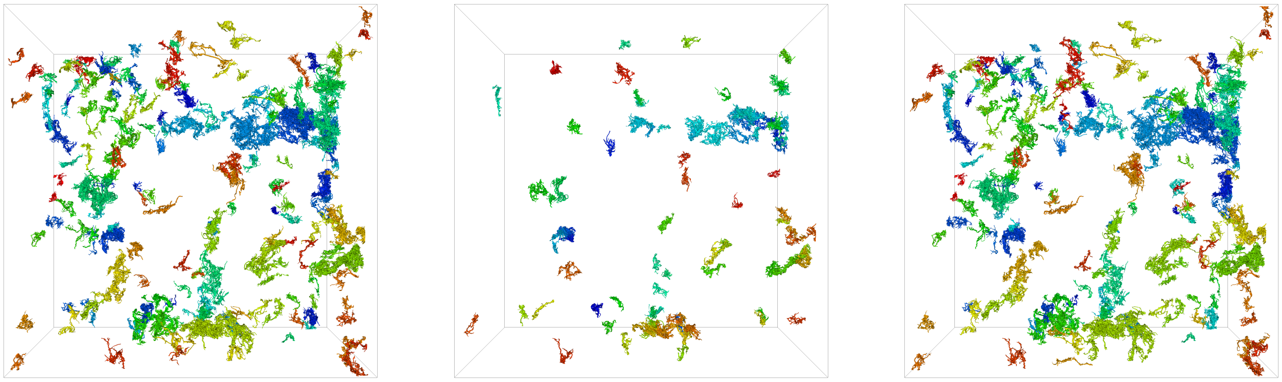
Evaluating the Efficacy of Wavelet Compression for Turbulent-Flow Data Visualization

Shaomeng Li*
Department of Computer and Information Science
University of Oregon



(a) Image from analysis of a $4,096^3$ turbulent flow data set.

(b) 8:1 compressed data using a multi-resolution technique and the Haar kernel.

(c) 128:1 compressed data using prioritized coefficients and the CDF 9/7 kernel.

Figure 1: Three renderings of a turbulent-flow data set. The left image comes from an analysis performed previously in conjunction with a simulation scientist. This analysis focused on connected components, and so the rendering assigned each component a unique color. The remaining two images repeat this analysis on wavelet compressed versions of the same data. The middle image has only an 8:1 reduction, and yet poorly captures the features of interest for this analysis. However, the wavelet settings used for the right image gives much better results: it is significantly more accurate even though it uses far fewer bytes to represent the data.

Abstract

We explore the ramifications of using wavelet compression on turbulent-flow data from scientific simulations. As upcoming I/O constraints may significantly hamper the ability of scientific simulations to write full-resolution data to disk, we feel this study enhances the understanding of exascale science with respect to potentially applying wavelets *in situ*. Our approach repeats existing analyses with wavelet-compressed data, using evaluations that are quantitatively based. The data sets we select are large, including one with a $4,096^3$ grid. Our findings show that the efficacy of wavelets vary across the analyses, and that prioritized coefficient compression is consistently superior to a multi-resolution approach, and that the biorthogonal kernels CDF 9/7 and CDF 8/4 perform better than the Haar kernel.

1 Introduction

The design of modern supercomputers is constrained by both financial and power budgets. These constraints force supercomputer architects to make difficult tradeoffs among the system components (e.g., networking, I/O, memory, and computational speed) to balance the budgets. Over the last decade, architects have devoted smaller and smaller percentages of their budgets toward I/O. While I/O bandwidth is still increasing on almost every new supercomputer, it is often not keeping pace with these supercomputers' abilities to generate data.

Simulation codes use the I/O subsystem to store their state (a time slice), for subsequent *post hoc* analysis and visualization. When I/O bandwidth slows down, relatively speaking, simulation codes often respond by saving their state less often, leading to temporal sparsity. Looking forward, the I/O subsystem may become so underpowered that the resulting temporal frequency will not be sufficient for meaningful analysis and visualization. If this happens, simulation codes will need to pursue a new strategy, where they dramatically reduce the size of their time slices before writing to disk, i.e., compression. Obviously, simulation scientists prefer to perform visualization and analysis on their raw data; an important premise of this research is that future supercomputing architectures will have I/O capabilities that require simulation scientists to accept the compromises inherent to compression.

There are several existing compression schemes which could be appropriate for scientific data (meaning mesh-based simulation data). Here, we explore wavelet compression, which is most frequently used with images, movies, and signal processing. There have been far fewer applications to scientific data, however, and this relative dearth of work is significant: as the nature of scientific data is different from that of wavelet's traditional applications, the efficacy of wavelets for this data is still not fully understood.

With this study, we applied wavelet compression to large scientific data sets, specifically turbulent-flow data sets, and measured how well the compressed forms could be used for analysis. An important goal for our effort was to quantify the efficacy — we only wanted to consider analyses where we could evaluate how effectively the wavelet representation captured the original data, and where we could state accuracies in quantitative ways. That said,

*e-mail: samuelli@cs.uoregon.edu

evaluating accuracy is frequently an application-dependent process. Lost data accuracy may be insignificant for some analyses, but very important for other analyses. Our approach for this issue was to consider two specific analyses, drawn from existing in-depth studies with simulation scientists, and then repeat them with wavelet-compressed data.

The work in this paper is an evaluation study. We felt the evaluations were needed because wavelets are frequently mentioned as a possible solution for the upcoming constraints posed by exascale computing [1, 8], and because the effectiveness of wavelets for analysis on scientific data is still not fully understood. The specific contributions of this paper are:

- Additional evidence of the usefulness of wavelet compression on scientific data sets, as well as evidence of how much compression is possible while still keeping specific analyses viable;
- Evaluation of wavelet compression that focuses on quantifying differences, which enables intuitive comparisons; and
- Exploration of multiple wavelet compression techniques, illuminating which techniques may be useful for scientific data.

This paper is organized as follows: after reviewing related work in Section 2, Section 3 describes options for applying wavelet compression, Section 4 gives an overview of our experiment methodology, Section 5 and 6 summarize the analysis results on wavelet-compressed data, Section 7 discusses savings in I/O, and Section 8 presents I/O performance measurements in practice.

2 Related Work

We survey related work in wavelet compression applied to visualization (Section 2.1) and in analysis of turbulent-flow data (Section 2.2).

2.1 Wavelet Compression in Visualization

Wavelet compression is a common method for reducing the memory footprint of data, and has been applied widely in visualization approaches. When reconstructing slices of a CT data set, two-dimensional wavelet transforms have been proven to provide high compression rates with fast decoding for performing random access of voxels [14, 19]. When used on three-dimensional volume data sets, such as hydrodynamic simulations or global ocean models, wavelet compression is effective for both data reduction and visualization with different levels of detail [2, 22]. Wavelet compression also brings new possibilities for real-time analysis on large scale data sets, for example, both large mesh reconstruction and animated volume data decompression are achievable in real-time on commodity hardware using wavelet compression [10, 12].

