Data Reduction Using Lossy Compression for Cosmology and Astrophysics Workflows

Jesus Pulido[1,2], Zarija Lukic[3], Paul Thorman[4], Caixia Zheng[5], James Ahrens[2], Bernd Hamann[1]

[1]Department of Computer Science, University of California, Davis, CA 95616, USA
[2]Los Alamos National Laboratory, P.O. Box 1663, Los Alamos, NM 87545, USA
[3]Lawrence Berkeley National Laboratory, 1 Cyclotron Rd, Berkeley, CA 94720, USA
[4]Haverford College, 370 Lancaster Avenue, Haverford, PA 19041, USA
[5]Northeast Normal University, 2555 Jingyue Street,Changchun 130117, China

E-mail: [1]jpulido@ucdavis.edu

Abstract. This paper concerns the use of compression methods applied to large scientific data. Specifically the paper addresses the effect of lossy compression on approximation error. Computer simulations, experiments and imaging technologies generate terabyte-scale datasets making necessary new approaches for compression coupled with data analysis. Lossless compression techniques compress data with no loss of information, but they generally do not produce a large-enough reduction when compared to lossy compression methods. Lossy multi-resolution compression techniques make it possible to compress large datasets significantly with small numerical error, preserving coherent features and statistical properties needed for analysis. Lossy data compression reduces I/O data transfer cost and makes it possible to store more data at higher temporal resolution. We present results obtained with lossy multi-resolution compression, with a focus on astrophysics datasets. Our results confirm that lossy data compression is capable of preserving data characteristics very well, even at extremely high degrees of compression.

1. Introduction

The scientific datasets generated today via computer simulations are too large for direct processing and analysis. Effective and efficient methods are becoming increasingly important to compress datasets, keeping in mind the need to preserve relevant scientific behavior in compressed representations as much as possible. Processing extraordinarily large datasets exposes limitations in current hardware, making data analysis challenging. As more powerful computer and imaging systems are being developed, these challenges grow exponentially. Currently, processing such large datasets introduces massive strain on data transfer and storage systems. On the observational side, one massive-dataset-generating system is the Large Synoptic Survey Telescope (LSST), a wide-field survey telescope currently under construction [1]. It is estimated to produce hundreds of gigabytes of raw data per night. Regarding computer simulations, the AMReX/Nyx computational cosmology code produces massive N-body and gas dynamics simulation outputs consisting of hundreds of terabytes of data for a typical large-scale run [2].

For these two specific applications, transferring and processing raw data creates many challenges as network and I/O (input/output) systems are strained. Previous efforts have

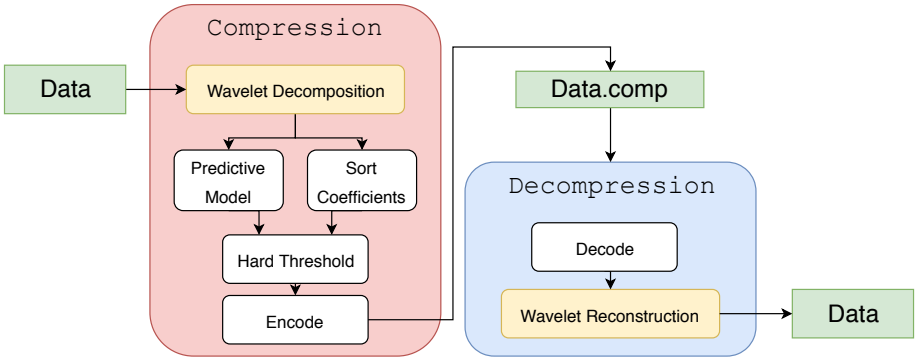

Figure 1. Compression and decompression pipeline.
Simplified pipeline shows data products (green), compression (red), and decompression (blue) operations.

utilized mainly lossless data compressors to reduce data size in memory before conducting I/O operations but they have generally not achieved significant data reduction [3]. Both transferring data between on-site systems and sharing data for collaborative research imposes significant bottlenecks, making adaptive lossy data reduction methods more viable. These benefits have been demonstrated in domains with similar data properties, such as simulations of turbulence, enabling collaborative remote visualization [4]. The consensus is that lossless compression methods are unable to reduce massive datasets sufficiently to offset I/O and memory bottlenecks.

