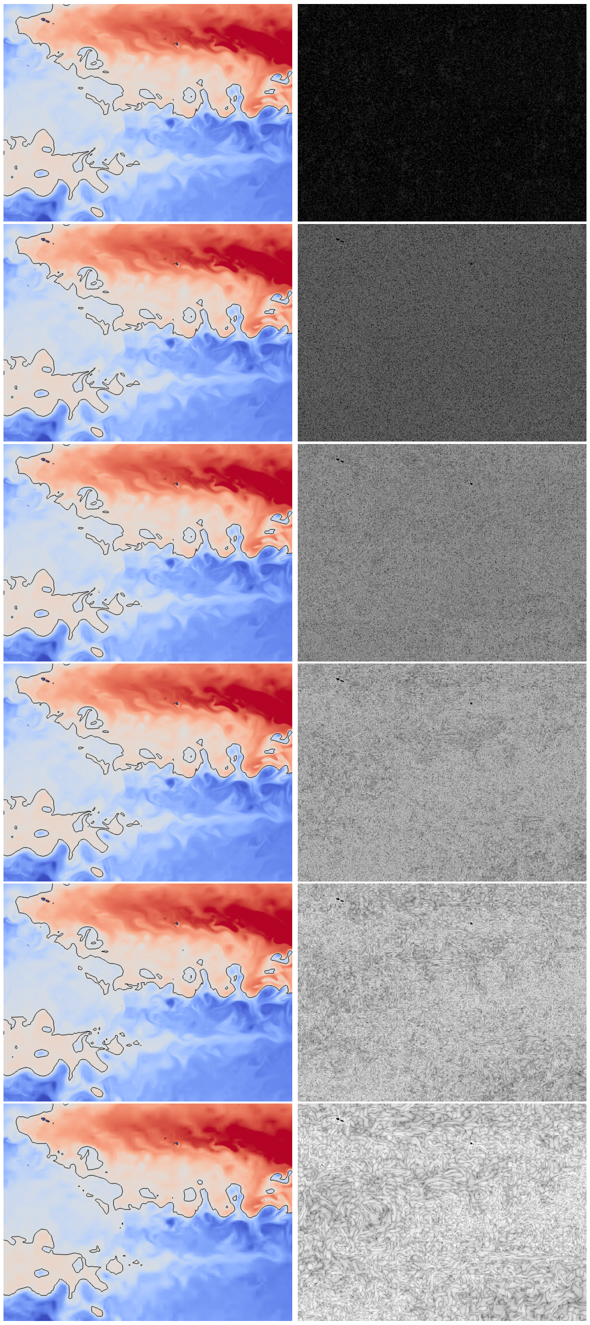

Figure 2: POP salinity data with an isoline filter are shown in color. Paired with it are pointwise errors are shown in black and white, with log-scale coloring. Compression bit rate decreases, top to bottom: 8, 4, 2, 1, 0.5, and 0.25. It is difficult to see the variation between the visualizations of the compressed data, while the localized errors are much more telling. The maximum errors from largest bit rate to smallest are: 1.49e-09, 5.31e-07, 5.23e-06, 2.31e-05, 8.59e-05, and 0.000303. The data range from 0.00421 to 0.04209. The worst relative data error is 0.8% at 0.25 bits per pixel.