The work most closely related to our project aims not only to exploit the power of wavelet compression, but to also help users understand the precision lost from the data compression process. The latter is essential for making good decisions on the tradeoffs between compression and information loss. Wong et. al. proposed an energy-based model to analyze the authenticity of orthogonal wavelet compressed volume data [23]. Woodring et. al. and Ma et. al. further introduced visualization techniques to encode the amount of variance in the same volume space, providing the ability to examine local information loss at points of interest [16, 24]. Exact error bounds can be found for each data compression level [24], allowing domain scientists to get a precision guarantee when analyzing compressed data sets.

In contrast to previous work, the focus of this study is in understanding the effects of two widely used wavelet kernels on specific data analysis procedures. The goal is to understand the efficacy of wavelets in a variety of configurations and evaluate the trade off between data integrity and visual analytics tasks on turbulent-flow data sets.

2.2 Turbulent-Flow Data Analysis

Our study investigates the use of wavelet compressed data with two established analyses of turbulent-flow data. We do not aim to advance the analyses themselves, but instead evaluate the effects of wavelet kernel and compression ratio on the resulting analysis compared to uncompressed data.

The first analysis finds critical structures in massive time-varying turbulent-flow simulations [9]. Clusters of enstrophy are found and grouped into connected components, using a distributed-memory parallel algorithm [13]. Characteristics of the components are then tracked over time, providing a time-varying profile of the enstrophy structure population. A performance study showed that the analysis was parallelizable and scalable, which is essential for analyzing large data sets.

The second analysis is by Gruchalla et. al., which studies the structural and statistical analysis of turbulent flows by focusing on local structures and their dynamics [11]. In their analysis, identified critical structures were first grouped into three populations based on their dynamic properties, which were then profiled based on vorticity and helicity. More local dynamic properties were further studied, including detailed dynamics of single structures as well as aggregated group statistics.

3 Data Compression with Wavelets

Let $x[n]$ be a one-dimensional data array and $u_k[n]$ be a set of K basis functions. Then a wavelet transform expands $x[n]$ as:

$$x[n] = \sum_{k=0}^{K-1} a_k \cdot u_k[n]. \tag{1}$$

The set of basis function consists of two subsets, a *scaling* function set Φ and a *wavelet* function set Ψ. A basis function $u_k[n]$ belongs to either one of these two subsets. Each coefficient a_k measures the similarity between the corresponding basis function $u_k[n]$ and the data array $x[n]$. This transform itself is lossless and $x[n]$ can be recovered by calculating the above expansion using all K coefficients. However, wavelets are frequently used in a lossy manner [7, 17, 21]: when reconstructing $x[n]$, an approximation can be calculated by using a subset of the K coefficients, specifically the first $\widetilde{K}$:

$$\tilde{x}[n] = \sum_{k=0}^{\widetilde{K}-1} a_k \cdot u_k[n]. \tag{2}$$

Data compression is thus achieved by storing fewer coefficients and reconstructing $\tilde{x}[n]$ using only those fewer coefficients. The compression ratio is derived from the fraction of coefficients used to reconstruct the data set. For instance, to achieve the 8:1 compression ratio, one eighth of the total number of coefficients would be used.

3.1 Wavelet Kernels

We consider two types of wavelet kernels: Haar and biorthogonal. Biorthogonal kernels [6] can have multiple forms, and in this study we consider two of those forms: CDF 9/7 and CDF 8/4.

3.1.1 Haar Kernel

The Haar kernel is one of the most basic wavelet kernels. It uses a series of “square-shaped” functions for its basis function set.

The Haar kernel is one type of orthogonal kernels, i.e., wavelet kernels that require any pair of its basis functions to have a zero inner product. While the Haar kernel is simple and easy to use, the image processing community has favored biorthogonal kernels, since they can compact image data into fewer wavelet coefficients.

3.1.2 Biorthogonal Kernels

Biorthogonal kernels relax the constraint of orthogonal kernels that all basis functions must be orthogonal. A biorthogonal kernel only requires the scaling function set Φ to be orthogonal with the dual of the wavelet function set Ψ, and the wavelet function set Ψ to be orthogonal with the dual of the scaling function set Φ. This additional freedom increases the flexibility to design and optimize the basis functions. Biorthogonal kernels always use the same number of coefficients as the original data array. The Haar kernel also has this property, although orthogonal wavelets, with the exception of Haar, are expansive and their transforms result in more coefficients than samples from the original array.

The two biorthogonal kernels in our experiments are from the popular Cohen-Daubechies-Feauveau family of wavelets: CDF 9/7 and CDF 8/4 [6]. Each member of the CDF wavelet family has its own filter size, which is indicated by its suffix, e.g., 9/7 or 8/4. CDF 9/7 has been widely embraced, and is often used in image compression (e.g. JPEG 2000 [20]). However, CDF 8/4 has slightly less computational complexity.

3.2 Compression Using Wavelets

There are different strategies for laying out wavelet coefficients in order to achieve compression. With our study, we evaluated a multi-resolution scheme and prioritized coefficient storage scheme.

3.2.1 Multi-resolution

With a multi-resolution approach, the wavelet transform is applied in a recursive fashion: each application of the transform divides the signal — the data array in our case — into "approximation" (i.e., scaling) and "detail" (i.e., wavelet) coefficients. The approximation coefficients provide a coarsened representation of the signal, while the detail coefficients contain the information that is missing from the approximation coefficients. The transform is repeatedly applied to the approximation coefficients, and each pass creates a coarser and coarser approximation. In the case of the Haar wavelet, the approximation is a simple unweighted averaging of neighboring samples. Finally, note that coefficients stored in this manner have implicit addressing based on their spatial locations and resolution levels, and thus do not introduce additional storage costs.

In effect, the multi-resolution property offers a pyramid representation. This pyramid representation is strictly limited to power-of-two reductions along each axis. Thus, in the case of three-dimensional regular grids, the available reduction ratios are of the form 8^N:1, i.e., 8:1, 64:1, 512:1, etc. Techniques such as this one are widely used by the visualization community, although the wavelet approach differs from approaches such as mipmapping, as it does not require additional storage for coarse versions, and such as space-filling curves, as these curves create coarse representations where the value for a region comes from a single data point, rather than average of all points in that region.

3.2.2 Prioritized Coefficients

When reconstructing the original data from the wavelet expansion given in equation 1, coefficients are of different importance, i.e., coefficients representing the more rapidly changing parts of the data array are more important than coefficients representing the more self-similar parts of the data array. The prioritized-coefficient technique makes use of this property by storing all coefficients in order of their importance. This practice collects the most informative wavelet coefficients into a relatively small group stored together. The original data array can still be reconstructed, however, this scheme augments the coefficients with information about the coefficients' spatial location in the volume. This extra information requires additional storage overhead, and the cost for this bookkeeping is discussed in Section 7.

