Evaluating the Efficacy of Wavelet Configurations on Turbulent-Flow Data

Shaomeng Li
University of Oregon

Kenny Gruchalla
National Renewable Energy Laboratory

Kristin Potter
University of Oregon

John Clyne
National Center for Atmospheric Research

Hank Childs
University of Oregon
Lawrence Berkeley Nat'l Lab

Abstract

I/O is increasingly becoming a significant constraint for simulation codes and visualization tools on modern supercomputers. Data compression is an attractive workaround, and, in particular, wavelets provide a promising solution. However, wavelets can be applied in multiple configurations, and the variations in configuration impact accuracy, storage cost, and execution time. While the variation in these factors over wavelet configurations have been explored in image processing, they are not well understood for visualization and analysis of scientific data. To illuminate this issue, we evaluate multiple wavelet configurations on turbulent-flow data. Our approach is to repeat established analysis routines on uncompressed and lossy-compressed versions of a data set, and then quantitatively compare their outcomes. Our findings show that accuracy varies greatly based on wavelet configuration, while storage cost and execution time vary less. Overall, our study provides new insights for simulation analysts and visualization experts, who need to make tradeoffs between accuracy, storage cost, and execution time.

1 Introduction

The design of modern supercomputers is constrained both in how much money can be spent on its components and in how much energy the machine can consume when operating. These constraints force supercomputer architects to make difficult tradeoffs among the system components (e.g., networking, I/O, memory, and computational speed) to balance their budgets. Over the last decade, architects have devoted smaller and smaller percentages of their budgets toward I/O. As a result, both the I/O bandwidth and the storage capacity are often not keeping pace with these supercomputers' abilities to generate data.

For visualization and analysis, *in situ* processing has shown great promise in addressing this reduced I/O capabilities. However, *in situ* works best when analysts know the visualizations and analyses that they want to perform *a priori*. In a data exploration-oriented setting, the traditional *post hoc* model — where data is stored to disk and explored on human time scales afterwards — is still the dominant paradigm. And, in this paradigm, I/O performance is frequently the bottleneck for overall performance [4].

In the image processing community, data reduction, especially wavelet-based lossy compression techniques, has proven to be effective in addressing both I/O bandwidth and storage capacity. While image data and scientific data share characteristics, namely that they are both non-periodic and self-similar (i.e., neighboring data points are likely to have similar values), the efficacy of wavelet compression observed for image data does not clearly map to scientific data. Though one could reasonably assume that the benefits from wavelet-based lossy compression will translate, the magnitude of effects are still unknown. That is, visualization experts and simulation scientists do not have a clear sense of the tradeoffs in accuracy, storage cost, and execution time for their data sets. With this study, we sought to answer these questions.

Our experiments applied wavelet compression to large scientific data sets, specifically turbulent-flow data sets, and measured how well lossy compressed forms could be used for visualizations and analyses. Each of our evaluations, including our accuracy evaluations, was quantitative in nature, making for clear comparisons. That said, 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.

Traditionally, the scientific visualization community has used wavelet transforms in a configuration employing multi-resolution hierarchies and the Haar kernel. This arrangement provides progressive data access, and also achieves data reduction when reading data representations at coarser resolutions. The image processing community, however, is employing alternate approaches. Currently, this community makes heavy use of prioritized coefficients to organize their data, and Cohen-Daubechies-Feauveau (CDF) wavelet kernels. With this study, we want to consider all of these configurations, their efficacies, and how they compare.

The work in this paper is an evaluation study. The goal of this evaluation is to provide more guidance to visualization experts and simulation scientists when deciding to apply wavelet compression in their work. Its specific contributions are:

- Demonstration that configurations currently favored by the image processing community also perform well in the context of meaningful scientific analyses applied to turbulent-flow data. Specifically:
  - Coefficient prioritization performs better than the multi-resolution approach.
  - The CDF 9/7 kernel performs better than the Haar kernel.
- Evaluation of wavelet configurations that focuses on quantifying differences, which enables intuitive comparisons.

This paper is organized as follows: Section 2 describes basics of wavelet compression, and surveys relevant research. We provide an overview of our study in Section 3, and describe the two established visual analyses that we repeat in Section 4. Section 5 describes our evaluation metrics and efficacy results for both visual analyses. Section 6 augments this analysis to illuminate tradeoffs in accuracy, storage cost, and execution time. We conclude the work in Section 7.

2 Background

We begin by discussing the basics of data compression using wavelet transforms in Section 2.1. We then discuss wavelet configurations: Section 2.2 describes two compression strategies and Section 2.3 describes three wavelet kernel choices. Finally, we survey prior work using wavelet compression for visualization and analyses in Section 2.4.

2.1 Data Compression Using Wavelet Transforms

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

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

Each coefficient a_k measures the similarity between the corresponding basis function $u_k[n]$ and the input data array $x[n]$. This transform itself is lossless and $x[n]$ can be reconstructed by calculating the above expansion using all K coefficients. However, wavelets are frequently used in a lossy manner [8, 19, 26]; when reconstructing $x[n]$, an approximation, $\tilde{x}[n]$, can be calculated by using a subset of the $\widetilde{K}$ coefficients:

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

Since the basis functions, $u_k[n]$, can be calculated on the fly, data compression is achieved by storing a subset of coefficients, and reconstructing $\tilde{x}[n]$ using only this subset. The compression ratio is thus derived from the fraction of coefficients used. For instance, to achieve an 8:1 compression ratio, one eighth of the total number of coefficients would be used.

Wavelet transforms can be applied recursively on coefficients from previous wavelet transforms, resulting in a hierarchy of coefficients. This recursive application of wavelet transforms helps concentrate information content into fewer and fewer coefficients, and the resulting coefficient hierarchy enables a data representation spanning multiple resolutions.

