Exploring Data Reduction Techniques for Additive Manufacturing Analysis

Coleman Nichols*, Megan Hickman Fulp*, Nathan DeBardeleben†, Jon C. Calhoun*
*Clemson University, Clemson, SC, USA
†Los Alamos National Laboratory, Los Alamos, NM, USA
cnicho5@clemson.edu, mlhickm@clemson.edu, ndebard@lanl.gov, jonccal@clemson.edu

***Abstract*—Additive manufacturing is a rapidly growing area that has the potential to revolutionize society. In order to better understand and improve this process, scientists and engineers conduct detailed studies on the applicability of various materials and the process that additively constructs the object. One method of analyzing the additive process is to use cameras to take images of the object as it is built layer by layer. As the complexity of the process, image resolution, and image capture frequency increases, so too does the volume of data generated, which can lead to data storage/movement issues. In this paper, we present an exploratory study of applying various lossless and lossy reduction techniques to an additive manufacturing data set from Los Alamos National Laboratory. Results show that SZ gives the best reduction ratio, ZFP yields the best accuracy, and Hybrid Data Sampling is the fastest method.**

***Index Terms*—lossy compression, lossless data compression big data, additive manufacturing**

I. Introduction

As technological advances (IoT, Cloud, AI, etc.) continue to be integrated in the industrial settings, we are on the verge of the fourth industrial revolution (Industry 4.0) [1]. In this new industrial age, data coming from manufacturing and industrial process is seamlessly processed, analyzed, and communicated. Thus, it enables more detailed insight and opens up new avenues and capabilities. One emerging area of Industry 4.0 is additive manufacturing [2].

Additive manufacturing is a construction process where a computer-aided design (CAD) model is constructed via applying very fine layers of material on a 2D plane [3], [4]. Each layer is laid on top of the previous and overtime the 3D object is reconstructed. 3D printing is a popular and common form of additive manufacturing that uses molten plastic to construct the object. However, AM is actively used in a wide variety of domains such as aerospace [5], medicine [6], and transportation [7]. Moreover, as AM continues to take hold, the market value is estimated to grow by 19.4% in North America along before the end of the decade [8].

In order for additive manufacturing to expand into new areas and to continue to improve the manufacturing process, engineers and scientists must understand what occurs throughout the process. To see inside the process, arrays of sensors and cameras are employed that capture data at a fine granularity. For example, a camera with an exposure duration of 33 ms generates 30 frames each second. Over the course of a day, this yields terabytes of raw data. Moreover, as the speed of the AM process increases, this frame rate causes blurring as the motion and positions of the applicator obscure fine features in the image, such as spatter. Spatter is the molten shower of material that is pushed from the application area and deposits itself in non-intended regions of the domain and may cause defects in the product [9]. Leveraging high-speed cameras with an exposure of 33 μsec dramatically increase the frame rate. This oncoming data deluge is complicated further with higher resolution cameras. Going forward, analysis of additive manufacturing processes must deal with large amounts of data generated at a high velocity.

Data reduction is a common technique to reduce the size of data, preventing data storage and transmission issues. Data reduction is broken down into two categories: lossy and lossless. In lossless reduction, the number of bytes requires storing the data is less, and after the resulting data after reconstruction is bitwise reproducible to the original data [10]. However, lossless reduction has been shown to only gives ˜1–5$\times$ reduction and does not perform well for floating-point datasets common in scientific computing [11]. Lossy reduction is able to obtain orders-of-magnitude larger reduction ratios, but at the expense of distortions in the data [12], [13]. Modern lossy compression algorithms such as SZ [11] and ZFP [14] enable the user to bound the level of distortion in the data. Other forms of lossy reduction such as data sampling yield higher reduction ratios and enable in-situ visualization without the overhead of decompression, but do not provide as robust of error bounding as SZ and ZFP.

With combined increase in image acquisition rates for additive manufacturing analysis and the resolution of these images, data reduction is needed for both long term storage and data transit. In this paper, we evaluate several data reduction techniques — e.g., lossless compression, data sampling, lossy compression — to determine their applicability to AM analysis data. In particular, this paper contributes the following:

- describes an emerging domain where data reduction is needed;
- presents a comparative analysis of state-of-the-art lossless, lossy compression and data sampling algorithms to be used to reduce additive manufacturing analysis data; and
- concludes that SZ gives the best reduction ratio, ZFP yields the best accuracy, and Hybrid Data Sampling is the fastest method.

The remainder of this paper is as follows: Section II provides additional background on additive manufacturing and the specific data reduction methods we use in this work. We describe our AM test dataset and preset our comparison methodology in Section III. Experimental results are found in Section IV. Finally, we conclude and present future work in Section VI.