As lossy methods have become necessary with the presented data challenge, it is important to understand the errors introduced by lossy data reduction for varying degrees of compression. We introduce a lossy multi-resolution compression framework and provide quantitative and qualitative results concerning error for regular-grid datasets, specifically for cosmology and astronomical image datasets.

2. Method

Lossy data compression can be achieved by selecting a wavelet basis, performing a data decomposition, then removing the lowest-magnitude coefficients via hard thresholding. This selection process can be either done with a predictive model or by sorting the coefficients. A hard threshold operation is performed by setting-to-zero those values that are below a certain coefficient magnitude, effectively throwing them away. The resulting coefficients are followed up by an encoding process to achieve additional amounts of data reduction by establishing several levels of numerical precision and applying dictionary-based methods for packaging. Decompression is a much simpler operation that simply decodes the data stream and reconstructs the remaining set of wavelet coefficients. Fig. 1 describes the complete compression and decompression pipeline.

2.1. Algorithms

It is possible to achieve high degrees of data reduction through discrete wavelet sampling. Wavelets are a generalization of the Fourier transform by using a basis that represents both location and spatial frequency [5]. A typical wavelet representation contains several vanishing moments that makes possible a sparse and highly accurate representation via dataset-sampling based on a small number of coefficients. The underlying basis functions support a discrete representation of a continuous signal, designed to produce responses for different frequencies. When applying a wavelet basis to a dataset, the result is a set of coefficients for that basis

function, capturing behavior at various resolutions. Wavelets that are commonly used include Haar wavelets, Daubechies wavelets, bi-orthogonal spline wavelets, and coiflet wavelets.

Pulido et al. [6] analyzed and compared in great detail many multi-resolution representation methods in the context of feature extraction and data analysis. Among those tested, higher-order B-spline wavelets were the most effective in terms of the ability to capture a broad range of quantities relevant for the representation of turbulence structures with a reduced set of coefficients. When viewing these results in the context of regular-grid and continuous datasets, the higher-order B-spline wavelet families consistently scored near the top of the compression schemes considered; therefore, biorthogonal cubic B-spline wavelets [7] were used for generating the results presented in this paper.

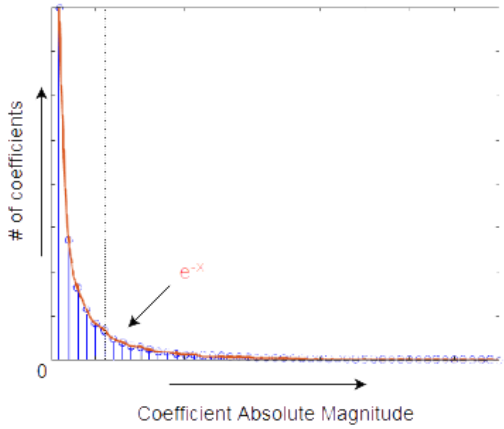

Figure 2. Absolute magnitude coefficient distribution.
An exponential decay behavior describes the relationship between coefficient magnitudes and number of coefficients. A small number of high-magnitude coefficients holding the most entropy can be kept while many small-valued coefficients can be removed to achieve effective data reduction.

Once a cubic B-spline wavelet signal is sampled via a regular-grid dataset, wavelet coefficients are produced using a dimension-by-dimension approach, i.e., in one step only one dimension is considered for the computation of coefficients. Although this method takes advantage of and is designed for regular structured grids, it can handle non-regular grids and simulated particle data by treating the data as a single, one-dimensional data stream. To preserve the most relevant features during lossy compression, one must keep coefficients with highest magnitudes and discard those with relatively very low magnitude. Datasets in many scientific domains produce a distribution of coefficients, i.e., a distribution of coefficients by magnitude, where the number of coefficients decreases as coefficient magnitude increases. This distribution can be modeled with an exponentially decreasing function, see Figure 2. Considering this behavior, a value for hard-thresholding must be selected. This can either be done by using a parallel quick-sort algorithm applied to the absolute magnitudes of the coefficients to select an exact value, or a predictive model for approximation for an understood specific scientific application.

When using a predictive model for approximation, significant time can be saved during compression by avoiding an expensive sorting step. By providing minimum, maximum and target compression values (% of coefficients or compression ratio), this model can be adjusted on-the-fly per application domain.