2.2 Compression Strategy Options

There are multiple ways to lay out wavelet coefficients. Because we always read the $\widetilde{K}$ coefficients from a consecutive storage space, the ordering of coefficients determines which coefficients are included to reconstruct the approximation, and thus determines the compression strategy. With our study, we evaluated a multi-resolution strategy and prioritized coefficient strategy.

Multi-resolution: With a multi-resolution approach, coefficients are laid out naturally with respect to the coefficient hierarchy. Because each level of coefficients reconstructs an approximation of the original data array, compression is achieved by storing only some levels of coefficients from the hierarchy. Coefficients stored in this manner retain their addresses, i.e., where they belong to in the coefficient hierarchy, and thus do not require additional addressing mechanisms.

Each iteration of wavelet transform coarsens the data array into half of its previous resolution. For example, the Haar wavelet kernel calculates the unweighted average of two neighbor elements as the approximation, which has one value. As a result, the multi-resolution compression strategy offers a pyramid representation that is strictly limited to power-of-two reductions along each axis. In the case of a three-dimensional regular mesh, the applicable compression ratios are of the form 8^N:1, where N is the number of iterations of wavelet transform to apply. That is, applicable compression ratios include 8:1, 64:1, 512:1, etc.

Finally, the multi-resolution wavelet approach differs from similar techniques used by the visualization community, such as mipmapping [31], and space-filling curves [20, 17]. In the first case, mipmapping requires additional storage space for the coarsened approximations. In the second case, the coarsened approximations come from single point data (nearest neighbor sampling), rather than average of all points in that region.

Prioritized Coefficients: When reconstructing the original data from the wavelet expansion given in Equation 1, coefficients have different importance, i.e., coefficients representing the more rapidly changing parts of the original data contribute more than coefficients representing the more self-similar parts. The prioritized coefficient technique makes use of this property by laying out the coefficients based on their importance, i.e., important coefficients are placed toward the beginning of the storage space. Compression is thus achieved by storing only the collection of important coefficients, and treating the rest coefficients as zeros.

The prioritized coefficient strategy differs from multi-resolution strategy in that: 1) it supports an arbitrary compression ratio, by choosing what percentage of total coefficients to keep; 2) it supports reconstructing the mesh on its full resolution, by filling zeros to the coefficient locations that are not stored; and 3) it requires extra mechanisms to keep track of where the prioritized coefficients belong in the coefficient hierarchy. The software we used to perform wavelet compression keeps coefficient addresses explicitly, thus introducing storage overhead. We will further discuss this addressing choice and quantify this storage overhead in Section 6.2.

2.3 Wavelet Kernel Choices

Wavelet kernels are used to generate the basis functions from Equation 1. These basis functions, in turn, have different efficacies when used for data compression. In our study, we consider three wavelet kernels: Haar, CDF 9/7, and CDF 8/4.

The Haar [14] kernel is one of the most basic and widely understood wavelet kernels. It serves as a baseline for our evaluation. The Haar kernel generates a series of “square-shaped” functions for its basis functions. When used with the multi-resolution strategy, the Haar kernel yields a hierarchy representation that is identical to that produced with a linear (trilinear in three dimensions) downsampling filter. Because the Haar kernel is so simple, it introduces only a modest computational cost (see Section 6.3).

Both CDF 9/7 and CDF 8/4 are from the Cohen-Daubechies-Feauveau [7] biorthogonal wavelet family. Members in this family differ based on their filter sizes, which are indicated as the suffix, e.g., 9/7 or 8/4, and to the degree with which they resemble orthogonal wavelets. The CDF family of wavelets is widely used in the compression of non-periodic signals (e.g. images and video) due to its effective boundary handling capabilities [29]. The CDF 9/7 kernel, in particular, has been empirically shown to yield superior compression distortion results for imagery [1, 30], and is widely used for multimedia compression applications including JPEG 2000 [25]. The CDF 8/4 kernel is the default wavelet kernel used in an open-source visualization and analysis software, VAPOR [5, 6], so we also include it into our evaluation.

2.4 Prior Research of Wavelets in Visual Analytics

Wavelet compression has been previously employed with scientific visualization. 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 [15, 23]. When used on three-dimensional volume data sets, such as hydrodynamic simulations, global ocean models, or terrain data, wavelet compression has been shown to be effective when visualizing different levels of detail [2, 24, 27, 21]. Wavelet compression also brings new possibilities for real-time analysis on large-scale data sets on commodity hardware [10, 13]. A $1{,}024^3$ turbulent-flow data set was also visualized at a rate of 5 seconds per time step on desktop PCs, via reduced I/O from applying wavelet compression [28].

Some notable prior studies have also tried to understand the effects of wavelet compression on scientific data. Wong et al. proposed an energy-based model to analyze the authenticity of orthogonal wavelet compressed volume data [32]. Woodring et al. and Ma et al. further introduced visualization techniques to encode the

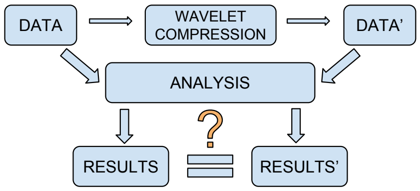


Figure 1: Our experiment methodology. We used wavelet compression to create a compressed form (DATA') from its original form (DATA). *RESULTS* and *RESULTS'* represent the analysis results from *DATA* and *DATA'*, respectively. Our study then quantitatively evaluated the difference between *RESULTS* and *RESULTS'*.