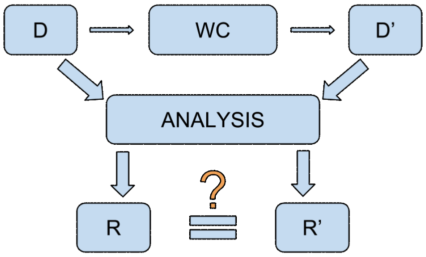


Figure 2: Our experiment methodology. D denotes the raw data, WC denotes an operator that applies wavelet compression to the data (with VAPOR), D' denotes the newly created wavelet data, $ANALYSIS$ represents the analysis performed (with VisIt or VAPOR), and R and R' represent the results of the analysis for D and D' respectively. Our study then evaluated the difference between R and R'.

4 EXPERIMENT OVERVIEW

Our experiments were designed to quantitatively evaluate the efficacy of wavelet compression on the analysis of turbulent flow data. We evaluated many configurations, varying over kernels, coefficient storage schemes, and compression levels. Section 4.1 describes our experiment methodology for a generic configuration, and Section 4.2 describes the different configurations we studied.

4.1 Experiment Methodology

Our experiment methodology, illustrated in Figure 2, was as follows:

- We began with turbulent flow data in its raw form.
- We applied wavelet compression to the raw data. This step was performed using the VAPOR software package [4, 5].
- We applied an analysis routine to both data sets (raw form and wavelet compressed). The analysis was done using either VisIt [3] or VAPOR, depending on the original analysis performed.
- We evaluated the difference between the resulting analyses.

4.2 Configurations Studied

We varied three parameters in our experiments: wavelet kernel, wavelet compression strategy, and compression ratio. We first considered wavelet compression strategy (i.e., how does multi-resolution compare with prioritized coefficients) using only the Haar kernel because our implementation of biortogonal kernels only supported priortized coefficients. We then looked at the effects of the wavelet kernel (i.e., Haar vs two biorthogonal kernels: CDF 9/7 and CDF 8/4), all within the context of a prioritized coefficient compression strategy. We compared seven different compression ratios: 8:1, 16:1, 32:1, 64:1, 128:1, 256:1, and 512:1. However, using the multi-resolution compression strategy restricted possible compression ratios to only 8:1, 64:1, and 512:1. Finally, in the evaluation of wavelet applied in the local dynamics analysis, we involved all available kernel and compression strategy combinations, and studied them using five compression ratios. Figure 3 shows the studied configurations. The cross sign denotes wavelet settings used in evaluating wavelet compression strategies; the circle denotes wavelet settings used in evaluating wavelet kernels; and the square denotes wavelet settings used in evaluating local dynamics.

4.3 Analysis Performed

We performed two distinct visual analysis routines on two scientific data sets, both of which come from turbulent flow simulations in a cubic space. For simplicity, we denote the first data set as

Kernel	Compression Strategy	8:1	16:1	32:1	64:1	128:1	256:1	512:1
Haar	Multi-res	✕ □			✕ □			✕ □
	Prioritized	✕ ○ □	○	○	✕ ○ □	○ □	○ □	✕ ○ □
CDF 9/7	Prioritized	○ □	○	○	○ □	○ □	○ □	○ □
CDF 8/4	Prioritized	○ □	○	○	○ □	○ □	○ □	○ □

Figure 3: Wavelet configurations studied. Cross signs show compression settings examined when comparing multi-resolution vs prioritized coefficients (Section 5.3.1). Circles show the compression settings when comparing Haar vs biorthognal kernals (Section 5.3.2). Squares denote compression settings used in evaluating the wavelet in local dynamics analysis (Section 6.3).

	Mesh Resolution	Number of Time Slices	Number of Scalar Fields	Number of Vector Fields
DS1	$4,096^3$	13	1	0
DS2	$1,024^3$	1	3	1

Table 1: Properties of our two scientific data sets.

DS1, and the second as DS2. Table 1 describes their key properties. Both analyses, described earlier in Section 2.2, came from established studies that included domain scientists [9, 11]. Though each analysis is performed multiple times on data sets with different wavelet settings (raw form, Haar+multi-resolution, CDF 9/7+prioritized, etc.), the underlying analysis itself is the same each time.

5 Critical Structure Identification

This section is divided into three sub-sections: overview of the analysis task (5.1), description of evaluation methodology (5.2), and results (5.3).

5.1 Analysis Overview

This analysis identifies critical structures using various settings for each type of wavelet compression method, comparing each to the baseline, or uncompressed, data. It operates on the "enstrophy" scalar field of DS1. A critical structure is defined as a region with significantly higher enstrophy values than the area surrounding it. Identification of these critical structures requires two steps. The first step isolates regions with enstrophy values higher than some α, a fixed value provided by the domain scientists. DS1 contains millions of these high-enstrophy regions. The second step eliminates structures with a volume smaller than some threshold β, again a fixed value provided by the domain scientists. In DS1, this process results in hundreds of critical structures for further study. Figure 1a shows the results from this analysis using the first time slice of DS1.

While sensitivity to α and β is one potential concern about this analysis, a bigger concern is that of errors resulting from compression. If a compressed version of the enstrophy field breaks a component (or joins two disjoint components), then the result may put that component below β (or the joined component above β). Our goal with this research is not to modify the established analysis method to perform more robustly — although that is a worthy goal even outside the context of compression — but rather to identify how well wavelet compression can maintain the integrity of the data for such an analysis.

5.2 Evaluation Methodology

The baseline analysis, i.e., performing the identification task on the raw data, yields some number of critical structures. Similarly, the analysis on the wavelet-compressed data also yields some number of critical structures. In the ideal case, the number of critical structures for both would be the same, and each critical structure in the baseline analysis would have a corresponding structure in the same location in the compressed data. However, the critical structures do not align in this ideal way in practice.

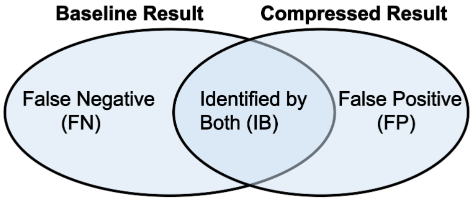


Figure 4: A Venn diagram describing the three types of structures: false negative, false positive, and correct (i.e., identified by both). False negatives refer to structures identified in the baseline but not in the compressed data. False positive structures identified in the compressed data but not in the baseline.

There are two types of error that can occur. First, a critical structure can appear in the compressed data that does not appear in the raw data. We refer to this type of error as a *false positive*. Second, a critical structure can fail to appear in the compressed data, even though it does appear in the raw data. We refer to this type of error as a *false negative*. Figure 4 illustrates this concept as a Venn diagram.