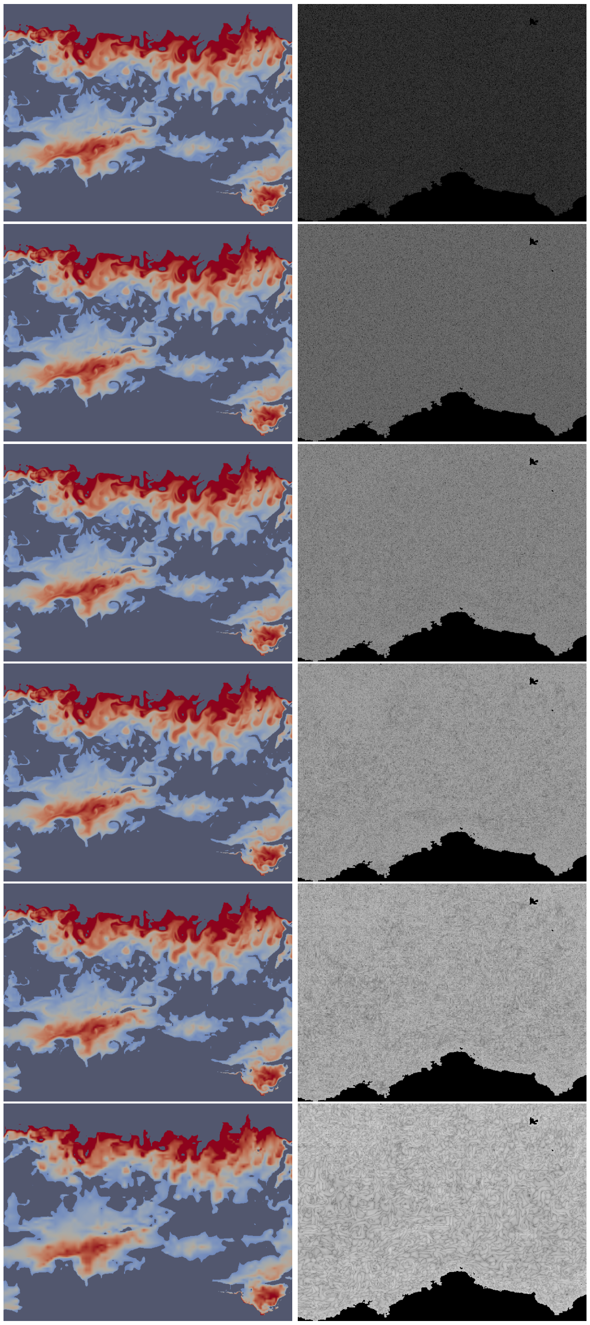

Figure 3: POP temperature data with an threshold filter are shown in color. Paired with it are pointwise errors are shown in black and white, with log-scale coloring. Compression bit rate decreases, top to bottom: 8, 4, 2, 1, 0.5, and 0.25. It is difficult to see the variation between the visualizations of the compressed data, while the localized errors are much more telling. The maximum errors from largest bit rate to smallest are: 7.63e-06, 2.29e-05, 0.00173, 0.0188, 0.811, 0.293, and 1.54. The data range from -1.99987 to 31.197. The worst relative data error is 4.6% at 0.25 bits per pixel.

maximum error, we provide a minimum precision guarantee that the data are precise to *x* decimal places, beyond which the scientist can expect noise with increasing probability in the least significant bits. In our studies, approximately 20% to 40% of compressed POP data have error greater than RMSE. Therefore, maximum error acts as a bounding error epsilon in which the decimal places are uncertain beyond a fixed precision.

We measure the maximum error by directly comparing each data point in compressed data with the original data set and recording the L^∞ norm. The minimum precision, can be calculated as the order of magnitude of the L^∞ norm. In Figures 2 and 3, we show POP salinity data with an isoline filter and POP temperature data with a thresholding filter, with decreasing bit rate, top to bottom. The black and white images show the pointwise error with a log-scale transfer function to contrast against the visualizations, to show that while the visualizations may seem similar, it may hide localized data error that can affect scientific analysis. This is one reasons that domain scientists have been mistrusting or reluctant to adopt scientific visualization methods. With the L^∞ metric, a domain scientist can be assured that compressed data are accurate to a fixed precision, rather than an average assurance, such as RMSE. In Section 5, we show the relationship of maximum error to RMSE.

4.1 ParaView Multi-Scale Integration

Prior to visualization and analysis, scientific data are compressed to storage as JPEG 2000 data files (JPEG 2000 bitstream and *.jpx meta information headers), after the necessary steps to prepare the data for compression. After compression, the entire dataset or a spatial sub-region can be requested during decompression in JPEG 2000, at varying bit rate decomposition levels. Progressive updating of the decompressed data can be achieved by iteratively updating from the lowest resolution to the highest, using a stored buffer of low-resolution data and progressively refining it, via the JPEG 2000 bitstream storage mechanism.

Thus, JPEG 2000 implements the necessary semantics for multi-scale visualization of image and volume data. For the visualization and analysis of POP data, we used multi-scale ParaView as described in [14]. Within ParaView/VTK, we implemented a Kakadu JPEG 2000 reader module with additional filters specifically for analysis and comparison of compressed data: MSE, RMSE, SNR, error variance, error mean, maximum error, and relative maximum error. The JPEG 2000 reader module acts as an interface between the ParaView/VTK multi-scale pipeline semantics and the JPEG 2000 multi-scale semantics, similar to the VAPOR [20] reader which can read VAPOR wavelet data within ParaView.