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

Figure 2: Wavelet configurations studied. Cross signs represent configurations examined in our first round of experiments, circles represent configurations examined in our second round of experiments, and squares represent configurations examined in our third round of experiments.

amount of variance in a location, providing the ability to examine local information loss at points of interest [33, 18]. Further, the Woodring et al. study provided exact error bounds for each data compression level, allowing domain scientists to get a precision guarantee when analyzing compressed data sets. Finally, Gralka et al. included wavelet compression into an application-specific compression pipeline for particle-based data, and evaluated the effectiveness of their compression pipeline [11].

In contrast to previous studies of wavelet compression on scientific data, we consider a variety of wavelet configurations, and evaluate tradeoffs in accuracy, storage cost, and execution time.

3 Study Overview

We studied multiple wavelet configurations, varying over compression strategies, wavelet kernels, and compression ratios. Section 3.1 describes our experiment methodology for a generic configuration, and Section 3.2 describes the different wavelet configurations we studied.

3.1 Experiment Methodology

Our experiment methodology, illustrated in Figure 1, was as follows:

- We began with turbulent flow data in its raw form.
- We applied wavelet compression to the raw data to get the compressed form.
- We applied an analysis routine to the data in both its raw and compressed forms.
- We quantitatively evaluated the difference between the results.

This wavelet transformations were performed using the VAPOR software package [5, 6]. VAPOR also has advantages over other implementations, as in [33] for example, that VAPOR natively supports wavelet transforms in three dimensions, and operates on floating point data. For the analysis routines, we used the software that was used to perform the analysis originally: VisIt [3] or VAPOR.

3.2 Wavelet Configurations Studied

We performed our experiments in three rounds.

In the first round, we considered the wavelet compression strategy. Specifically, we compared multi-resolution with prioritized coefficients (see Section 2.2), both using the Haar kernel. Since the multi-resolution approach requires two-to-one reduction in all three dimensions, only reductions that are powers of eight are possible. The three compression ratios we studied for this round were 8:1, 64:1, and 512:1.

In the second round, we studied the effects of the wavelet kernels. Specifically, we compared the Haar kernel, the CDF 9/7 kernel, and the CDF 8/4 kernel. All tests in this round used the prioritized coefficients strategy. The compression ratios for this round were 8:1, 16:1, 32:1, 64:1, 128:1, 256:1, and 512:1. With prioritized coefficients, arbitrary ratios would have been possible, but we chose powers of two for easy comparisons with the multi-resolution approach.

In the third round, we repeated the wavelet configurations from the first two rounds, but with a different analysis routine.

Figure 2 summarizes the wavelet configurations we studied over all three rounds.

4 Visual Analytics Overview

Our study incorporated two analysis routines, both of which came from established research on turbulent-flow data. These two analyses were both performed on rectilinear data sets from simulations. In our study, we located the original data sets used in the two predecessor studies. We denote their data sets DS1 and DS2 for simplicity. DS1 contained a single scalar field defined over thirteen time slices on a $4,096^3$ mesh. DS2 did not vary in time. It contained a scalar field and a vector field, both defined on a $1,024^3$ mesh. For DS1 and DS2, the scalar field was "enstrophy," which directly measures the kinetic energy in a flow model. For DS2, the vector field was velocity.

Both established analysis routines identified and analyzed critical structures, which are defined as regions with significantly higher enstrophy values than the areas surrounding it. However, they focused on the critical structures' properties in different scopes. The first analysis (Section 4.1) focused on the global population of all critical structures, while the second analysis (Section 4.2) focused on the local dynamics of individual critical structures.

Finally, we note that our goal is to evaluate tradeoffs in compression and accuracy on established analysis routines with wavelet compressed data over a variety of wavelet configurations. In particular, if an analysis routine is especially sensitive to lost accuracy in the data, then we view that as a finding for our study, but not as a cue that we should extend or modify the established analyses.

We describe the key steps of these two visual analytics routines in the following subsections.

4.1 Critical Structure Identification

Gaither et al. performed an analysis that included measuring the global population of critical structures [9]. Identifying these critical structures took two steps. The first step isolated regions with enstrophy values higher than α, a fixed value provided by domain scientists. For reference, the test data set DS1 contains millions of these high-enstrophy regions. The second step eliminated structures with a volume smaller than a threshold β, again a fixed value provided by domain scientists. For DS1, this process reduces the number of critical structures down to hundreds. Figure 3 shows a screenshot of these identified critical structures at the first time slice of DS1, as well as a close-up look at one of the critical structures.

This analysis routine can potentially be quite sensitive to changes in the enstrophy field from compression. If the compressed enstrophy breaks a component, then the result may put that component below β. Similarly, if the compressed enstrophy joins two disjoint

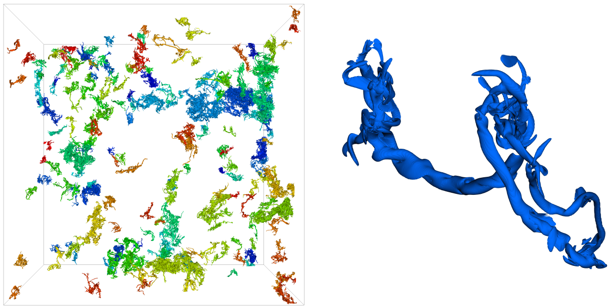


Figure 3: The left rendering shows critical structures identified from the first time slice of DS1. Each structure has a unique color in this view. The right rendering shows a close-up view of one of the critical structures.

components, then the result may put the joined component above β. Such a change would affect the statistics of the global population of critical structures.