As final encoding step we perform data re-quantization and use an off-the-shelf dictionary compressor for packaging, e.g., LZ4 [8]. Wavelet coefficients are derived as floating-point,

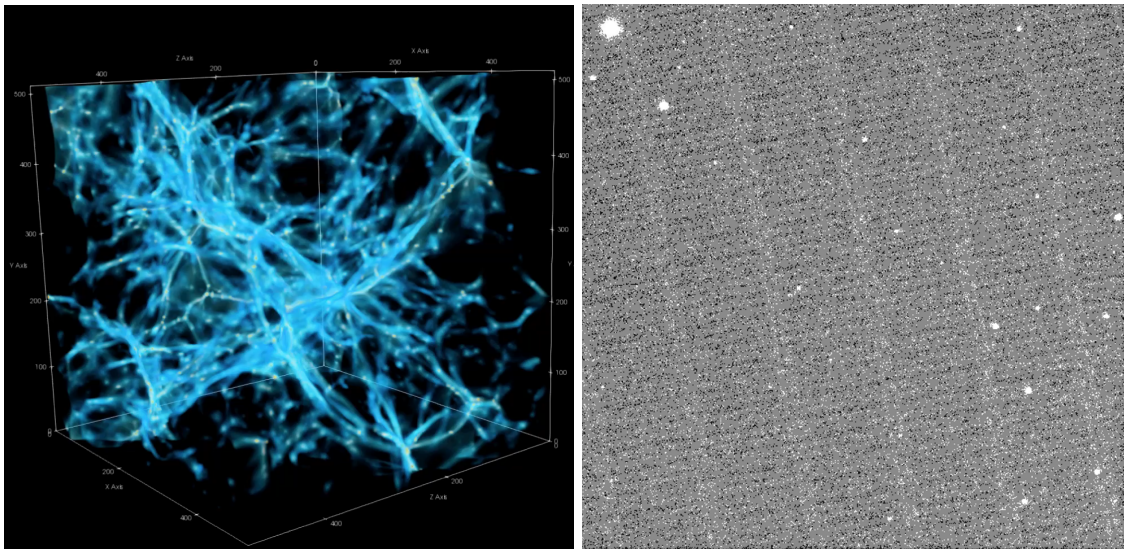

Figure 3. Cosmology and Astronomy features of interest.
3D-grid cosmology simulation showing baryon density volume (left) demonstrates clearly defined density clusters and interconnects to preserve during compression. A 2D-image sky-survey simulation (right) presents elliptical bright objects as stars and galaxies, combined with simulated noise.

numerical values – presenting an additional challenge when data is converted from an integer (astronomical images) to floating-point format. Traditionally, integer values lead to higher degrees of compression than floating-point values when using dictionary-based lossless compressors. By using floating-point-to-integer re-quantization, it is possible to preserve the majority of a wavelet coefficient's entropy at the cost of a small reduction in numerical precision. This is achieved by selecting a fixed number of significant digits and preserving them during the conversion to integer numerical space.

Decompression requires significantly less effort compared to compression. To retrieve compressed data, a file is decoded, re-quantized back to floating-point precision and reconstructed using a wavelet signal. A lossy representation of the original data is then retrieved.

2.2. Features and Implementation

Using wavelet-based methods provides several benefits such as de-noising, region-specific reconstruction (decompression), data streaming, and coefficient-space data analysis capability, without the need of full decompression. The primary component of this compressor uses a wavelet CPU implementation based on a heavily modified version of the GNU Scientific Library (GSL) [9]. Modifications were applied to the open-source library, including: support for 3D/N-D data, support for non-square and non-powers-of-two resolution data, higher-order B-spline basis function support, handling of boundary conditions, and parallel computation. An early open-source implementation of this compression code in both C++ and Matlab is available, see [10].

3. Results

3.1. Multi-dimensional Cosmology

The lead Nyx Lyman-alpha simulation significantly furthers the state-of-the-art in the field, using as many as 8192^3 grid cells. The simulation discussed in this paper for compression is a smaller, test simulation with 512^3 cells with six scalar components. The only difference between the lead and the test simulation is the physical size, which is 250 megaparsecs (Mpc) for the 8192^3 run and 15.625 Mpc for the smaller test case. The smaller simulation has the same physical resolution as the original, performed with the same physics and using the same choice of cosmological parameters: $h = 0.675$, $\Omega_m = 0.31$, $\Omega_\Lambda = 0.69$, $\Omega_b = 0.0487$, $\sigma_8 = 0.82$, $n_s = 0.965$, and $w_{\text{de}} = -1.0$.

