

Accelerating In-transit Isosurface Generation With Topology Preserving Compression

Yanliang Li

University of Alabama at Birmingham
Birmingham, AL 35294
yli3@uab.edu

Jieyang Chen

University of Alabama at Birmingham
Birmingham, AL 35294
jchen3@uab.edu

Abstract—Data visualization through isosurface generation is critical in various scientific fields, including computational fluid dynamics, medical imaging, and geophysics. However, the high cost of data sharing between simulation sources and visualization resources poses a significant challenge. This paper introduces a novel framework that leverages lossy compression to accelerate in-transit isosurface generation. Our approach involves a Compressed Hierarchical Representation (CHR) and topology-preserving compression to ensure the fidelity of the isosurface generation. Experimental evaluations demonstrate that our framework can achieve up to 4x speedup in visualization workflows, making it a promising solution for real-time scientific data analysis.

I. INTRODUCTION

Data visualization such as *isosurface* generation is one of the most important routines that scientists use to extract insights from data. Isosurface generation is commonly used in several science domains such as computational fluid dynamics (CFD) [1], medical imaging [2], geophysics [3], and meteorology [4]. An isosurface represents a surface that consist of points of a constant *isovalue* within a volume of space. Given a domain significance value as *isovalue*, the position of an isosurface, as well as its relation to other neighboring isosurfaces, can provide clues to the underlying structure of a scalar field. As visualizations are commonly used for monitoring scientific simulations or experiments for scientists to steering their scientific workflows in time, it is essential to build efficient visualization workflows that enables near real-time (NRT) feedback for users.

One major challenge of building efficient visualization workflows is reducing the cost of data sharing between data source (such as computing nodes for simulation or science instruments for experiments) and computing resource for visualization, since it would prohibitively expensive to share via file-based I/O [5]. In-situ visualization that allow sharing data bypassing the filesystems to achieve much more responsive visualization has been proposed and developed for the past decade [6]. For example, in-transit based in-situ [7] that streams data via network to remote computing facilities for visualization is a more viable option for workflows runs on interconnected science instruments. However, with the advancement of scientific instruments and simulation codes, data are being generated with a dramatic increasing volume and velocity, which makes network-based data streaming the dominating bottleneck for in-transit visualization workflows.

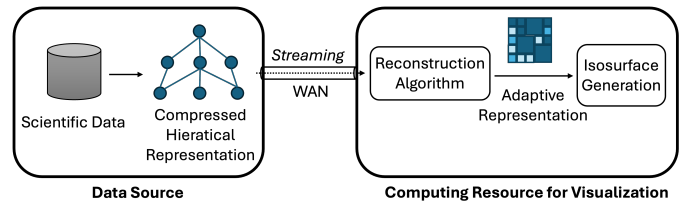


Fig. 1. Our accelerated in-transit isosurface generation framework

Recently, many data reduction tools designed for scientific data have been proposed that can greatly reduce the data volume. Among those, error-controlled lossy compression is one the most effective reduction tools since they provide high compression ratios while controlling the lossy of accuracy of reconstructed data. When reducing large scale datasets, lossy compression often offers more effective data volume reduction compared with lossless compression since not every bytes of the original data are preserved. To control the loss of information in data, several error controlled lossy compression methods have been proposed. Those compressors enable users to prescribe error bound at the compression time so that decompressed data respect such error bound relative to the original data. Depending on the reduction method used, these compressors can be classified as prediction-based (e.g., ISABELA [8], SZ [9], [10], and FPZIP [11]) and transform-based (e.g., ZFP [12] and MGARD [13]).

Despite many data reduction techniques have been proposed, not much work has been done to use lossy compression to accelerate in-transit visualization. So, in this work we aim to develop an novel accelerated in-transit visualization framework existing leveraging lossy compression. As the initial stage of the project, we focus on adapting and optimizing lossy compression for isosurface generation.

II. COMPRESSION ACCELERATED IN-TRANSIT ISOSURFACE GENERATION

In this work, we focus on adapting and optimizing modern lossy compressors to in-transit isosurface generation workflows. Figure 1 shows our compression accelerated in-transit isosurface generation framework. Left part represents the data source such as physical simulations running on large-scale high-performance computing (HPC) facilities or experimental instruments that generate data constantly. The data will be pre-processed and compressed into a Compressed Hierarchical Representation (CHR). On the visualization side (right), user

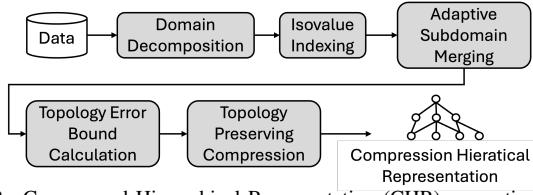


Fig. 2. Compressed Hierarchical Representation (CHR) generation process can use our reconstruction algorithm to request data need to generate isosurface with desired isovalue and accuracy.