4.2 Local Dynamics Analysis

Gruchalla et al. analyzed the dynamics of individual structures in a turbulent-flow simulation [12]. Specifically, for a single structure, they studied the change in velocity from the inside of the structure to the outside. For their analysis, structures were first identified following steps similar to those described in Section 4.1, and then representatives from two distinct types of local dynamics were picked to perform further analysis. Renderings of these two representations can be found in Figure 8a and 8b in the the results discussion. The high-enstrophy areas of these structures are rendered in blue, and their local dynamics are illustrated by yellow streamlines seeded in the velocity field. We refer to these two types of structures as S1 and S2. Visually, the two types of dynamics can be distinguished from each other, since streamlines twist around the core with S1, and follow the writhe of the tube with S2.

Their primary analysis looked at *radial-enstrophy profiles* — enstrophy as a function of radius — for individual structures. They created this profile by considering fifteen cross sections along the major axis of a structure. Within a cross section, they identified the center of the structure for that cross section. They then calculated the average enstrophy around this center for many different radii. This resulted in a radial-enstrophy profile for that cross section. They then averaged the radial-enstrophy profiles over all cross sections to create the final radial-enstrophy profile.

A radial-enstrophy profile captures local flow dynamics. For S1, the profile shows high enstrophy values in the core, and drops rapidly when exiting the structure. For S2, the profile shows moderate enstrophy values in the core and drops slowly when exiting the structure.

5 Efficacy Evaluations

In this section we present evaluation results from our two established analyses. Section 5.1 presents the results from the first analysis: critical structure identification. This section consists of the first two rounds of our experiment. Section 5.2 presents the results from the second analysis: local dynamics analysis. This section consists of the third round of our experiment.

For each of the two analyses, we first describe our evaluation metric, and then present the evaluation results.

5.1 Evaluation: Critical Structure Identification

5.1.1 Evaluation Metric

The critical structure identification task yields some number of identified structures on both the raw data and the wavelet-compressed data. 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, in practice, the critical structures do not always align in this ideal way.

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*. Our evaluation metric is based on the number of false positives and false negatives — the lower these two numbers are, the better the results from the compressed data match the baseline results from the raw data.

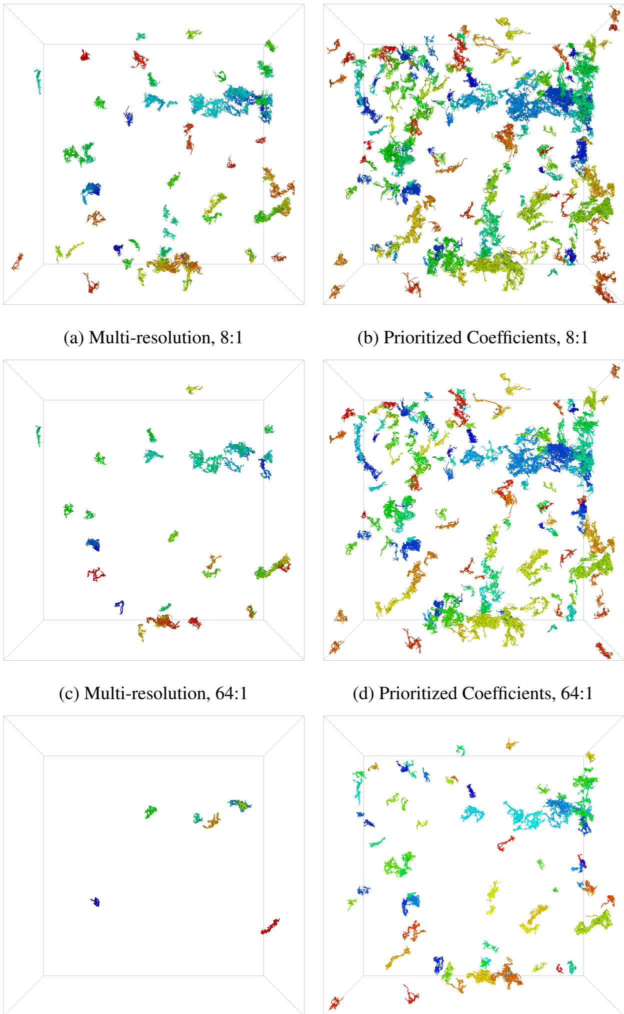
To provide a better comparison among all time slices, we considered the proportion of error among the critical structures, rather than absolute numbers. Formally, let FN be the number of false negatives, FP be the number of false positives, and $COMM$ be the number of critical structures common to both. We then focused on two error metrics: $FN\_Proportion$, defined as $FN/(FN+COMM)$, and $FP\_Proportion$, defined as $FP/(FP+COMM)$. Both metrics range between zero and one, with values closer to zero being better.

Determining if an identified critical structure is common to both is not a trivial task. We used a proximity test to match up critical structures in $COMM$. 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 one, and no overlap would score zero. 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).

5.1.2 Results: Wavelet Compression Strategy

We evaluated the two wavelet compression strategies — prioritized coefficients and multi-resolution (see Section 2.2) — with three compression ratios: 8:1, 64:1, and 512:1. Each test used the Haar kernel. This resulted in six different wavelet settings. Figure 4 shows renderings from our analysis using each of the six settings on the first time slice of DS1. Visual inspection shows that results using the prioritized coefficient strategy not only retain more critical structures from the baseline, but also preserve more shape details.

Figure 5 compares FN_Proportion and FP_Proportion between the prioritized coefficient strategy and the multi-resolution strategy. It plots the average values over all thirteen time slices. Prioritized coefficients clearly outperform the multi-resolution approach, as the blue lines are significantly lower than the red ones. That said, FN for multi-resolution drops at the 512:1 ratio. This is because the multi-resolution strategy fails to identify most of the structures at this compression level, so the few identified ones are likely to be correct. Restated, this low false positive proportion does not indicate better performance for the multi-resolution strategy.