To provide a better comparison among all compression ratios, we consider the proportion of error among the critical structures, rather than absolute numbers. Formally, for a compressed data set being compared to the raw data, let FN be the number of false negatives, FP be the number of false positives, and IB be the number of critical structures common to both. Then we define:

$$proportion\ of\ FN = \frac{FN}{FN + IB} \tag{3}$$

$$proportion\ of\ FP = \frac{FP}{FP + IB} \tag{4}$$

Overall, the lower these two proportions are, the better a compression setting matches the baseline.

We only placed a critical structure in IB, i.e., identifying it as the same between the raw and compressed data, if it passed our proximity test. This test compared the bounding boxes of all structures in the baseline with the bounding boxes of all structures in the compressed data. For each pair of baseline-and-compressed structures, the overlap was measured. The overlap was calculated so that structures with similar sizes and similar spatial extents would have high values. Specifically, if V was the volume of intersection between the two, V_B was the volume of the baseline structure, and V_C was the volume of the compressed structure, then their overlap was scored as $V^2/(V_B \times V_C)$. A perfect overlap would score 1, and no overlap would score 0. A baseline-compressed structure pair was then identified as the "same" if, for a baseline structure B and a compressed structure C, then B's best match (i.e., highest score) was C, and C's best match was B. This meant that large baseline structures that got split during compression would contribute false positives (as only one compressed structure would match, but one would find no match), and separate baseline structures that got combined during compression would contribute false negatives (as only one baseline structure would match the combined structure).

5.3 Results

We break our evaluation into two steps. First, in Section 5.3.1, we study the impact of coefficient storage schemes. Second, in Section 5.3.2, we explore the impact of wavelet kernel choice.

5.3.1 Wavelet Compression Strategy

We evaluate the two coefficient storage schemes — prioritized coefficients and multi-resolution — with three compression ratios: 8:1, 64:1, and 512:1, with each test using the Haar kernel. This results in six different wavelet settings. Figure 5 shows renderings from our analysis using each of the six settings on the first time slice of DS1.

Figure 6 compares the false positive and false negative proportions between prioritized coefficients and a multi-resolution technique. At the 8:1 compression ratio, results with prioritized coefficients have approximately 10% error for both metrics. This likely could be acceptable to the domain scientist in practical use. Interestingly, the other wavelet settings all fall outside the 10% range, making them likely unacceptable for this analysis. That said, prioritized coefficients clearly outperform the multi-resolution approach: prioritized coefficients with 512:1 compression are more accurate than the multi-resolution approach with 8:1 compression. As can be seen in Figure 6, the multi-resolution approach fails to identify many of the structures that exist in the baseline, leading to a high false negative rate.

5.3.2 Wavelet Kernel Choice

We then expanded our kernel choices to include the biorthogonal kernels CDF 9/7 and CDF 8/4. This meant there were a total of three kernels, since we still considered the Haar kernel. We no longer considered a multi-resolution approach, and this allowed us to consider more compression ratios. We studied five: 8:1, 64:1, 128:1, 256:1, and 512:1. Thus, the resulting number of experiments was fifteen. (Three of these fifteen are repeated from Section 5.3.1: Haar+8:1, Haar+64:1, and Haar+512:1.) Figure 7 shows the difference among three wavelet kernels using the 256:1 compression ratio.

We plotted the false positive and false negative proportions for these three wavelet kernels in Figure 8 . Figure 8a shows that the CDF 9/7 kernel has the lowest false positive rates at compression ratios 8:1, 64:1, 128:1, and 256:1 by a clear margin. Figure 8b shows that the CDF 9/7 and CDF 8/4 have similar false negative rates, but they are both lower than the Haar kernel. Summing up, these results indicate that the CDF 9/7 is the best choice among these three wavelet kernels for this analysis. However, once again, the false negative and false positive proportions are likely too high for real-world analysis for any compression ratio but 8:1.

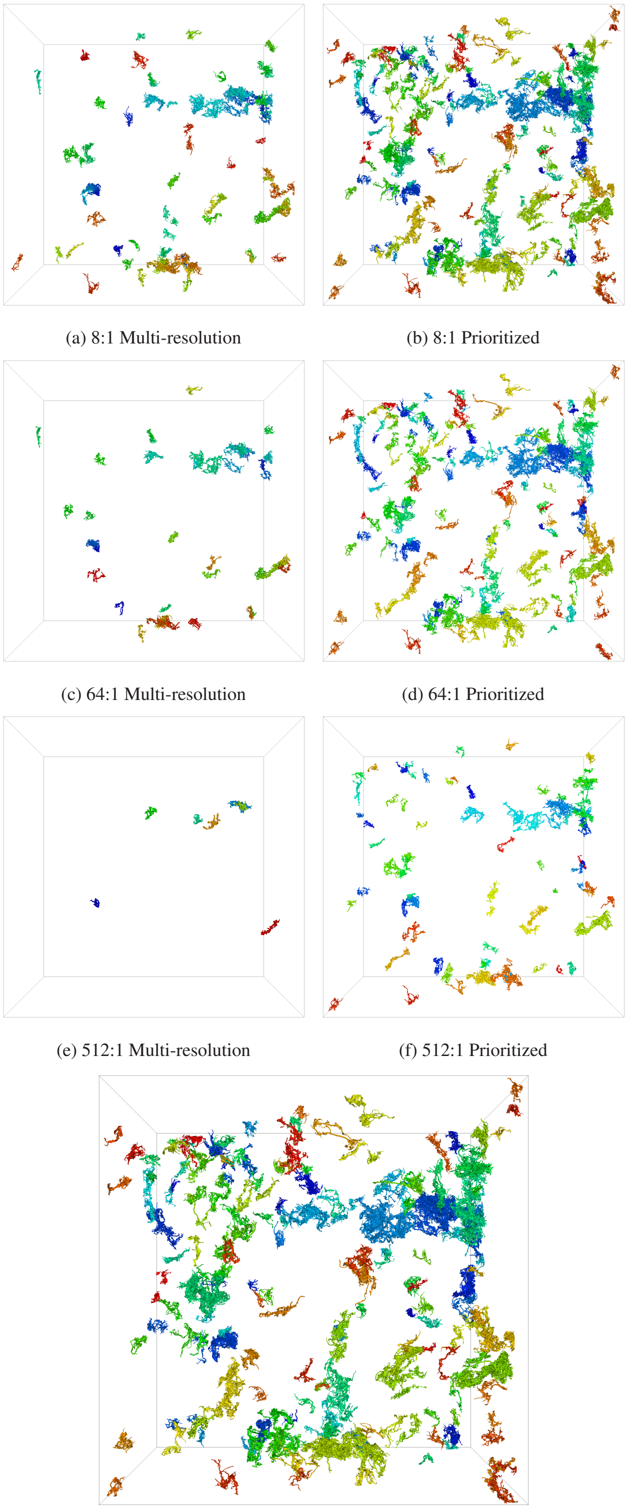
6 Local Dynamics Analysis

This section is again divided into three sub-sections: overview of the analysis task (6.1), description of evaluation methodology (6.2), and results (6.3).