A. Compressed Hierarchical Representation

Figure 2 illustrates the process of generating Compressed Hierarchical Representation (CHR). We first apply domain decomposition by dividing data into blocks isovalue based data pruning and differentiated error control. Then, to enable fast isosurface generation, we build index on blocks based on a series of candidate isovalues. Next, we identify blocks that can be merge into larger rectangular grids so that they can be compressed and visualized more efficiently. Finally, we apply our topology preserving compression to generate CHR.

B. Topology Preserving Compression

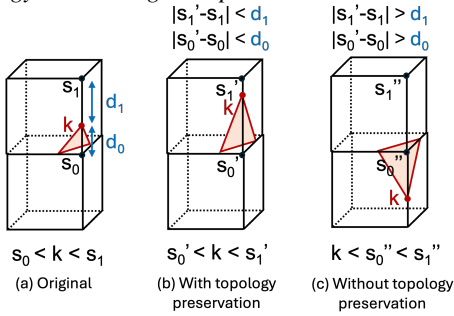


Fig. 3. Isosurface generation with and without topology preservation

We propose to control the accuracy of isosurface generation by translating the error of isosurface to the error of raw data so that we can leverage existing lossy compressors to compress data while preserve the fidelity of isosurfaces. To build such error translation, we first introduce the concept of *topology preservation* for isosurfaces. The isosurface generation process can be viewed as computing a surface out of a scalar field of the raw input data and the surface in each cell correspond to a certain case (i.e., cube configurations in term of Marching Cube [2]). So, we define topology preservation as guaranteeing that the same cases will be generated using decompressed data. Figure 3 illustrates the process of generating isosurface in a cell with $k = \text{isovalue}$. For the original data, k falls in between the value of two vertexes s_0 and s_1 . If we incur error to s_0 and s_1 , we might change the relationship between s_0 , s_1 , and k . On one hand, if the error is greater than the distance between k and its two vertexes (i.e., d_0 and d_1), then case might change for the cell. On the other hand, if the error is guaranteed to be smaller than both d_0 and d_1 , we can guarantee that the case would remains the same. So, to compute the error bound on raw data that preserve topology, we need to (1) find all edges (s_0, s_1) where $s_0 < k < s_1$; (2) compute d_0 and d_1 for all edges and save in an array D ; (3) the smallest value in D gives the absolute error bound for topology preserving compression. In addition, for cases where

some topology errors are acceptable (e.g., tolerate erroneous 5% cells), we can select the n^{th} smallest in D as error bound.

III. EXPERIMENTAL EVALUATION

We evaluate our framework on a compute node with Nvidia A100 GPUs on Jetstream 2 supercomputer [14] [15] at Indiana University. We use the temperature field data from NXY cosmological hydrodynamics simulation code [16] for compression and isosurface generation. We simulate the data streaming process by calculating the streaming cost of transferring the original or compressed data over a 1Gbps WAN. Figure 4 shows the end-to-end time breakdown of in-transit isosurface generation when using MGARD, SZ, and ZFP as compression backend. Comparing with the original workflow without compression, using 100% preserving compression can achieve up to $2.6\times$ performance improvement. For cases that we guarantee 99%, 95%, and 80% topology accuracy, we can achieve up to $3.8\times$, $3.8\times$, $4.0\times$ speedups.

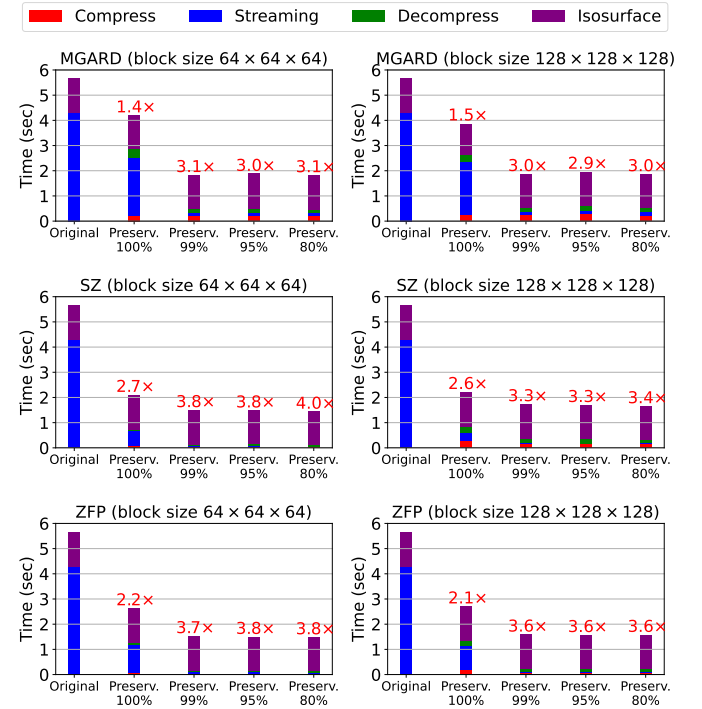


Fig. 4. End-to-end time breakdown of in-transit isosurface generation with and without our proposed topology preservation compression. Two block sizes (64^3 and 128^3) for domain decomposition are tested.