The parallel, shared memory version of this compressor was tested on Intel's Haswell and Xeon Phi (KNL) architectures on Cori (NERSC, Lawrence Berkeley National Laboratory) to determine scaling performance. The two datasets have the same physical simulation box, but they vary in cell resolutions. For a physical grid size of 512^3, we found that a single Haswell node with 64 threads completes the main wavelet decomposition in 1.594 seconds on average, for a single scalar field of 512MB size. When using a resolution of 2048^3, the same routine requires 59.653 seconds on average for completetion, for a single scalar field of 32GB size. A single KNL node with 256 threads processes the 512^3 physical box in 2.334 seconds and a 2048^3 resolution in 88.491 seconds on average. These performance results are preliminary results, as it is still possible to further optimize our implementation.

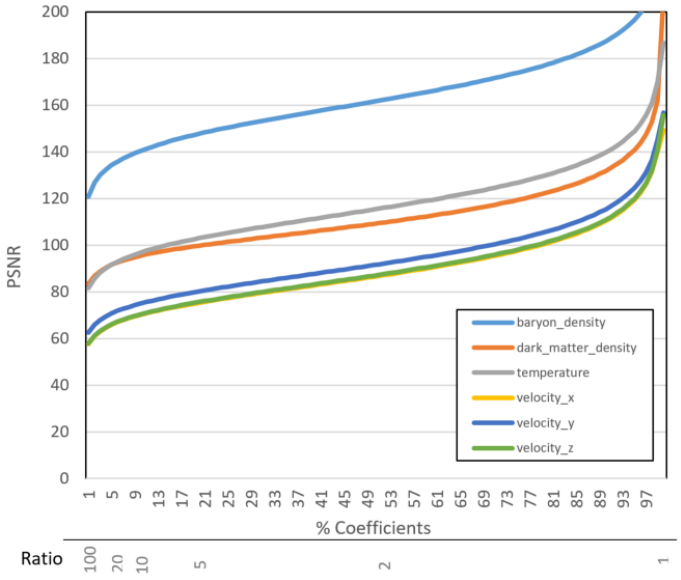

Figure 4. PSNR behavior on lossy compression.
PNSR comparisons show compression behavior across the six scalar fields; densities compressing the best compared to velocities. An inflection point is reached between 5%-7% with diminishing returns in quality. Velocity components compress similarly with a small positive bias in the y-axis.

Figure 4 shows the compression capabilities of the method, demonstrating a high degree of control over the quality of a lossy compression. Using the 512^3 dataset to represent the initial physical simulation box size, the peak-signal-to-noise (PSNR) plot shows the relative compression quality for each scalar component. Baryon density can be compressed the most,

while velocity components can be compressed the least. The y-direction velocity component can be compressed slightly more than the x- and z-directions. This fact could be the result of the specific initial conditions used in the simulation.

The same PSNR plot shows that generally all velocity components contain the same amount of error when compressed. The most important factor to consider is the observed inflection point that occurs for thresholding around 5-7 % coefficients (20:1 - 15:1 compression ratios). The inflection point generally signifies a "point of compromise"; using more coefficients for compression would only yield diminishing returns in quality. Therefore, 20:1 to 15:1 compression ratios are recommended for this specific cosmological dataset.

Figure 5 shows the progression from 1%, to 5%, and eventually 10% coefficients by rendering contours for dark matter density (logarithmic scale). For reference, a 10% visualization is "visually identical" to the original data. In our numerical results, compression degrees between 5% and 10% represent the inflection point. Visually, there is very little change from the latter two. By using 1% coefficients for compression (100:1), we see a deterioration of the contours in high-density regions.