II. Background

A. Adaptive Manufacturing Dataset

Additively manufactured parts, for a variety of reasons (complexity of designs, exotic alloys, newness of the technique, myriad of tunable parameters, etc.) often have small defects. These defects need to be analyzed to determine if they will impact the performance of the part when used. Both destructive — e.g., cutting up the part and analyzing defects with microscopy — and non-destructive — e.g. CT scan [15] — are commonly employed and generally focus on *voids* in the metal such as places where the metal did not fuse correctly.

In-situ analysis of additively manufactured parts would enable real-time defect analysis and could enable design modifications / adjustments of parts. This type of image data analytics for anomalies is common in machine learning. Thus, our problem comes down to capturing a rich training set at an extremely high rate, pushing the data in real time upstream to a cluster for storage, training on a massive dataset to create a model, pushing the model back to the edge for inference, and then using the model to detect anomalies in real time. This process of capturing large volumes of large files necessitates compression for storage and will undoubtedly impact machine learning accuracy. In general, we need to consider the speed of the compression algorithm (buffering may not be possible given the constant rate of data acquisition), compression rate, and reconstruction accuracy.

B. Data Reduction

In this subsection, we present background on the four data reduction algorithms we evaluate: Blosc (lossless), SZ (lossy), ZFP (lossy), and Hybrid Data Sampling (lossy).

1) BLOSC: https://www.blosc.org/pages/blosc-in-depth/

Blosc is a high performance lossless compression library [10]. Blosc decomposes the input data set into blocks to more easily apply various compression operations. Blosc enables user customization of per block filters and various lossless compression algorithms — e.g., zstd [16] — to be applied to obtain faster speeds or larger compression ratios. In this paper, we use the ztd compressor from Blosc as our lossless compression algorithm.

2) SZ: SZ is a lossy compression framework for scientific data that has seen rapid development over the recent years [11], [17]–[19]. The SZ compression pipeline is composed of four linearly dependent steps: data prediction, error quantization, Huffman coding, and lossless compression.

During the *data prediction* step of SZ, unprocessed data items are predicted based on already processed data items via various algorithms such as Lorenzo [17] or linear regression [18]. In general, the predicted datum does not exactly match the value from the original dataset. Storing the exact error as a floating-point value nets no compression. Instead, SZ maps the floating-point prediction error to an integer through a process known as *quantization* to improve the compression performance of subsequent steps. If the prediction is too far off to be successfully quantized, that datum is marked as unpredictable and compressed via a separate process [11].

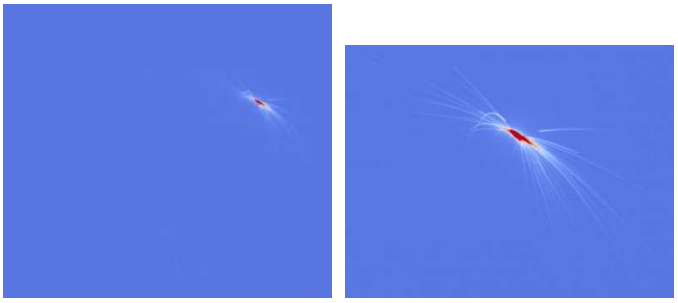
The quantization codes are combined to generate a Huffman tree that encodes the fixed-width code words into variable length code words to save space. Finally, all code words along with all necessary metadata for decompression is run through a lossless compression pass such as Gzip [20] or zSTD [16].

3) ZFP: Unlike SZ, which relies heavily on prediction, the ZFP lossy data compressor leverages transforms to decorrelate the data [14]. ZFP decomposes the data array into small 4^d blocks, where d is the number of dimensions of the data array. For each block, the data values are converted to a common fixed-point representation before a near-orthogonal transform is applied. To achieve a desired fixed-rate or a desired accuracy, the transformed data is then encoded by truncating the least significant bits of each value.

4) Hybrid Data Sampling: Unlike lossy compression, data sampling allows high levels of reduction at user specified reduction ratios. The simplest algorithm is Random Sampling, where a random number x_i is computed for each datum. Each x_i is compared to a sampling threshold α. If $x_i < \alpha$ then that datum is included in the set of samples, otherwise it is discarded. Systematic Sampling [21] selects every $\alpha \cdot N$ data points from a dataset of size N.

The previous sampling methods are dataset independent and often yield the lowest quality but the fastest reduction rate. Leveraging dataset properties allows more accurate preservation of regions-of-interest in the data. Value-Based Sampling [22] biases rare values in the dataset. Their algorithm uses the distribution of data values to calculate an importance factor, I_F, such that rare values have a greater I_F and common values have a lower I_F. The subsequent sampling process samples rare values before sampling common values. Multi-Criteria Based Sampling Algorithm [23] extends value based sampling by considers multiple criteria — e.g., rarity of the value and gradient magnitude — when constructing I_F. Hybrid Sampling [13] further extends Multi-Criteria Based Sampling by leveraging temporal similarities between time-steps. The algorithm first decomposes the domain into a series of blocks. The data within each block is compared to that from the previous time-step. Using a similarity metric such as histogram-intersection or root mean squared error, the algorithm determines if each block is similar enough. If it is, it creates a link to the previous data file. Otherwise, it elects to sample that block.