5 Performance Results

Our motivating case study and the scientific data we used is from a $\frac{1}{10}^\circ$ climate simulation using the Parallel Ocean Program (POP) [46]. POP simulations generated remotely at ORNL are run in double precision, but the results are then truncated and written to disk in single precision before compressing the data. Future work could directly incorporate the JPEG 2000 compression in the simulation I/O routine, rather than using JPEG 2000 as a post-processing operation, which removes one read and one write of the data, reducing the end-to-end workflow time [55].

A time step in a POP data set consists of a 3600 × 2400 × 42 rectilinear grid with four floating point scalar fields (or two scalar fields and one vector field): salinity (SALT), temperature (TEMP), east-west flow velocity (UVEL), and north-south flow velocity (VVEL). The data value ranges for each are, respectively: 0.00421 to 0.04209 g of salt/g of water, -1.99987 to 31.197 degrees Celsius, -159.684 to 230.467 cm/s, and -179.224 to 218.375 cm/s. The continents are represented as masked values, that are replaced with the average value for a time step, and represent 37.5% of the points in the data set in the top slice, going up to 98.9% of the points in the bottom slice.

We used an "irreversible" (i.e., linear) and "reversible" (i.e., lossless but non-linear) 5-tap/3-tap wavelet filter bank from JPEG 2000 Part 1 for compression after 25 bit PCM quantization, storing five levels of progressive decomposition in the JPEG 2000 bitstream. The 2D slices of climate data were compressed at various bit rates after quantization: the maximum bit rate with the 5-3 lossless, the maximum bit rate with 5-3 lossy and other bit rates of 8, 4, 2, 1, 0.5, and 0.25 with 5-3 lossy.

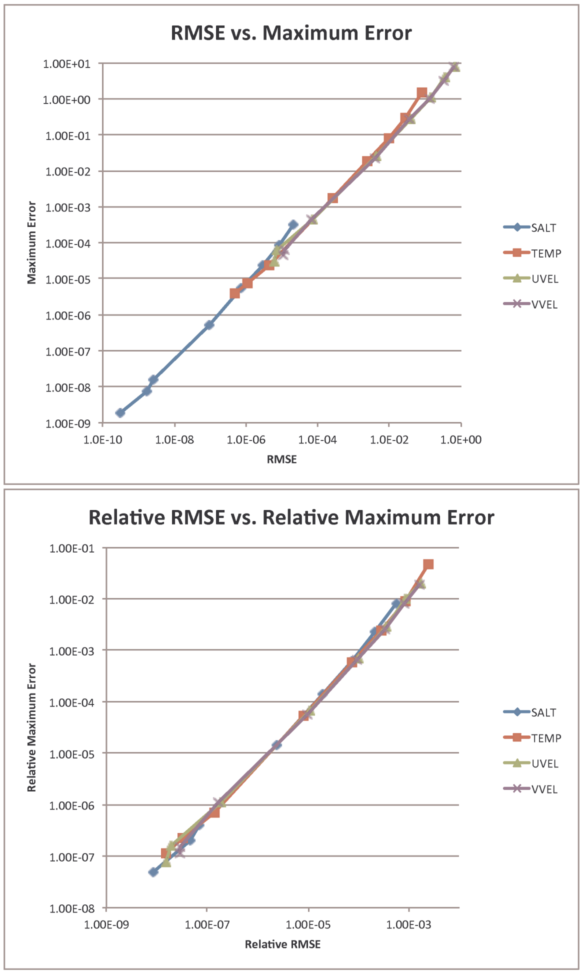

Figure 4: Root mean squared error (RMSE) versus maximum error, absolute and relative error. The points represent various compression rates and the RMSE is measured along with the maximum error for the data (higher error is found in lower bit rates). In the compressed POP data, there is a nearly linear relationship between the L^2 norm and the L^∞ norm. The L^∞ is roughly one order of magnitude greater than the L^2 norm.

In our comparisons with POP and JPEG 2000, we primarily measured (L^∞ norm) as it is a better metric to evaluate compressed data, because it provides a minimum precision error bound for scientific analysis. To compare against average error, we found that there is a nearly relationship between root mean squared error (RMSE, L^2 norm) and maximum error (L^∞ norm), seen in Figure 4. The difference between L^∞ and L^2 is roughly one order of magnitude. We see that the relationship between RMSE and maximum error deviates at lower bit rates, as the maximum error increases faster. The maximum error at various bit rates can be see in Figure 5, where it is more apparent that the maximum error accelerates at lower bit rates beyond 2.