Figure 6 shows the effect of different compression levels for the power spectra of dark matter density (left panel) and gas density (right panel). We see that the power spectra are in excellent agreement, especially considering that there is little cosmological interest in the $k > 10\,\mathrm{Mpc}^{-1}h$ regime. In particular, we observe that compression works extraordinarily well on baryon (gas) density. This is a physical explanation: Dark matter is pressure-less, thus forming a structure even on the smallest scales resolvable in a simulation. Gas, on the other hand, is described by the Euler equations, and is pressure-smoothed, with the smoothing scale depending on gas temperature. For this reason, density fluctuations are filtered on scales of a few cells, suppressing small structures and enabling a high degree of compression.

3.2. Astronomical Images

The LSST dataset [11] is a simulated image dataset that attempts to emulate observational features of the actual telescope, the LSST. Using GalSim, the lens properties and other characteristics of the telescope can be simulated to produce expected outputs for the future observational datasets. The sample set of FITS files are of size 4000 x 4000 containing several frequency bands. For sky survey simulations, noise is typically added to replicate characteristics of the actual telescope. Each pixel in an image represents about 0.2" of physical space, i.e., a single image represents up to 800" x 800" or 0.05 square degree depth of coverage.

As a pre-processing step, a grid-based de-noising scheme is used to remove the obvious observable artifacts, where otherwise a stacking the 2D image set would already remove them. Data compression and object detection can then be performed for this data.

Section 2.1 discusses the possibility of using alternative lossless compression schemes to encode coefficients of basis functions. To explore this issue, Figure 7 compares the use of stand-alone lossless compression and combinations of lossy compression with the methods we have introduced in this paper. We examine the use of traditional used Gzip [12] in standard FITS file compression, LZ4 [8], and BZip2 [13] as stand-alone compressors. Typically, computation times for a FITS file of this size vary from a few seconds to tens of seconds. When sorting methods from fastest to slowest, one obtains the following order: LZ4, GZIP, Wavelets with LZ4, Wavelets with BZip2, and BZip2.

In summary, we can employ lossy compression by preserving up to 50% wavelet coefficients and combine it with Bzip2 encoding to achieve file sizes as low as 9.8% of the original size. This result compares favorably relative to using stand-alone methods such as LZ4 (28%), Gzip (23.5%), and BZip2 (14.9%). More extreme levels of compression can be achieved by reducing the percentage of coefficients preserved, at the cost of numerical precision. Ideally, we would like to take advantage of fast decompression by pairing wavelets and LZ4, still producing compression