IV. CONCLUSION

This work proposed a compression accelerated data streaming framework for in-transit visualization that are commonly used in scientific workflows with interconnected instruments. Although many lossy compression have been proposed with error control, none of them have been specifically designed for controlling the error of visualization output such as isosurface. We enabled topology error control by translating the error in topology to error into raw data. With our proposed framework, we can achieve up to $2.6\times$ speedup for topology error free isosurface generation and $4\times$ speedup for with error control.

REFERENCES

- [1] C. Feltcher, "Computational techniques for fluid dynamics, 1," 1988.
- [2] L. WE, "Marching cubes: A high resolution 3d surface construction algorithm," *Computer graphics*, vol. 21, no. 1, pp. 7–12, 1987.
- [3] G. Dupuy, B. Jobard, S. Guillon, N. Keskes, and D. Komatitsch, "Isosurface extraction and interpretation on very large datasets in geophysics," in *Proceedings of the 2008 ACM symposium on Solid and physical modeling*, 2008, pp. 221–229.
- [4] M. Rautenhaus, G. Bauer, and A. Dörnbrack, "A web service based tool to plan atmospheric research flights," *Geoscientific Model Development*, vol. 5, no. 1, pp. 55–71, 2012.
- [5] F. Poeschel, J. E. W. F. Godoy, N. Podhorszki, S. Klasky, G. Eisenhauer, P. E. Davis, L. Wan, A. Gainaru, J. Gu *et al.*, "Transitioning from file-based hpc workflows to streaming data pipelines with openpmd and adios2," in *Smoky Mountains Computational Sciences and Engineering Conference*. Springer, 2021, pp. 99–118.
- [6] K.-L. Ma, "In situ visualization at extreme scale: Challenges and opportunities," *IEEE Computer Graphics and Applications*, vol. 29, no. 6, pp. 14–19, 2009.
- [7] J. C. Bennett, H. Abbasi, P.-T. Bremer, R. Grout, A. Gyulassy, T. Jin, S. Klasky, H. Kolla, M. Parashar, V. Pascucci *et al.*, "Combining in-situ and in-transit processing to enable extreme-scale scientific analysis," in *SC'12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. IEEE, 2012, pp. 1–9.
- [8] S. Lakshminarasimhan, N. Shah, S. Ethier, S.-H. Ku, C.-S. Chang, S. Klasky, R. Latham, R. Ross, and N. F. Samatova, "Isabela for effective in situ compression of scientific data," *Concurrency and Computation: Practice and Experience*, vol. 25, no. 4, pp. 524–540, 2013.
- [9] D. Tao, S. Di, Z. Chen, and F. Cappello, "Significantly improving lossy compression for scientific data sets based on multidimensional prediction and error-controlled quantization," in *2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 2017, pp. 1129–1139.
- [10] K. Zhao, S. Di, M. Dmitriev, T.-L. D. Tonellot, Z. Chen, and F. Cappello, "Optimizing error-bounded lossy compression for scientific data by dynamic spline interpolation," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2021, pp. 1643–1654.
- [11] P. Lindstrom and M. Isenburg, "Fast and efficient compression of floating-point data," *IEEE transactions on visualization and computer graphics*, vol. 12, no. 5, pp. 1245–1250, 2006.
- [12] P. Lindstrom, "Fixed-rate compressed floating-point arrays," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 2674–2683, 2014.
- [13] M. Ainsworth, O. Tugluk, B. Whitney, and S. Klasky, "Multilevel techniques for compression and reduction of scientific data—the univariate case," *Computing and Visualization in Science*, vol. 19, no. 5, pp. 65–76, 2018.
- [14] D. Y. Hancock, J. Fischer, J. M. Lowe, W. Snapp-Childs, M. Pierce, S. Marru, J. E. Coulter, M. Vaughn, B. Beck, N. Merchant *et al.*, "Jetstream2: Accelerating cloud computing via jetstream," in *Practice and Experience in Advanced Research Computing*, 2021, pp. 1–8.
- [15] T. J. Boerner, S. Deems, T. R. Furlani, S. L. Knuth, and J. Towns, "Access: Advancing innovation: Nsf's advanced cyberinfrastructure coordination ecosystem: Services & support," in *Practice and Experience in Advanced Research Computing*, 2023, pp. 173–176.
- [16] A. S. Almgren, J. B. Bell, M. J. Lijewski, Z. Lukić, and E. Van Andel, "Nyx: A massively parallel amr code for computational cosmology," *The Astrophysical Journal*, vol. 765, no. 1, p. 39, 2013.