6.1 Analysis Overview

The second analysis studies dynamic properties of local structures, comparing the results of analysis on compressed data to the raw data baseline. The analysis operates on the three data fields of the DS2 data set: vector field “velocity,” and scalar fields “enstrophy”, “vorticity”, and “helicity”. Local structures are first identified following steps similar to those described in Section 5.1, and then an analysis on two distinct types of local dynamics is performed. The two types of local dynamics differ in their helicity distribution – one has the majority of its helicity values close to zero, and the other has the majority of values close to one. A radial-enstrophy profile captured the local dynamics, since it was able to provide flow information from inside and immediately outside of the structure.

While the analysis described in Section 5.1 studied the global behavior of structures, this analysis focused on individual structures. Specifically, one type of structure, which we refer to as S1, twists around a core (and has low mean helicity). Another type of structure, which we refer to as S2, writhes around a core (and has


(a) 8:1 Multi-resolution

(b) 8:1 Prioritized

(c) 64:1 Multi-resolution

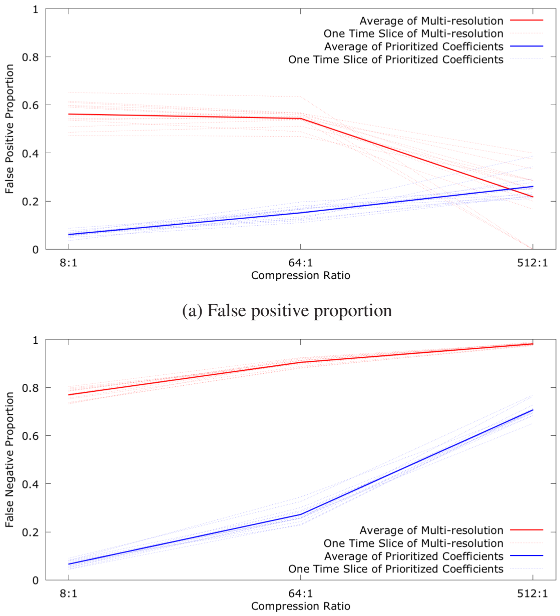
(d) 64:1 Prioritized

(e) 512:1 Multi-resolution

(f) 512:1 Prioritized

(g) Baseline analysis on raw data

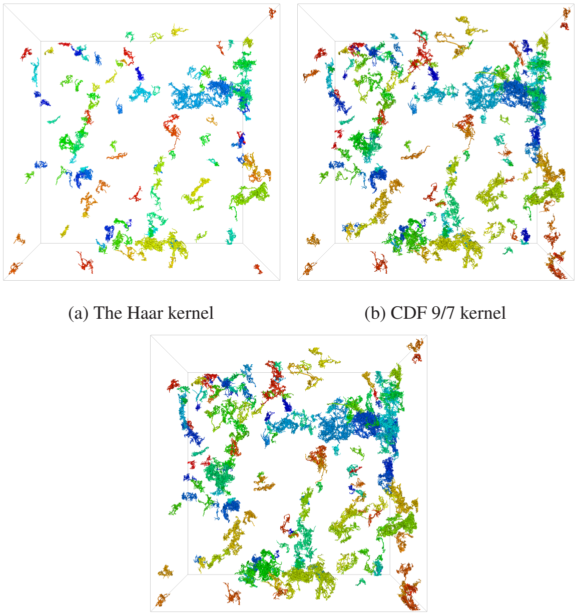
Figure 5: Screenshots from our critical structure identification task using multi-resolution coefficients (left column) and prioritized coefficients (right column), with the baseline result on the raw data at the bottom. Each identified structure has a unique color. Preliminary visual inspection shows that results using prioritized coefficients produce a significantly higher fidelity than results using a multi-resolution technique: results using prioritized coefficients not only retain more critical structures from the baseline, but also preserve more shape details.



(a) False positive proportion

(b) False negative proportion

Figure 6: False positive (a) and false negative (b) proportions for the multi-resolution (red) and prioritized (blue) coefficients at three compression levels. Each dotted line connects results from one time slice of our thirteen-time-step turbulent-flow data. The bold lines represent the average from all thirteen lines of that setting.

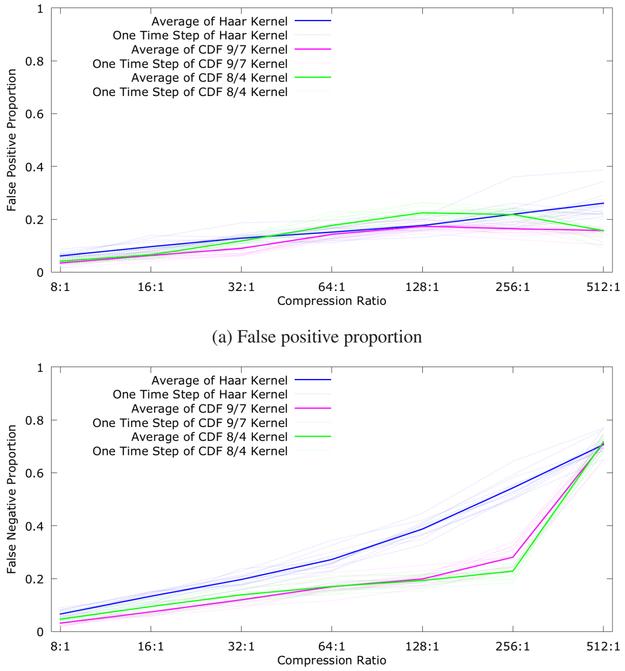


(a) The Haar kernel

(b) CDF 9/7 kernel

(c) CDF 8/4 kernel

Figure 7: Screenshots from our critical structure identification analysis using three different wavelet kernels on the first time slice of DS1. Each is compressed by a ratio of 256:1 and using prioritized coefficients. Preliminary visual inspection shows they all successfully capture many structures, but CDF 9/7 and CDF 8/4 manage to keep more details than Haar.



(a) False positive proportion

(b) False negative proportion

Figure 8: False positive (a) and false negative (b) proportion for three wavelet kernels: Haar (blue), CDF 9/7 (purple), and CDF 8/4 (green). All experiments use prioritized coefficients. Each dotted line represents results from one time slice of our thirteen-time-slice turbulent-flow data set. The bold lines represent the average value from all thirteen lines for a setting.

high mean helicity). Figure 9 shows renderings of S1 and S2, with the local dynamics illustrated by streamlines seeded in the velocity field.