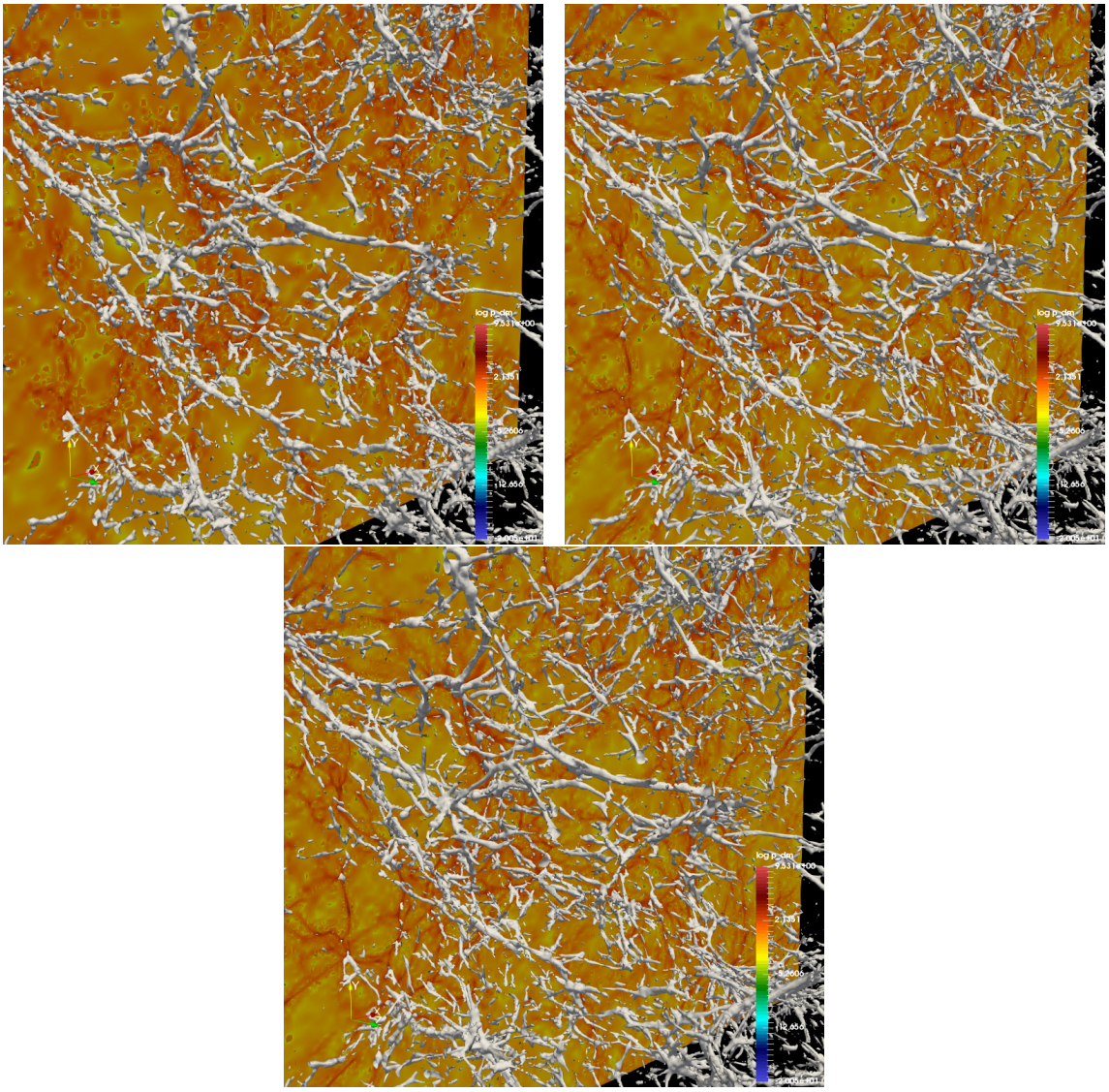

Figure 5. Dark matter density contour visualization comparisons.
Visual iso-contour structures can be compared at various compression levels. Left to Right: 1%, 5%, 10% coefficients. Areas of high-density clustering and general web-like structures are well preserved at high levels of compression. At extreme compression (1%), the web-like structure begins to deteriorate slightly.

sizes smaller than that obtained with traditional standalone Gzip. The choice of percentage of coefficients has the most impact on lossy compression quality.

A highly effective method for object detection by Zheng et al. [14] is used to evaluate the quality of lossy compression and object detection at various levels. In the original data, a total of 401 objects can be extracted with this method on a 64MB FITS container. By only using less than half of the wavelet coefficients (40%), it is possible to preserve at best 98% of the originally detected objects, with only 9.5% of the original file size. At more extreme levels, 90% object detection is still achievable when using only 15% of coefficients, effectively reducing the file size