(a) Multi-resolution, 8:1 (b) Prioritized Coefficients, 8:1

(c) Multi-resolution, 64:1 (d) Prioritized Coefficients, 64:1

(e) Multi-resolution, 512:1 (f) Prioritized Coefficients, 512:1

Figure 4: Screenshots from our critical structure identification task using the multi-resolution strategy (left column) and the prioritized coefficient strategy (right column). These screenshots are from the first time slice of DS1, so the left image of Figure 3 shows the baseline result for raw data for this task.

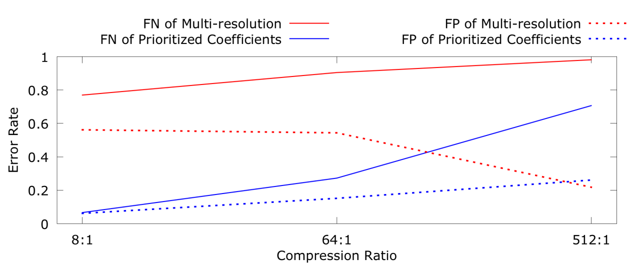
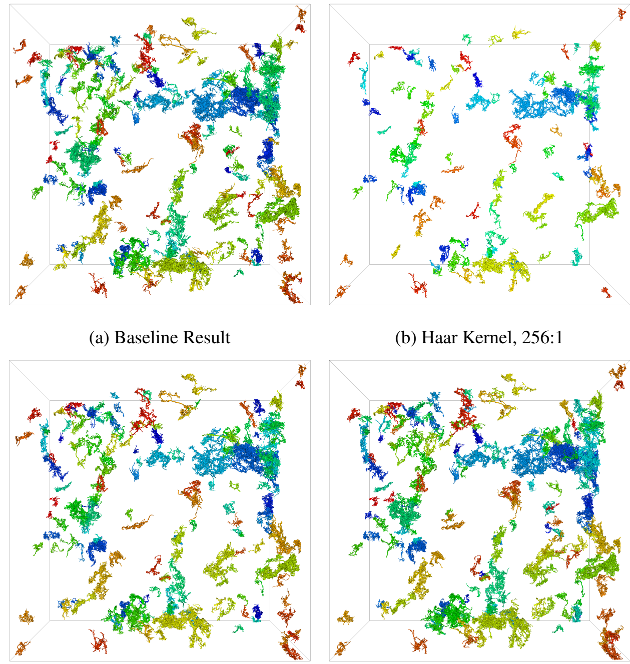


Figure 5: False negative (solid lines) and false positive (dashed lines) proportions for two compression strategies. The multi-resolution results are colored red, and the prioritized coefficient results are colored blue. Each line is an average of results from all thirteen time slices.



(a) Baseline Result (b) Haar Kernel, 256:1

(c) CDF 9/7 Kernel, 256:1 (d) CDF 8/4 Kernel, 256:1

Figure 6: Screenshots from our critical structure identification analysis on the first time slice of DS1. All the compressed results (b, c, and d) use a prioritized coefficient strategy.

5.1.3 Results: Wavelet Kernel Choice

We then expanded our kernel choices to include the CDF 9/7 and CDF 8/4 kernels. This meant there were a total of three kernels, as 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 seven: 8:1, 16:1, 32:1, 64:1, 128:1, 256:1, and 512:1. Thus, the total number of experiments was 21. Figure 6 shows the visual difference among three wavelet kernels using the 256:1 compression ratio (6b, 6c, and 6d), and their comparison to the baseline (6a). Visual inspection shows that while all kernels capture many structures, CDF 9/7 and CDF 8/4 manage to keep more details than Haar.

We plotted FN_Proportion and FP_Proportion for the three wavelet kernels in Figure 7. Again they are averaged over all thirteen time slices. The top plot shows that the CDF 9/7 and CDF 8/4 have similar false negative proportions, and they are both lower than the Haar kernel. The bottom plot shows that the CDF 9/7 kernel has the lowest false positive proportions at every compression ratio by a clear margin. Summing up, results from this evaluation indicate that CDF 9/7 is the best choice among these three wavelet kernels for this analysis.

5.2 Evaluation: Local Dynamics Analysis

5.2.1 Evaluation Metric

Our evaluation process began by identifying the two structures, S1 and S2, in the raw and wavelet-compressed versions of the data. We then calculated their radial-enstrophy profiles. Ideally, the profile produced using the compressed data would be the same as the profile from raw data, i.e., the radial-enstrophy plots would overlap with each other. However, in practice, there were differences between the two profile lines. We evaluated the wavelet compression by quantifying the difference between these two radial-enstrophy profiles — the smaller the difference, the better the compressed data preserved local dynamics.

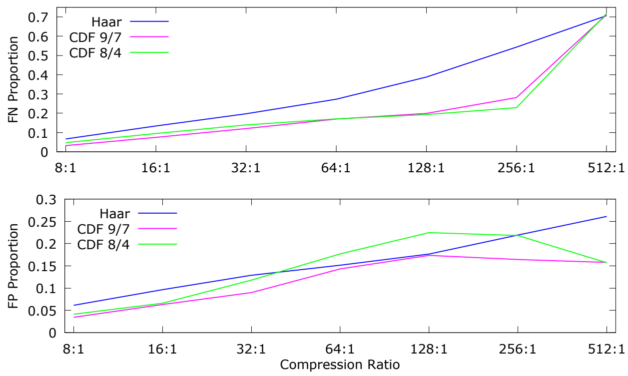
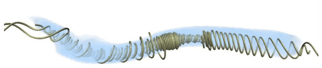
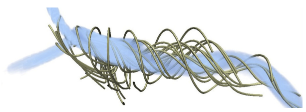


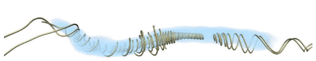
Figure 7: False negative (top) and false positive (bottom) proportions for three wavelet kernels. Each line is an average of results from all thirteen time slices.