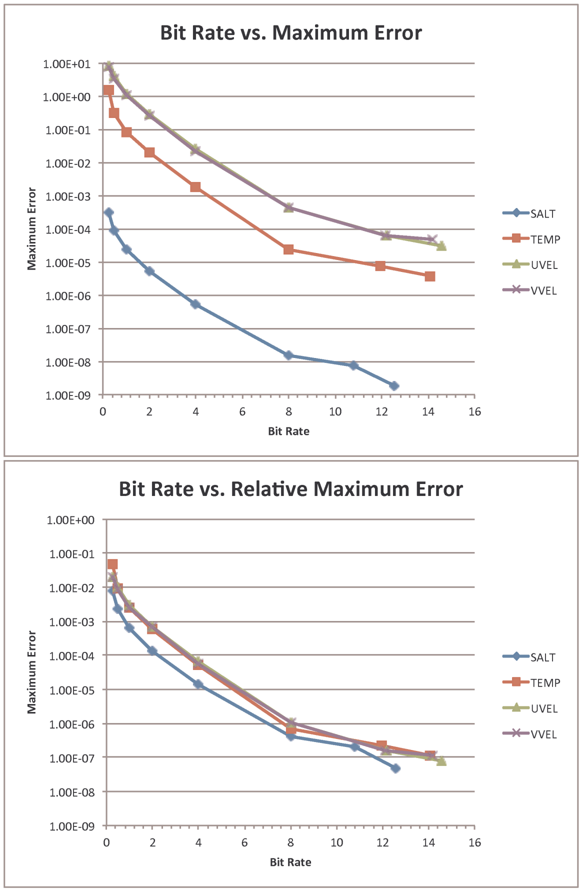

Figure 5: The relationship between bit rate and maximum data error, absolute and relative error. The maximum error drop-off can be seen beginning at around bit rate of 2 in compressed POP data.

Additionally, we look at the compression effects on the maximum error in the partial derivatives in x of the POP field data, seen in Figure 6, as the derivatives are an important measure used by both analysis and visualization. We only show the partial derivative of x because the graph for y is nearly the same. At lower bit rates if we compare to Figure 5, we see that there is an interesting counter trend when compared to the overall data error. At low bit rates, the maximum error accelerates in the scalar fields, but the maximum error decelerates in the partial derivatives. This can be clearly seen in Figure 7, where we compare the maximum error in the data fields versus the maximum error of the partial derivatives of the fields. There is nearly a one for one relationship between data error and derivative error, except at the low bit rates.

Figure 8 shows the graphs that we would provide the scientists to show the relationship between maximum error and the resulting transfer time. They allow a domain scientist to trade between data accuracy and transfer time in a controlled and quantified way, as they are able to easily see the relationship between maximum error and the transfer time. The upper-most point in both graphs is the transfer time for uncompressed POP data, which takes 23 minutes to transfer from ORNL to LANL, in our 1MB/s case. Though the uncompressed data are "infinitely" precise, we put the point at 10^{-10} to be able to graph it. (In actuality, 10^{-10} is an over-estimate for the precision, as machine epsilon for single precision floating point data is approximately 10^{-8}.) The next point down is the transfer time for data compressed with the 5-3 lossless filter that reduces the transfer time to 10 minutes. The points after that are the results for the 5-3 lossy filter with various bit rates, with increasingly faster transfer times, but with larger maximum errors.