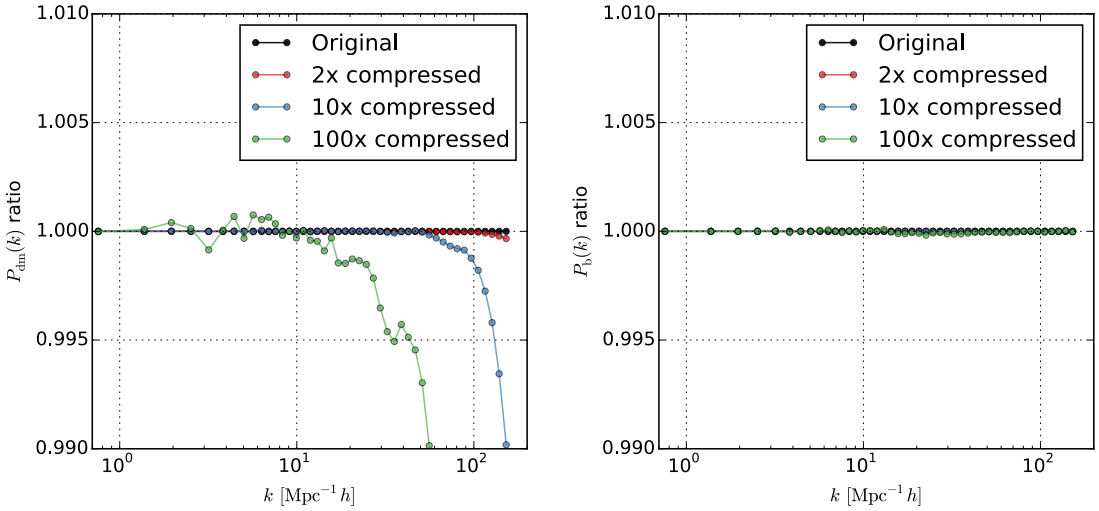

Figure 6. Nyx relative power spectra for different compression levels. Dark matter density (left) shows well-preserved energy ranges across all levels up to 10x compression, and low-power behavior up to 100x. Baryon density (right) shows excellent preservation even at extreme 100x compression.

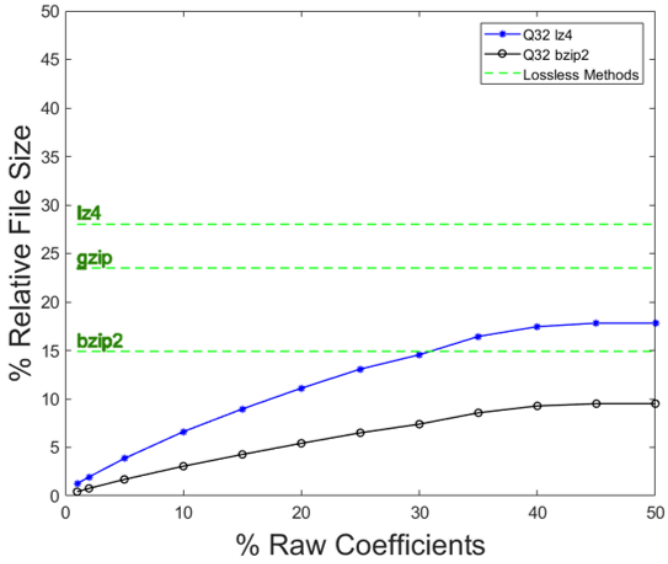

Figure 7. Comparison between compression methods. The dotted green lines show file sizes using three standard lossless compression methods while the others are lossy with wavelets (Q32). When lossy compression is used, data sizes can be significantly reduced by several magnitudes more.

to 5% of the original, see Figure 8 and Figure 9.

Figure 9 shows a comparison of the magnitudes of detected objects ("R_mag" in LSST catalog) for all ranges. These results show that compression at extreme levels affects the

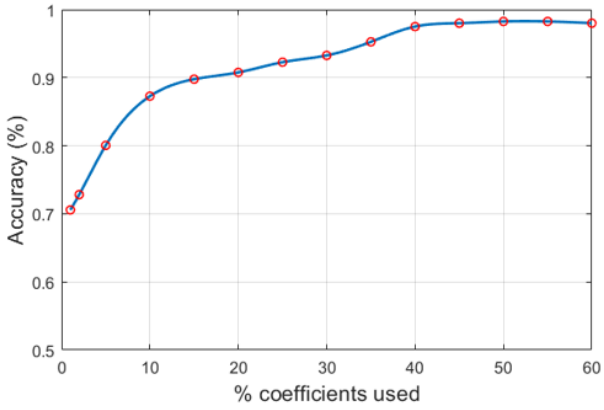

Figure 8. Object detection.
Trends in object detection show that using as little as 40% coefficients preserves over 98% of detected objects. Using an extreme data reduction that targets a 5% relative file size to the original, choosing to keep 15% coefficients can preserve 90% of detected objects.