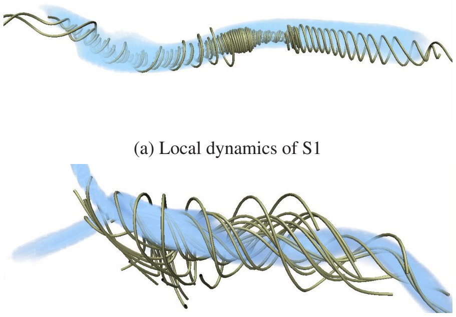
The nominal radii of each structure was approximately five cells (inside a 1024^3 grid), so the radial-enstrophy profile uses a radius of ten cells to enclose the space surrounding the structure. The analysis samples the enstrophy values at each radius, using fifteen planes orthogonal to the major axis of each structure. The final radial-enstrophy profile was an average of these fifteen samples.

6.2 Evaluation Methodology

Our evaluation process began by identifying the two structures, S1 and S2, in the wavelet-compressed versions of the data and then calculating their radial-enstrophy profiles. We then evaluated how well the compressed data preserved local dynamics by quantifying the difference between their radial-enstrophy profiles. Ideally, the profile produced using compressed data would be the same as the baseline, i.e., the radial-enstrophy plots produced using compressed data would overlap with the baseline plots. However, in practice, there were differences between the two profile lines.

To measure the difference between two profiles, we used a normalized root mean square error (NRMSE) metric. Specifically, given the baseline radial-enstrophy series $E[r]$ $(0 \leqslant r < N)$ and the compressed radial-enstrophy series $\widetilde{E}[r]$ $(0 \leqslant r < N)$, we define:

$$RMSE = \sqrt{\frac{\sum_{r=0}^{N-1} (E[r] - \widetilde{E}[r])^2}{N}} \qquad (5)$$



(a) Local dynamics of S1

(b) Local dynamics of S2

Figure 9: Visualizations of local dynamics. Streamlines are seeded in the velocity field in areas of high vorticity, showing the distinct local dynamics of the two structures. On the top (and referred to as S1), streamlines twist tightly around the core; these types of structures were identified by low mean helicity. On the bottom (and referred to as S2), streamlines writhe around the core; these types of structures were identified by high mean helicity.

and RMSE is then normalized to be:

$$NRMSE = \frac{RMSE}{E_{max} - E_{min}} \tag{6}$$

If $E[r]$ and $\widetilde{E}[r]$ are exactly the same, then NRMSE would have a value of zero. However, in the worst case, if one array had nearly all values equal E_{max} and the other had nearly all values equal E_{min}, then the NRMSE would approach a value of one.

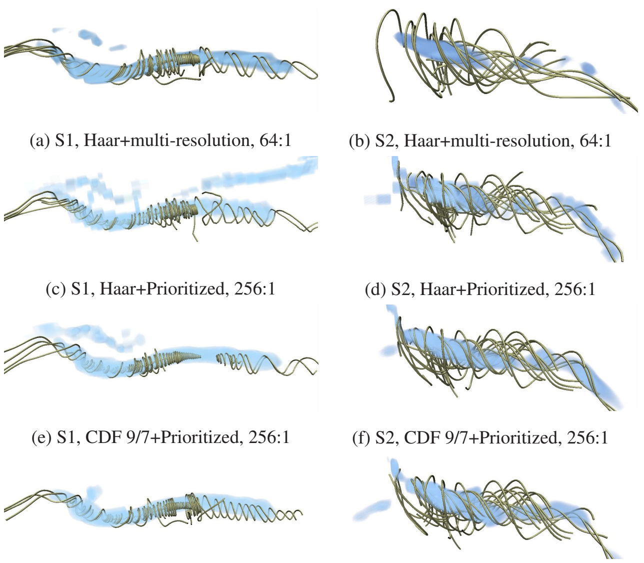
6.3 Results

The two populations of vortical structures could be identified and distinguished at high levels of compression. Figure 10 displays screenshots of the local dynamics from compressed data sets using different wavelet settings. It shows that using the Haar kernel with multi-resolution coefficients differentiates the two types of local dynamics at a compression ratio 64:1, while all three wavelet settings using prioritized coefficients differentiate the two types at a compression ratio 256:1.

Figure 11 shows the NRMSE of the radial-enstrophy profile for the two structures. Both NRMSE charts show that three wavelet settings using prioritized coefficients are significantly better than Haar+multi-resolution. This result is consistent with our findings in Section 5.3.1. In addition, when using prioritized coefficients, the two biorthogonal kernels (CDF 9/7 and CDF 8/4) always perform better than the Haar kernel at all compression ratios other than 512:1. We notice that for both structures, S1 and S2, the 128:1 compressed data using biorthogonal kernels and prioritized coefficients yields an NRMSE less than 3%. This indicates that biorthogonal kernels and prioritized coefficients at compression ratio 128:1 are likely to be acceptable in many real-world applications.

7 Storage Considerations

Wavelet compression is achieved by reconstructing the data from a subset of all wavelet coefficients (Section 3). Ideally the resulting file size is strictly proportional to the applied compression ratio, e.g., an 8:1 compressed file has one-eighth the size of the raw data. Table 2 shows data sizes and compression ratios for the first time slice of the turbulent flow data (a ∼256 GB data set) using the Haar kernel with multi-resolution and prioritized coefficient schemes. (We found that the prioritized coefficient scheme introduces exactly the same storage overhead with different wavelet



(a) S1, Haar+multi-resolution, 64:1

(b) S2, Haar+multi-resolution, 64:1

(c) S1, Haar+Prioritized, 256:1

(d) S2, Haar+Prioritized, 256:1

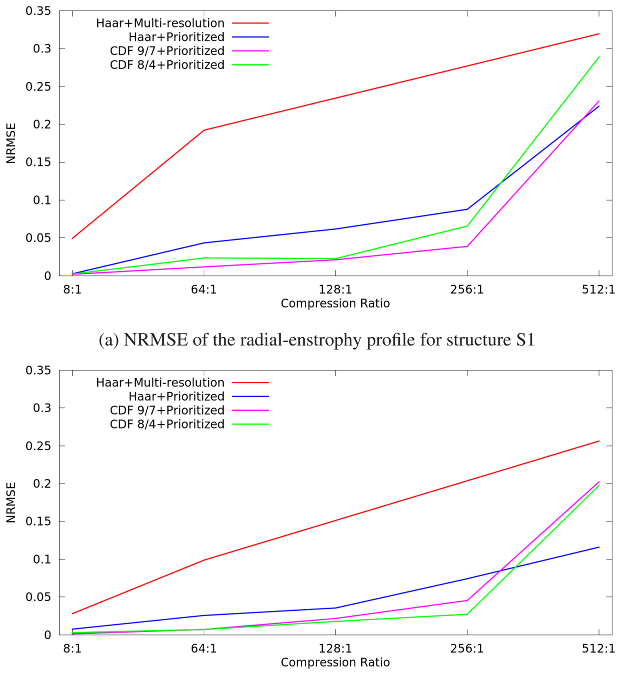
(e) S1, CDF 9/7+Prioritized, 256:1

(f) S2, CDF 9/7+Prioritized, 256:1

(g) S1, CDF 8/4+Prioritized, 256:1

(h) S2, CDF 8/4+Prioritized, 256:1

Figure 10: Renderings of local dynamics of two structures: S1 (left) and S2 (right). Visual inspection shows that the Haar+multi-resolution wavelet setting (a) and (b) at compression ratio 64:1 yields comparable results as other wavelet settings using prioritized coefficients at compression ratio 256:1.