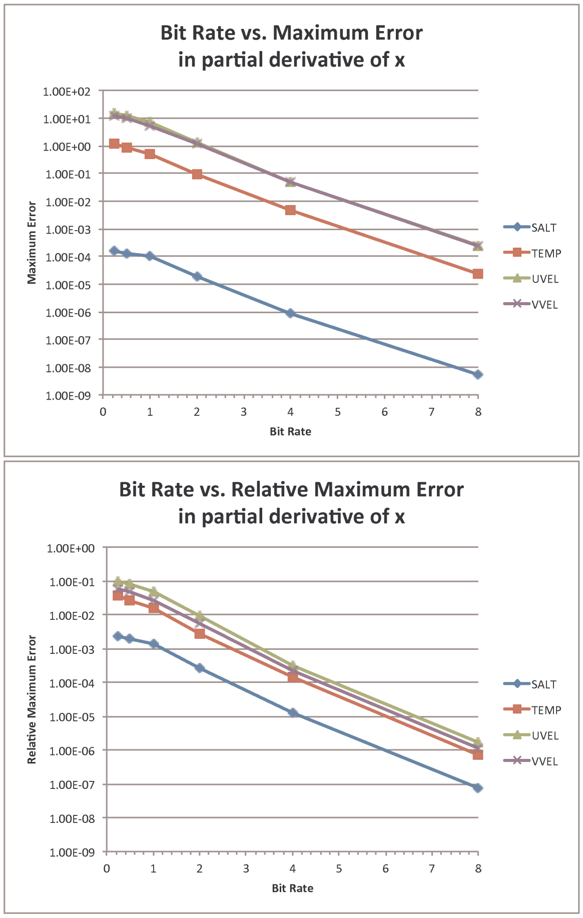

Figure 6: The relationship between bit rate and maximum error in the partial derivative of x, absolute and relative error. For brevity, we do not show the partial derivative in y because it is nearly the same. The error in the derivatives decelerates at lower bit rates.

6 Conclusion

We have described the use of JPEG 2000 for the compression of scientific climate data in a production scientific computing environment. With a standards-based data compression technique, we are able to deliver large-scale data over bandwidth-limited links in a controlled and quantified manner. The primary metric that we used is maximum data error (L^∞ norm), because it allows us to provide a precision guarantee for the domain scientist, giving an exact error bound on the data transformation. By using JPEG 2000, we leverage the best practices and the state-of-the-art methods provided by the signal processing and data compression community. Additionally by using compression standards, it is likely that the methods will be supported in the future to decompress any archived climate data for future use. With this work, we are able to control the size of large-scale data in a quantified, scalable manner for the bandwidth-limited environments that will prevail into exascale and for the foreseeable future.

References

[1] http://www.ece.uvic.ca/~mdadams/jasper/.

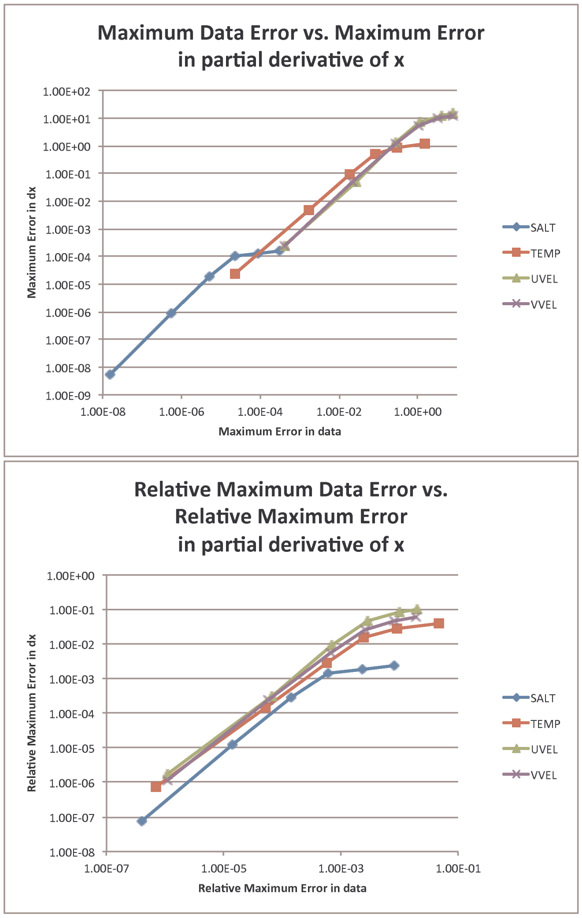

Figure 7: The relationship between maximum data error and maximum error in the partial derivative of x, absolute and relative error. For brevity, we do not show the partial derivative in y because it is nearly the same. As we can see there is nearly a one to one relationship between data error and derivative error, except at low bit rates. The derivative error does not increase as fast as the data error at low bit rates.

[2] http://www.openjpeg.org/.

[3] http://www.kakadusoftware.com.

[4] *WSQ Gray-Scale Fingerprint Image Compression Specification*, IAFIS-IC-0110v2, Washington, DC, Feb. 1993. US Federal Bureau of Investigation.

[5] *Information technology—Digital Compression and Coding of Continuous-Tone Still Images, Part 1: Requirements and Guidelines*, ISO/IEC Int'l. Standard 10918-1. Int'l. Org. Standardization, 1994.