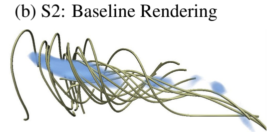
(a) S1: Baseline Rendering



(b) S2: Baseline Rendering



(c) S1: CDF 9/7+Prioritized, 128:1



(d) S2: Haar+Multi-resolution, 64:1

Figure 8: Visualizations of identified critical structures (rendered in blue) and their local dynamics (rendered in yellow). The top subfigures show the baseline rendering using the raw data, and the bottom subfigures show the rendering using the compressed data.

We used the root mean square error (RMSE) metric to measure the difference between the two profiles. Specifically, given the baseline radial-enstrophy profile $E[r]$ $(0 \leqslant r < N)$ and the radial-enstrophy profile from compressed data $\widetilde{E}[r]$ $(0 \leqslant r < N)$, RMSE is then defined as:

$$RMSE \ = \ \sqrt{\frac{\sum_{r=0}^{N-1} \ (E[r] \ - \ \widetilde{E}[r])^2}{N}} \ . \tag{3}$$

In this work, we normalized RMSE by the observed data range. Thus, the normalized RMSE (a.k.a. NRMSE) evaluated to zero when $E[r]$ and $\widetilde{E}[r]$ are exactly the same, and evaluated to one in the worse possible case.

5.2.2 Evaluation Results

We included all three wavelet kernels and both compression strategies for this evaluation. The multi-resolution strategy used three compression ratios (8:1, 64:1, and 512:1), and the prioritized coefficient strategy used five compression ratios (8:1, 64:1, 128:1, 256:1, and 512:1). So the total number of wavelet configurations tested was eighteen. Figure 8a and 8b show baseline renderings for these tests, and Figure 8c and 8d show two examples from the eighteen configurations. Visual inspection shows that data compression changes the streamlines in both structures compared to the baseline renderings (Figure 8).

Figure 9 shows the NRMSE of the radial-enstrophy profile for the two structures after data compression. Both NRMSE plots show that the three wavelet settings using prioritized coefficients yield significantly lower errors than Haar+multi-resolution. In addition, when using prioritized coefficients, the two CDF kernels always perform better than the Haar kernel at compression ratios up to 256:1, and the CDF 9/7 kernel has the lowest errors at most configurations. This result is consistent with our findings in Section 5.1.

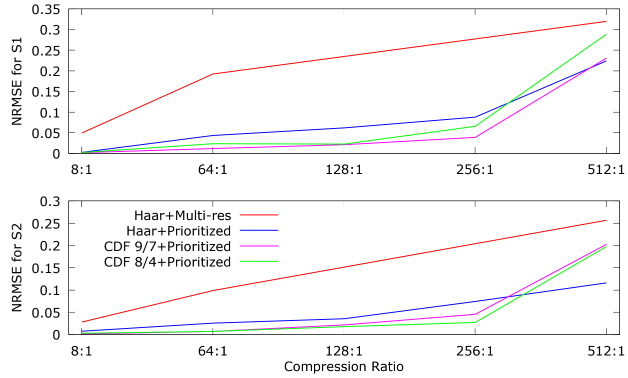


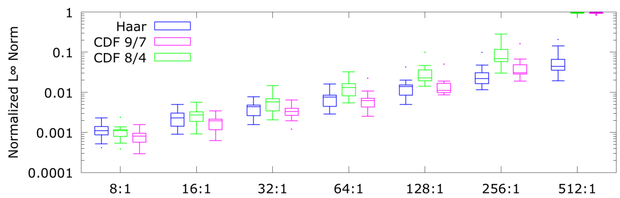
Figure 9: NRMSE of the radial-enstrophy profile for S1 (top) and S2 (bottom). Each color represents a wavelet setting in our experiment, with the legend displayed in the bottom figure. The red color represents Haar+multi-resolution, which only supports compression ratios at 8:1, 64:1, and 512:1 (see Section 2.2). We connect the data points of 64:1 and 512:1 in this setting using a straight line.

Table 1: Accuracy summary of different wavelet configurations. The multi-resolution+Haar configuration serves as the baseline, and the improvements from other configurations are shown comparatively.

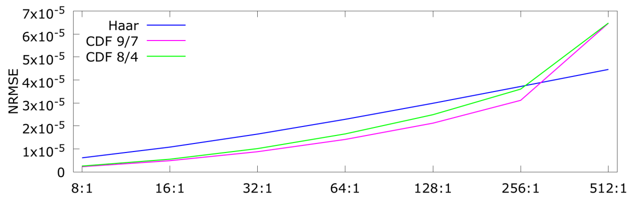
Wavelet Configurations	Analysis 1 Proportion of FN	FP	Analysis 2 NRMSE for S1	S2
(8:1 Comp.)				
Multi-res+Haar	1x	1x	1x	1x
Prioritized+Haar	11.63x	9.16x	19.31x	3.74x
Prioritized+CDF 8/4	16.40x	13.60x	21.97x	9.93x
Prioritized+CDF 9/7	24.27x	16.35x	26.72x	21.88x
(64:1 Comp.)				
Multi-res+Haar	1x	1x	1x	1x
Prioritized+Haar	3.32x	3.59x	4.43x	3.86x
Prioritized+CDF 8/4	5.31x	3.07x	8.19x	14.05x
Prioritized+CDF 9/7	5.34x	3.80x	16.40x	14.05x
(512:1 Comp.)				
Multi-res+Haar	1x	1x	1x	1x
Prioritized+Haar	1.39x	0.83x	1.43x	2.21x
Prioritized+CDF 8/4	1.37x	1.39x	1.11x	1.30x
Prioritized+CDF 9/7	1.38x	1.38x	1.39x	1.27x