(a) NRMSE of the radial-enstrophy profile for structure S1

(b) NRMSE of the radial-enstrophy profile for structure S2

Figure 11: NRMSE of the radial-enstrophy profile for the two studied structures. Lines of different colors represent different wavelet settings. The red ones represent the setting of Haar+multi-resolution, which only supports compression ratios at 8:1, 64:1, and 512:1 (Section 3.2.1). In this line chart we connect the data points of 64:1 and 512:1 using a straight line. However, it does not indicate any valid NRMSE values at 128:1 and 256:1 positions.

	Ideal	Multi-res	Prioritized
1:1	256.0	256.0179 (0.99:1)	274.1094 (0.93:1)
8:1	32.0	32.0022 (7.99:1)	50.1094 (5.11:1)
16:1	16.0	N/A	25.0938 (10.20:1)
32:1	8.0	N/A	12.5781 (20.35:1)
64:1	4.0	4.0003 (63.99:1)	6.3125 (40.55:1)
128:1	2.0	N/A	3.1719 (80.71:1)
256:1	1.0	N/A	1.5938 (160.62:1)
512:1	0.5	0.5000 (511.97:1)	0.7969 (321.24:1)

Table 2: Ideal and actual on-disk sizes for compression in different ratios. Our test data set has an original file size of 256 gigabytes (GB), and is compressed using the Haar kernel with multi-resolution and prioritized coefficient schemes. The actual achieved compression ratios (including implementation overhead) are shown in parentheses. The four "N/A" cells are ratios that the multi-resolution scheme does not support.

kernels, thus we only represent results from the Haar kernel here.) The multi-resolution scheme achieves ratios close to ideal, with the slight differences explained by optional metadata stored in the VAPOR file format. However, the prioritized compression rates we achieve are significantly less than ideal. While the wavelet coefficients generated by the multi-resolution scheme may simply be stored in the order output by the forward wavelet transform - thus making their addressing implicit - the prioritized coefficients are ordered based on their information content, and this re-ordering requires us to keep track of their addresses. State of the art compression encoders, such as SPIHT [15] and SPECK [18], are able to cleverly encode wavelet coefficients in a manner that introduces no storage overheard. However, these schemes require that the coefficients are first byte-scaled to integers, introducing information loss. A requirement of the VAPOR storage format is the ability to reconstruct the original data perfectly (up to floating point round-off). Hence, VAPOR uses a less efficient encoding scheme, but one that avoids errors introduced by quantization.

8 I/O Performance Measurements

We measure I/O performance for various wavelet compression settings in this section. Here we consider specifically the time to read data from disk, and the time to possibly decompress the data. In a computing environment without data compression, reading data from disk dominates the total data preparation time with little or no computational overhead. However, reading wavelet compressed data sets requires time to perform the inverse wavelet transform needed to reconstruct the data set (thus decompress the data), which is no longer negligible. We present our I/O performance measurements with time breakdowns to reflect the two major steps: 1) data read from disk; and 2) inverse wavelet transform. We describe our experiment setup in Section 8.1, and present results in Section 8.2.

8.1 Experiment Setup

The test environment is a commodity Apple iMac desktop computer with a quad-core Intel Core i5 processor, a 7,200 RPM hard disk, and 16GB memory. The experiment task was solely for the purpose of performance measurement: it reads in the wavelet compressed data, performs inverse wavelet transform, and writes it back to disk in raw format. We perform this task 10 times on each available wavelet settings (Haar+multi-resolution, CDF 9/7+prioritized, etc.) and compression ratios (8:1, 64:1, etc.). We report the final results as average time used for reading data from disk and performing inverse wavelet transform.

The data set we experiment on is a portion of the entire $4,096^3$ turbulent-flow data set, which measures 4GB in raw format. We notice that cache mechanism of the operating system also have an impact on the measured disk-read time. To minimize this impact, we perform a separate disk-read task that reads in 16GB of irrelevant data first before each time we perform our intentional measurement.

8.2 Measurement Results

We present our measurement results in Table 3. It shows that the achieved read time speed-ups (in parentheses) keep in line with the applied compression ratios. We also notice that the inverse wavelet transform takes a majority of total data access time when the compression ratio goes higher. This is less likely to keep being a bottleneck in a supercomputing setting, since the relative speed of CPUs in supercomputers is much faster.

The inverse wavelet transform time varies when the multi-resolution scheme is in use, although it almost keeps constant across different compression ratios with the prioritized-coefficient scheme. This property also indicates an upper bound of how much performance one could possibly gain from prioritized-coefficient-based wavelet compressions for a particular computer system.

9 Conclusion

We performed an evaluation study looking at the efficacy of wavelet compression for very large turbulent-flow data sets. Our approach took two existing analyses and repeated them using multiple compressed data sets, varying over compression scheme (multi-resolution versus prioritized) and kernel (Haar, CDF 9/7, and CDF 8/4). The findings showed that, for our applications, prioritized coefficients are significantly better at storing the essence of the data than multi-resolution techniques. Further, the CDF 9/7 and CDF 8/4 kernels better encoded data than the Haar kernel.

Interestingly, the local dynamics analysis showed good results up to 256:1 compression, but the critical structure identification analysis showed error much more quickly, and likely was unusable starting at 64:1 compression. This variation reinforces the importance of keeping in mind the analysis to be performed.

In terms of future work, we would like to apply wavelet compression in an *in situ* setting. Ultimately, we would like to demonstrate that wavelet compression is a viable option for exascale simulation scientists.

Collaborators

My adviser, Professor Hank Childs, directed this project. This work was also in collaboration with Kenny Gruchalla from the National Renewable Energy Laboratory, Kristi Potter from University of Oregon, and John Clyne from the University Corporation for Atmospheric Research. Finally, the writing of this DRP report is heavily based on a paper submission to the conference of PacificVis 2015. Professor Hank Childs was significantly involved in the writing of that paper submission.