[6] *Information technology—JPEG 2000 image coding system, Part 1*, ISO/IEC Int'l. Standard 15444-1, ITU-T Rec. T.800. Int'l. Org. Standardization, Dec. 2000.

[7] *Information technology—JPEG 2000 Image Coding System, Part 5 (Reference Software)*, ISO/IEC Int'l. Standard 15444-5. Int'l. Org. Standardization, Nov. 2003.

[8] *Information technology—JPEG 2000 image coding system, Part 2: Extensions*, ISO/IEC Int'l. Standard 15444-2, ITU-T Rec. T.801. Int'l. Org. Standardization, May 2004.

[9] *Information technology—JPEG 2000 Image Coding System, Part 9: Interactivity tools, APIs, and protocols*, ISO/IEC Int'l. Standard 15444-9. Int'l. Org. Standardization, Oct. 2004.

[10] *Information technology—JPEG 2000 image coding system, Part 10: Extensions for three-dimensional data*, ISO/IEC Int'l. Standard 15444-10, ITU-T Rec. T.809. Int'l. Org. Standardization, 2008.

[11] T. Acharya and P.-S. Tsai. *JPEG2000 Standard for Image Compression: Concepts, Algorithms and VLSI Architectures*. Wiley-Interscience, 2004.

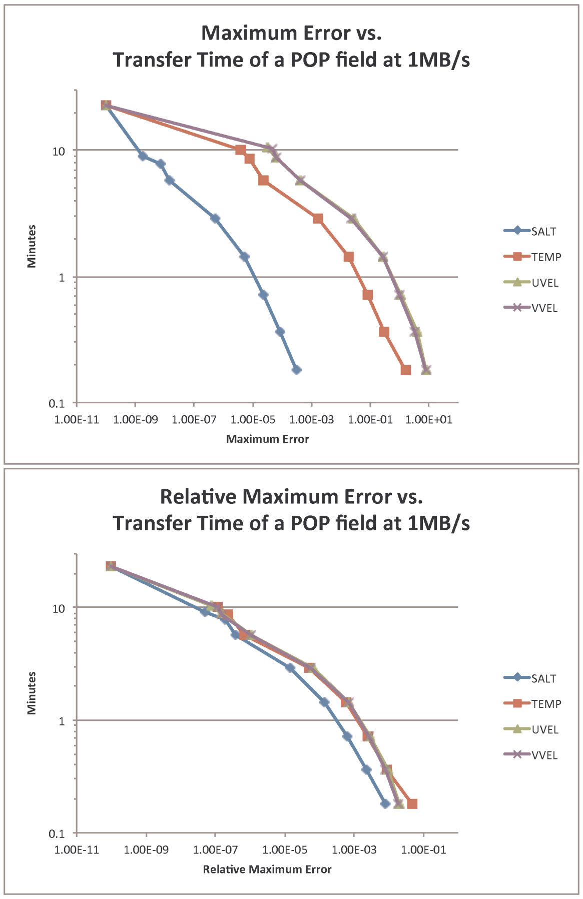

Figure 8: The relationship between maximum error to transfer time, absolute and relative error. These are the graphs we provide a domain scientist to trade between data quality and transfer time over a bandwidth- limited link. The graphs show the transfer time for one time step and one field with the bandwidth of 1MB/s, the same bandwidth as our scientists in this case study between LANL and ORNL. The original data transfer time is 23 minutes per field, while the highest compression bit rate (which is nearly indistinguishable from lossless because the error is close to the machine epsilon for single precision) can cut the transfer time to less than half (10 minutes). Lower bit rates can considerably reduce the transfer times with quantified error effects.

[12] J. Ahrens, K. Heitmann, M. Petersen, J. Woodring, S. Williams, P. Fasel, C. Ahrens, C. Hsu, and B. Geveci. Verifying scientific simulations via comparative and quantitative visualization. *IEEE Computer Graphics and Applications*, 30(6):16–28, 2010.

[13] J. Ahrens, B. Hendrickson, G. Long, S. Miller, R. Ross, and D. Williams. Data intensive science in the department of energy. Technical Report LA-UR-10-07088, Los Alamos National Laboratory, Oct. 2010.

[14] J. Ahrens, J. Woodring, D. DeMarle, J. Patchett, and M. Maltrud. Interactive remote Large-Scale data visualization via prioritized Multi-Resolution streaming. In *Proceedings of the 2009 Workshop on Ultrascale Visualization*, pages 1–10, Portland, Oregon, 2009.