Sampling only keeps a subset of the data points and stores them in full precision. In order to recover the discarded data points, reconstruction algorithms such as nearest-neighbor or linear interpolation are needed. It is within the interpolated —



(a) Individual image from our AM dataset

(b) Zoomed in image of the dataset's feature.

Fig. 1: Example visualizations of our AM dataset. Red areas indicate where material is deposited. White arcing lines are spatter.

i.e., reconstructed – data points where distortion exists. Depending on the sampling rate and reconstruction algorithm utilized, the accuracy in the reconstruction differs.

III. Additive Manufacturing Data Compression

A. Description of the Additive Manufacturing Dataset

The dataset we leverage in our study comes from Los Alamos National laboratory. The dataset is captured on an EOS M 290 laser powder bed fusion (LPBF) metal printer. This process uses a high speed laser, metal alloy powder which is spread across a base, and a CAD-like design to additively manufacture (AM) metal parts. These parts are high quality, production-grade, and the AM process enables designers to construct parts which are difficult or impossible to build through conventional means. Attached to the printer is a PCO Edge 5.5 high speed camera that captures radiance values of the build during construction. For the experiment discussed in this paper, the camera is set to capture images at 30 frames per second with an exposure rate of 33ms. A custom driver program was written which controls the camera and saves the images as lossless 16-bit `TIF` images 25602160 (W$\times$H) resulting in approximately 11MB per image or 330MB per second. This amounts to roughly 28TB of raw data capture per day, and it is not uncommon for AM designs to run for over a day for a single build.

Figure 1a shows a visualization of a single time-step from our dataset. We see that the images contains a region of interest where the material is being deposited. Figure 1b shows a zoomed in view of the feature. In the zoomed image, the spatter shower becomes more visible. These fine features extend from the source in arcs across the image. The arcing trails are an artifact of the data collection process. With an exposure of 33ms, the motion paths are revealed. If using a camera with a shorter exposure time, the arcing artifacts diminish, leaving a single small point of molten material. Moreover, the location where the material is deposited appears larger than it would for the same reasons.

These data collection artifacts highlight the need to move to higher quality instruments to analyze the additive manufacturing process. Currently, at 11 MB per image generated at 30 frames/sec yields 330 MB/sec of data generated. An expected daily size of 28 TB must be mitigated for effective analysis and long term archival storage. In the remainder of this section, we detail the metrics and methodologies we employ to evaluate various data reduction methods on our AM dataset.

TABLE I: Key Terms and Notation for Data Reduction Metric Formulas

Term	Description
D	Original dataset
CD	Compressed dataset
ρ	Reduction ratio
$s(\cdot)$	Computes the size of a dataset
$\tilde{D}$	Reconstructed dataset of same size and dimension of D
β	Reduction bandwidth

B. Comparison Metrics

In this subsection, we describe the metrics we use to evaluate the data reduction algorithms in Section IV. Note that not all metrics are valid for the reduction algorithms. All lossless methods preserve accuracy. Thus, measurements of accuracy loss do not make sense. Moreover, depending on the configuration of the lossy methods they may fully preserve the data, mimicking lossless methods with respect to accuracy.

To aid in our discussion of the metrics, we define key terms and notations in Table I.

1) Reduction Ratio: All data reduction algorithms take in a dataset D with a fixed number of bytes $s(D)$. In general, after reduction, the size of the reduced data $s(CD)$ in bytes is less than that of the original dataset. We define the *Reduction Ratio*ρ, Equation 1, as the factor by which the dataset's size in bytes is reduced.

$$\rho = \frac{s(D)}{s(CD)} \tag{1}$$

For example, $\rho = 2$ indicates that the reduced dataset is 50% smaller than the original version. In general, larger values of ρ are preferred.

2) Reduction Bandwidth: The speed at which a dataset D is reduced is often dependent on the configuration of the method. For example, for lossless methods tend to take more time to reduce the dataset as the level of compression increases. However, lossy methods often reduce faster when they are allowed to add more distortions into the data. We define the *reduction bandwidth*, β, in Equation 2 as the time in seconds,t_{reduce}, to reduce a dataset of size $s(D)$.

$$\beta = \frac{s(D)}{t_{reduce}} \tag{2}$$

In general, large reduction bandwidth values are preferable.