5.3 Summary From Two Analyses

Table 1 summarizes our findings regarding how accuracy compares over wavelet configuration. In this table, the multi-resolution+Haar configuration serves as the baseline result, with the accuracy gains from other configurations shown comparatively. Only compression ratios of the form 8^N are shown, since those are the only ratios supported by the multi-resolution strategy. In all cases, the prioritized coefficients strategy using a CDF kernels perform best. The two CDF kernels perform similarly in most cases, although CDF 9/7 outperforms CDF 8/4 several times. Finally, the improvement is very significant at finer representations, but less noteworthy at very coarse representations.



(a) Normalized L^{∞}-norm on wavelet compressed data. The normalized values are computed by scaling the absolute L^{∞}-norms by the maximum enstrophy in DS1. Each box in this box plot represents a distribution of the L^{∞}-norm over thirteen time slices of DS1. The Y-axis is on a logarithmic scale.



(b) NRMSE of wavelet compressed data. The NRMSE is computed by scaling the absolute RMSE value by the maximum enstrophy in DS1. Each line in this line chart represents an average of DS1's thirteen time slices.

Figure 10: Statistical error measurements of wavelet compressed data.

6 Further Evaluation of Accuracy, Storage Cost, and Execution Time

Our evaluation in the previous section helps illuminate tradeoffs between compression and accuracy in real-world scientific analyses. With this section, we study accuracy and compression further, and also explore storage overhead and execution time:

- Section 6.1 uses statistical error measurements to evaluate different wavelet configurations.
- Section 6.2 studies storage cost, as the prioritized coefficient strategy introduces overheads that make data savings be less than the desired compression ratio.
- Section 6.3 reports on the execution times to apply both the forward and inverse wavelet transforms.

6.1 Statistical Error Measurements

While the focus of our study was on evaluating wavelet efficacy for established analyses, we also wanted to understand traditional statistical error measurements. The measurements we chose to perform were the L^{∞}-norm and the root mean square error. We chose these two metrics because they measure extreme differences and average differences, respectively. Specifically, the L^{∞}-norm captures the largest possible point-wise difference between the original and compressed data, and RMSE provides an average error across all vertices in the volume.

We performed our calculations by directly comparing every pair of corresponding vertices from the original and compressed data for all thirteen time slices of DS1. Because this comparison is meaningful only when the compressed data has the same mesh resolution as the original data, the multi-resolution strategy was not used. Figure 10a represents the normalized L^{∞}-norm using box plots, and Figure 10b shows the normalized RMSE (NRMSE). The CDF 9/7 kernel consistently performs the best up to 128:1 compression with the L^{∞}-norm, and up to 256:1 with RMSE. This result is consistent with our previous findings. We also note that while the CDF 8/4 kernel generally outperforms the Haar kernel in evaluations both in Section 5 and in the RMSE evaluation of this section, it yields larger L^{∞}-norm values than the Haar kernel.

Table 2: File sizes for wavelet-compressed data in GB. Our test data set was 256 GB, and was compressed using the Haar kernel with multi-resolution and prioritized coefficient strategies. The actual achieved compression ratios are shown in parentheses. The four “N/A” entries are ratios that the multi-resolution scheme does not support.

Comp. Ratio	Ideal	Multi-resolution	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)

6.2 Storage Overhead

Table 2 shows file sizes at different compression ratios for the first time slice of DS1 (256 GB in raw form). The rest of the time slices have the same file size since they the same mesh resolution. We tested both multi-resolution and prioritized coefficient schemes. Because the prioritized coefficient scheme essentially introduces the same storage overhead regardless of wavelet kernel, we only report results from the Haar kernel. This table shows that the multi-resolution scheme achieves ratios close to ideal, i.e., the file sizes are very close to being in proportion with the compression ratio. It also shows that the prioritized coefficient scheme introduces slightly more than 50% overhead in storage.

The multi-resolution scheme is able to achieve full storage savings since the scheme can store its coefficients in the order generated by the forward wavelet transforms; the addressing of coefficients is implicit. The slight increases over the compression ratio are from metadata stored in the file.

The prioritized coefficient scheme must order coefficients based on their information content. This re-ordering requires tracking their addresses, and, in our study, the addressing is explicit. We note that, in the image processing space, encoders such as SPIHT [16] and SPECK [22] are able to avoid this overhead with complex encoding strategies. However, their approaches require byte-scaling floating point data to integers, which may introduce additional information loss.

6.3 Execution Time

Wavelet transformation introduces computational overhead when writing (to perform the forward wavelet transform that compresses the data) and reading (to perform the inverse wavelet transform that decompresses the data). In this section, we examine these computational overheads with different wavelet configurations, as the multi-resolution and prioritized coefficients compression strategies have different characteristics. For the multi-resolution strategy, the computational cost is closely related to the compression ratio. This is because more aggressive compression is achieved by applying the wavelet transform repeatedly to the data, thus introducing more computational burden. In our study, we performed the wavelet transform three times (and thus achieved a compression ratio of 512:1). For the prioritized coefficient strategy, the computational cost is independent of the compression ratio, because we always reconstruct the meshes at their native resolutions.

We ran our tests on a subset of DS1 that measured 4 GB in raw form. We did not use the whole data set since any given node of a supercomputer will be operating only on a portion of the overall data set.