[15] C. M. Brislawn, J. N. Bradley, R. J. Onyshczak, and T. Hopper. The FBI compression standard for digitized fingerprint images. In *Proc. SPIE*, volume 2847, pages 344–355, Denver, CO, Aug. 1996. SPIE.

[16] C. M. Brislawn and M. D. Quirk. Image compression with the JPEG-2000 standard. In R. G. Driggers, editor, *Encyclopedia of Optical*

Engineering, pages 780–785. Dekker, New York, 2003.

[17] M. Burtscher and P. Ratanaworabhan. FPC: a High-Speed compressor for Double-Precision Floating-Point data. *Computers, IEEE Transactions on*, 58(1):18–31, 2009.

[18] H. Childs, D. Pugmire, S. Ahern, B. Whitlock, M. Howison, Prabhat, G. Weber, and E. Bethel. Extreme scaling of production visualization software on diverse architectures. *Computer Graphics and Applications, IEEE*, 30(3):22–31, 2010.

[19] C. Christopoulos, A. Skodras, and T. Ebrahimi. The JPEG 2000 still image coding system: An overview. *IEEE Trans. Consumer Electron.*, 46(4):1103–1127, Nov. 2000.

[20] J. Clyne, P. Mininni, A. Norton, and M. Rast. Interactive desktop analysis of high resolution simulations: Application to turbulent plume dynamics and current sheet formation. *New Journal of Physics*, 9(8):301–301, 2007.

[21] M. F. Deering. The limits of human vision. In *2nd International Immersive Projection Technology Workshop*, 1998.

[22] D. Ellsworth, L. Chiang, and H. Shen. Accelerating time-varying hardware volume rendering using TSP trees and color-based error metrics. *Proceedings of the 2000 IEEE symposium on Volume visualization*, pages 119–128, 2000. ACM ID: 353908.

[23] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer, Norwell, MA, 1992.

[24] S. Guthe and W. Strasser. Advanced techniques for high-quality multiresolution volume rendering. *Computers & Graphics*, 28(1):51–58, Feb. 2004.

[25] S. Guthe, M. Wand, J. Gonser, and W. Strasser. Interactive rendering of large volume data sets. In *Visualization Conference, IEEE*, pages 50–60, Los Alamitos, CA, USA, 2002. IEEE Computer Society.

[26] I. Ihm and S. Park. Wavelet-based 3D compression scheme for very large volume data. In *Graphics Interface*, pages 107–116, 1998.

[27] N. S. Jayant and P. Noll. *Digital Coding of Waveforms*. Prentice-Hall, Englewood Cliffs, NJ, 1984.

[28] C. Johnson and R. Ross. Visualization and knowledge discovery: Report from the DOE/ASCR workshop on visual analysis and data exploration at extreme scale. Technical report, Department of Energy Office of Science ASCR, Oct. 2007.

[29] C. Johnson and A. Sanderson. A next step: Visualizing errors and uncertainty. *Computer Graphics and Applications, IEEE*, 23(5):6–10, 2003.

[30] T. Kim and Y. Shin. An efficient wavelet-based compression method for volume rendering. In *Computer Graphics and Applications, 1999. Proceedings. Seventh Pacific Conference on*, pages 147–156, 1999.

[31] E. LaMar, B. Hamann, and K. Joy. Multiresolution techniques for interactive Texture-Based volume visualization. In *Proceedings of the conference on Visualization'99: celebrating ten years*, pages 355–361, 1999.

[32] E. LaMar, B. Hamann, and K. Joy. Efficient error calculation for multiresolution texture-base volume visualization. In *Hierarchical and geometrical methods in scientific visualization*, pages 51–62. Springer, Feb. 2003.

[33] P. Lindstrom and M. Isenburg. Fast and efficient compression of Floating-Point data. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):1245–1250, 2006.

[34] E. Luke and C. Hansen. Semotus visum: A flexible remote visualization framework. In *IEEE Visualization, 2002. VIS 2002.*, pages 61–68, Boston, MA, USA, 2002.

[35] M. W. Marcellin, M. J. Gormish, A. Bilgin, and M. P. Boliek. An overview of JPEG 2000. In *Proc. Data Compression Conf.*, pages 523–541, Snowbird, UT, mar 2000. IEEE Computer Soc.