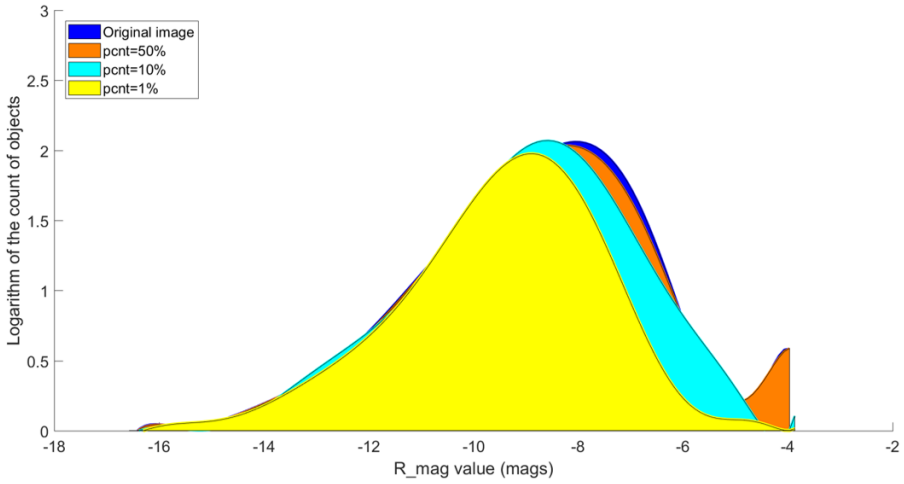

Figure 9. R_mag accuracy.
Coefficients by 1% (yellow), 10% (cyan), 50% (orange), and original (blue) show binned distributions of detected objects. A polynomial fit shows that bright and large objects are preserved well even at extreme compression percentages while smaller, usually fainter objects are lost.

detection of R_mag values representative of typically small objects. This result is consistent with reduced numerical precision, but it achieves better performance through the use of wavelets. Considering the representation with 50% coefficients, the R_mag ranges are nearly consistent with the originally detected objects, and even when using a representation with only 10% coefficients most properties are preserved.

4. Conclusions

We have demonstrated the capabilities of a lossy data compression approach and implementation for workflows arising in cosmology and astrophysics. Large degrees of data reduction are possible with the approach, without significantly compromising domain-specific features relevant for specific applications. For regular-grid astrophysics datasets, clustered density features are visually and numerically preserved, even at high levels of compression. Considering astronomical images, a majority of objects can still be detected and properly characterized even when high degrees of data reduction have been used.

We plan to conduct memory usage optimization tests for in-situ applications, where compression methods must be performed within a severely limited computation budget. The presented compression scheme is capable of performing temporal compression for 3D stacked image data or 4D simulated data. In order to better evaluate the computational capabilities of the compression approach, we need to perform strong and weak scaling tests, since computer systems are becoming increasingly parallel, and homogeneous architectures allow us to use both CPUs and GPUs for greatly accelerated computations.

Acknowledgments

We thank the DOE Exascale computing project for providing partial funding of this work at Los Alamos National Laboratory, under contract no. DE-AC52-06NA25396. We also thank the Argonne National Laboratory ALCF and Lawrence Berkeley National Laboratory NERSC groups for making their computer resources available for this project.

References

[1] Ivezić Ž and the LSST Science Collaboration 2013 Lsst science requirements document URL `http://ls.st/LPM-17`

[2] Almgren A S, Bell J B, Lijewski M J, Lukić Z and Van Andel E 2013 **765** 39 (*Preprint* `1301.4498`)

[3] Zeyen M, Ahrens J, Hagen H, Heitmann K and Habib S 2017 12–16

[4] Pulido J, Livescu D, Kanov K, Burns R C, Canada C, Ahrens J P and Hamann B 2018 *J. Parallel Distrib. Comput.* **120** 115–126

[5] Daubechies I 1992 *PA: SIAM*

[6] Pulido J, Livescu D, Woodring J, Ahrens J and Hamann B 2016 *Computers and Fluids* **125** 39 – 58 ISSN 0045-7930

[7] Cohen A, Daubechies I and Feauveau J C 1992 *Communications on Pure and Applied Mathematics* **45** 485–500

[8] Collet Y 2011 Lz4 - extremely fast compression URL `http://lz4.github.io/lz4/`

[9] Gough B 2009 *GNU Scientific Library Reference Manual - Third Edition* 3rd ed (Network Theory Ltd.) ISBN 0954612078, 9780954612078

[10] Pulido J 2018 Lossywave: A lossy, wavelet-based compressor for scientific simulation data. URL `https://github.com/lanl/VizAly-LossyWave/`

[11] 2014 Photon simulator (phosim) URL `https://www.lsst.org/scientists/simulations/phosim`

[12] GNU 1997 Gzip URL `https://www.gnu.org/software/gzip/`

[13] Steward J 1996 Bzip2 file compression URL `http://www.bzip.org`

[14] Zheng C, Pulido J, Thorman P and Hamann B 2015 *Monthly Notices of the Royal Astronomical Society* **451** 4445–4459 ISSN 0035-8711