Our experiment used one compute node on Maverick, a machine at the Texas Advanced Computing Center. Compute nodes on this

Table 3: Time cost, in seconds, to perform Discrete Wavelet Transform (DWT) and Inverse Discrete Wavelet Transform (IDWT) on a 4 GB subset of DS1.

	Multi-resolution	Prioritized Coefficients		
	Haar	Haar	CDF 9/7	CDF 8/4
DWT	11.4297	12.9927	14.2177	13.9134
IDWT	5.2971	2.2621	3.8233	3.0584

machine have 20 CPU cores and 256 GB system memory. We used community software (VAPOR) to perform the wavelet compression. This software was multi-threaded when using the prioritized coefficient strategy, i.e., it spawned 20 threads on our test machine. However, the software ran single-threaded when using the multi-resolution strategy.

Table 3 reports the run time to perform the forward Discrete Wavelet Transform (DWT) and the Inverse Discrete Wavelet Transform (IDWT), with each measurement averaged over ten runs. The results show that the prioritized coefficient strategy, even with improved parallelism, introduces significantly larger computational costs than the multi-resolution strategy. The CDF 9/7 kernel, which performed best in our accuracy evaluations, is the slowest to execute. Finally, we notice that the IDWT operations take significantly less time than DWT, meaning that it is much faster to decode wavelet-compressed data in a *post hoc* analysis.

While the run times to apply DWT are greater than ten seconds, they may be acceptable for *in situ* usage. We envision wavelet compression running only when the simulation wants to output data; since this happens infrequently and, since I/O is a costly operation, the overhead from the compression is likely small in comparison.

7 Conclusion

We performed an evaluation study on the efficacy of wavelet configurations for turbulent-flow simulations. Our approach took two existing visual analytics routines and repeated them on compressed data sets, varying over compression strategies (multi-resolution and coefficient prioritization), wavelet kernels (Haar, CDF 9/7, and CDF 8/4), and compression ratios. We complemented this analysis with traditional statistical error measurements, additional information on storage requirements, and computational overhead for applying wavelet transforms. In total, this study informs the tradeoffs between accuracy, storage cost, and execution time when applying wavelets to turbulent-flow data.

Our findings show that the coefficient prioritization approach and the CDF kernels provide significant benefits over the multi-resolution schemes that rely on (tri)linear filtering to produce coarsened data (Haar as an example, see Section 2.3). Since the experiments we performed were diverse and their results were consistent, we believe that these findings are likely to generalize to other scientific visualization usages as well. While our findings match best practices from the image processing community, our focus on quantifying the accuracies achieved for domain scientist's analyses allow us to determine the magnitude of the benefit for real world settings. Interestingly, the variation in overall accuracy was quite high across analysis routines, emphasizing the importance of keeping the final usage in mind.

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

8 Acknowledgments

The authors acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing visualization resources that have contributed to the research results reported within this paper. This work was supported by the DOE Early Career Award for Hank Childs, Contract No. DE-FG02-13ER26150, Program Manager Lucy Nowell. This work was also funded in part by US National Science Foundation grant No. 1440412.

References

[1] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Transactions on Image Processing*, 1(2):205–220, 1992.

[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.

[4] H. Childs, D. Pugmire, S. Ahern, B. Whitlock, M. Howison, Prabhat, G. Weber, and E. W. Bethel. Extreme Scaling of Production Visualization Software on Diverse Architectures. *IEEE Computer Graphics and Applications (CG&A)*, 30(3):22–31, May/June 2010.

[5] 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.

[6] 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.

[7] 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.

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

[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. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(7):1009–1020, 2004.

[11] P. Gralka, S. Grottel, G. Reina, and T. Ertl. Application-specific compression of large md data preserving physical characteristics. In *IEEE Symposium on Large-Scale Data Analysis and Visualization (LDAV)*, pages 85–93. IEEE, 2013.

[12] 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.

[13] 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.

[14] A. Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3):331–371, 1910.

[15] 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.

[16] 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.

[17] J.-Y. Liang, C.-S. Chen, C.-H. Huang, and L. Liu. Lossless compression of medical images using hilbert space-filling curves. *Computerized Medical Imaging and Graphics*, 32(3):174–182, 2008.

[18] 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.

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

[20] G. M. Morton. *A computer oriented geodetic data base and a new technique in file sequencing*. International Business Machines Company, 1966.

[21] R. Olanda, M. Pérez, J. M. Orduña, and S. Rueda. Terrain data compression using wavelet-tiled pyramids for online 3d terrain visualization. *International Journal of Geographical Information Science*, 28(2):407–425, 2014.

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

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

[24] R. Sakai, D. Sasaki, S. Obayashi, and K. Nakahashi. Wavelet-based data compression for flow simulation on block-structured cartesian mesh. *International Journal for Numerical Methods in Fluids*, 73(5):462–476, 2013.

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

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

[27] 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.

[28] M. Treib, K. Burger, F. Reichl, C. Meneveau, A. Szalay, and R. Westermann. Turbulence visualization at the terascale on desktop pcs. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2169–2177, 2012.

[29] B. E. Usevitch. A tutorial on modern lossy wavelet image compression: foundations of jpeg 2000. *Signal Processing Magazine, IEEE*, 18(5):22–35, 2001.

[30] J. D. Villasenor, B. Belzer, and J. Liao. Wavelet filter evaluation for image compression. *IEEE Transactions on Image Processing*, 4(8):1053–1060, 1995.

[31] L. Williams. Pyramidal parametrics. In *ACM Siggraph Computer Graphics*, volume 17, pages 1–11. ACM, 1983.

[32] 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.

[33] 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 *IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, pages 31–38. IEEE, 2011.