[36] S. Muraki. Approximation and rendering of volume data using wavelet transforms. *Proceedings of the 3rd conference on Visualization '92*, pages 21–28, 1992. ACM ID: 949694.

[37] V. Pascucci and R. Frank. Global static indexing for Real-Time exploration of very large regular grids. In *Supercomputing, ACM/IEEE 2001 Conference*, page 45, 2001.

[38] W. B. Pennebaker and J. L. Mitchell. *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold, New York, NY, 1992.

[39] K. Potter, J. Kniss, R. Riesenfeld, and C. Johnson. Visualizing summary statistics and uncertainty. *Computer Graphics Forum*, 29(3):823–832, 2010.

[40] J. G. Proakis. *Digital Communications*. McGraw-Hill, New York, NY, 4 edition, 2000.

[41] M. Rabbani and R. Joshi. An overview of the JPEG 2000 still image compression standard. *Signal Process. Image Commun.*, 17:3–48, 2002.

[42] F. Rodler. Wavelet based 3D compression with fast random access for very large volume data. In *Computer Graphics and Applications, 1999. Proceedings. Seventh Pacific Conference on*, pages 108–117, 1999.

[43] J. Sanyal, S. Zhang, J. Dyer, A. Mercer, P. Amburn, and R. Moorhead. Noodles: A tool for visualization of numerical weather model ensemble uncertainty. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1421–1430, 2010.

[44] H. Shen, L. Chiang, and K. Ma. A fast volume rendering algorithm for Time-Varying fields using a Time-Space partitioning (TSP) tree. In *Visualization Conference, IEEE*, page 62, Los Alamitos, CA, USA, 1999. IEEE Computer Society.

[45] B. Sklar. *Digital Communications: Fundamentals and Applications*. Prentice-Hall, Upper Saddle River, NJ, 2 edition, 2000.

[46] R. D. Smith and P. Gent. Reference manual of the parallel ocean program (POP). Los Alamos National Laboratory report LA-UR-02-2484, Los Alamos National Laboratory, Los Alamos, NM, 2002.

[47] A. S. Szalay, G. C. Bell, H. H. Huang, A. Terzis, and A. White. Low-Power Amdahl-Balanced blades for data intensive computing. *ACM SIGOPS Operating Systems Review*, 44(1):71, 2010.

[48] D. S. Taubman. *Directionality and Scalability in Image and Video Compression*. PhD thesis, University of California, Berkeley, CA, 1994.

[49] D. S. Taubman and M. W. Marcellin. *JPEG2000: Image Compression Fundamentals, Standards, and Practice*. Kluwer, Boston, MA, 2002.

[50] A. Trott, R. Moorhead, and J. McGinley. Wavelets applied to lossless compression and progressive transmission of floating point data in 3-D curvilinear grids. In *Visualization '96. Proceedings.*, pages 385–388. IEEE, Nov. 1996.

[51] B. E. Usevitch. A tutorial on modern lossy wavelet image compression: Foundations of JPEG 2000. *IEEE Signal Process. Mag.*, 18(5):22–35, Sept. 2001.

[52] C. Wang, J. Gao, L. Li, and H. Shen. A multiresolution volume rendering framework for Large-Scale Time-Varying data visualization. In *Volume Graphics, 2005. Fourth International Workshop on*, pages 11–223, 2005.

[53] C. Wang, A. Garcia, and H. Shen. Interactive Level-of-Detail selection using Image-Based quality metric for large volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(1):122–134, 2007.

[54] R. Westermann. A multiresolution framework for volume rendering. *Proceedings of the 1994 symposium on Volume visualization*, page 51–58, 1994. ACM ID: 197963.

[55] J. Woodring, J. Ahrens, J. Figg, J. Wendelberger, S. Habib, and K. Heitmann. In-situ sampling of a large-scale particle simulation for interactive visualization and analysis. *Computer Graphics Forum*, 30(3):1151–1160, June 2011.

[56] J. Woodring and H. Shen. Multiscale time activity data exploration via temporal clustering visualization spreadsheet. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):123–137, 2009.

[57] J. W. Woods, editor. *Subband Image Coding*. Kluwer, Boston, MA, 1991.

[58] Z. Zhu, R. Machiraju, B. Fry, and R. Moorhead. Wavelet-based multiresolutional representation of computational field simulation datasets. In *Visualization '97., Proceedings*, pages 151–158. IEEE, Oct. 1997.