References

[1] S. Ahern et al. Scientific Discovery at the Exascale: Report for the DOE ASCR Workshop on Exascale Data Management, Analysis, and Visualization, July 2011.

[2] M. Bertram, M. A. Duchaineau, B. Hamann, and K. I. Joy. Bicubic subdivision-surface wavelets for large-scale isosurface representation and visualization. In *Proceedings of the conference on Visualization'00*, pages 389–396. IEEE Computer Society Press, 2000.

[3] H. Childs, E. Brugger, B. Whitlock, J. Meredith, S. Ahern, D. Pugmire, K. Biagas, M. Miller, C. Harrison, G. H. Weber, H. Krishnan, T. Fogal, A. Sanderson, C. Garth, E. W. Bethel, D. Camp, O. Rübel, M. Durant, J. M. Favre, and P. Navrátil. VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data. In *High Performance Visualization–Enabling Extreme-Scale Scientific Insight*, pages 357–372. Oct 2012.

	Haar+Multi-resolution		Haar+Prioritized		CDF 9/7+Prioritized		CDF 8/4+Prioritized	
	Read	XForm	Read	XForm	Read	XForm	Read	XForm
1:1	23.0751 (1.14:1)	7.5342	29.6650 (0.88:1)	8.3280	25.2008 (1.04:1)	15.0866	27.5898 (0.95:1)	12.5682
8:1	5.4794 (4.79:1)	1.7891	5.6227 (4.66:1)	7.7527	4.6987 (5.58:1)	14.3582	4.8919 (5.36:1)	11.8683
16:1	N/A		1.9942 (13.15:1)	7.7008	1.6484 (15.91:1)	14.3104	1.6561 (15.84:1)	11.8645
32:1	N/A		0.8966 (29.25:1)	7.7438	0.8870 (29.57:1)	14.2981	0.8856 (29.61:1)	11.8884
64:1	1.2044 (21.78:1)	0.3556	0.4979 (52.68:1)	7.6804	0.4934 (53.16:1)	14.3296	0.4962 (52.85:1)	11.9784
128:1	N/A		0.2948 (88.96:1)	7.7132	0.2907 (90.23:1)	14.3491	0.2938 (89.27:1)	12.1512
256:1	N/A		0.1807 (145.11:1)	7.7460	0.1797 (145.93:1)	14.4212	0.1801 (145.57:1)	11.8204
512:1	0.2004 (130.87:1)	0.0000	0.1167 (224.65:1)	7.8913	0.1169 (224.41;1)	14.6089	0.1165 (225.05:1)	11.8743

Table 3: I/O Performance measurements for four wavelet settings and eight compression ratios. There are two columns of time measurement (both in seconds) for each wavelet setting: the left column shows read time from disk, and the right column shows inverse wavelet transform time. We also show the achieved speed-ups for read time in parentheses, which are calculated against the read time for raw data, 26.2273s. The four "N/A" cells are compression ratios that the multi-resolution scheme does not support.

[4] 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, 2007.

[5] J. Clyne and M. Rast. A prototype discovery environment for analyzing and visualizing terascale turbulent fluid flow simulations. In *Electronic Imaging 2005*, pages 284–294. International Society for Optics and Photonics, 2005.

[6] A. Cohen, I. Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, 45(5):485–560, 1992.

[7] I. Daubechies et al. *Ten lectures on wavelets*, volume 61. SIAM, 1992.

[8] J. Dongarra, P. Beckman, et al. The International Exascale Software Roadmap. *International Journal of High Performance Computer Applications*, 25(1), 2011.

[9] K. P. Gaither, H. Childs, K. W. Schulz, C. Harrison, W. Barth, D. Donzis, and P.-K. Yeung. Visual Analytics for Finding Critical Structures in Massive Time-Varying Turbulent-Flow Simulations. *IEEE Computer Graphics and Applications*, 32(4):0034–45, 2012.

[10] P. Gioia, O. Aubault, and C. Bouville. Real-time reconstruction of wavelet-encoded meshes for view-dependent transmission and visualization. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(7):1009–1020, 2004.

[11] K. Gruchalla, M. Rast, E. Bradley, J. Clyne, and P. Mininni. Visualization-driven structural and statistical analysis of turbulent flows. In *Advances in Intelligent Data Analysis VIII*, pages 321–332. Springer, 2009.

[12] S. Guthe and W. Straßer. Real-time decompression and visualization of animated volume data. In *Visualization, 2001. VIS'01. Proceedings*, pages 349–572. IEEE, 2001.

[13] C. Harrison, H. Childs, and K. P. Gaither. Data-Parallel Mesh Connected Components Labeling and Analysis. In *Proceedings of EuroGraphics Symposium on Parallel Graphics and Visualization (EGPGV)*, pages 131–140, Llandudno, Wales, Apr. 2011.

[14] I. Ihm and S. Park. Wavelet-based 3d compression scheme for interactive visualization of very large volume data. In *Computer Graphics Forum*, volume 18, pages 3–15. Wiley Online Library, 1999.

[15] B.-J. Kim and W. A. Pearlman. An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (spiht). In *Data Compression Conference, 1997. DCC'97. Proceedings*, pages 251–260. IEEE, 1997.

[16] J. Ma, D. Murphy, C. O'Mathuna, M. Hayes, and G. Provan. Visualizing uncertainty in multi-resolution volumetric data using marching cubes. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 489–496. ACM, 2012.

[17] S. Mallat. *A wavelet tour of signal processing: the sparse way*. Academic press, 2008.

[18] W. A. Pearlman, A. Islam, N. Nagaraj, and A. Said. Efficient, low-complexity image coding with a set-partitioning embedded block coder. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(11):1219–1235, 2004.

[19] F. 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. IEEE, 1999.

[20] A. Skodras, C. Christopoulos, and T. Ebrahimi. The jpeg 2000 still image compression standard. *Signal Processing Magazine, IEEE*, 18(5):36–58, 2001.

[21] G. Strang. Wavelets and dilation equations: A brief introduction. *SIAM review*, 31(4):614–627, 1989.

[22] H. Tao and R. J. Moorhead. Progressive transmission of scientific data using biorthogonal wavelet transform. In *Proceedings of the conference on Visualization'94*, pages 93–99. IEEE Computer Society Press, 1994.

[23] P. C. Wong and R. D. Bergeron. Authenticity analysis of wavelet approximations in visualization. In *Proceedings of the 6th conference on Visualization'95*, page 184. IEEE Computer Society, 1995.

[24] J. Woodring, S. Mniszewski, C. Brislawn, D. DeMarle, and J. Ahrens. Revisiting wavelet compression for large-scale climate data using jpeg 2000 and ensuring data precision. In *Large Data Analysis and Visualization (LDAV), 2011 IEEE Symposium on*, pages 31–38. IEEE, 2011.