3) Accuracy: Lossy reduction methods are capable of generating orders-of-magnitude levels of reduction. In order to obtain these high levels of reduction, distortions are introduced into the dataset. Thus, the effectiveness of lossy reduction methods depends on the perseveration accuracy *accuracy*

of the method. Error-bound reduction methods such as SZ and ZFP allow the user fine-grain control on accuracy by selecting an error bounding value and error bounding mode that is strictly enforced when reducing. Non-error bounding methods such as data sampling have coarse grained control. For example, the sampling method we evaluate [13], obtains better accuracy in the reconstructed data as the sampling percentage increases.

In this paper, we use the peak signal-to-noise ratio (PSNR) to quantify accuracy in units of decibels (dBs). PNSR is computed based on the ratio of the square of the maximal value in the original dataset, D, to the noise in the reconstructed dataset, $\tilde{D}$, measured using the mean squared error (MSE). The PSNR formula is given as:

$$PSNR = 10 \cdot log_{10} \frac{MAX(D)^2}{MSE(D, \tilde{D})} \tag{3}$$

Based on the formula for PSNR, as the accuracy in the dataset increases, the MSE deceases. As the MSE decreases, the calculated PSNR$\rightarrow \infty$. Similarly, as the level of distortion grows in the data and the MSE increases, the PSNR decreases.

IV. Experimental Results

A. Testing Environment

We run all experiments on Clemson's Palmetto Cluster. Each node we use contains the following hardware: 2 Intel Xeon Gold 6148 CPUs with a max clock frequency of 2.40GHz and 370 GB of DDR4 DRAM. To compile all our software, we employ gcc 8.5.0. The data compression algorithms are called through libPressio v0.79.0 [24], a compression abstraction library that contains numerous lossy and lossless compression algorithms. In particular, we use BLOSC 1.21.0, SZ 2.1.11.1, and ZFP 0.5.5. For data sampling, we use the code from [13]. As not all the methods we evaluate have optimized GPU or threaded versions, we report the single core performance. Future work will explore accelerated versions of the method(s) that are the most promising.

B. Configuring the Hybrid Data Sampling Algorithm

The Hybrid Data Sampling (HDS) algorithm from [13] seeks to preserve regions of interest that it determines exist in the dataset. It implements several variations on this algorithm. The first method (*Importance Based*) does not leverage temporal similarities and is equivalent to [23]. The second method uses histogram intersection (*Histogram Based*) to define spatial regions that are similar between time-steps, and requires each region to have identical histograms for reuse. The third variant (*Error Based*) uses the root mean squared error to determine reuse by comparing to a threshold of 10.

We explore various block sizes and to best configure each method. We find no difference in a method's accuracy/reduction bandwidth for our data set. We attribute this to the high degree of uniformity (constant background same histogram and no error) in the data, which results in a similar percentage of regions marked for reuse. Moreover, we explored sampling percentages of 1, 5 and 10%. We observed that as sampling percentage increased, PSNR and reduction bandwidth increased while reduction ratio decreased. This makes sense because if more data is sampled, then there should be less error; however, the sampling process should take longer and you are not able to reduce the size of the file as much.

Comparing the three HDS methods with respect to their accuracy over the time-steps in Figure 2, we see all methods preform similarly. Since our dataset contains large regions of similarity, the frequency of the background is high. Thus, Importance does not place many samples in this region. Moreover, this region is temporally smooth, leading to a large amount of data region reuse. Therefore, all methods concentrate sampling only on the region of interest and yield similar results.

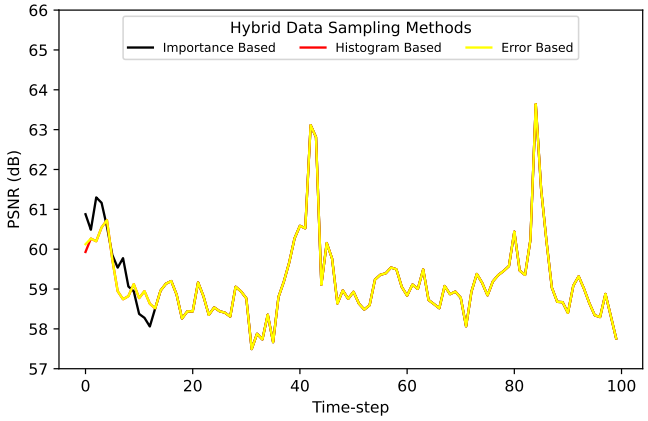


Fig. 2: Accuracy for various Hybrid Data Sampling methods with 5% sampling.

Both of the reuse methods are derived from the Importance based method. Figure 3 shows the sampling bandwidth for each sampling method. Importance is the most consistent, obtaining around 160 MB/s of bandwidth. For Error based reuse, the bandwidth drops after the first time-step due to the reuse calculation adding additional work on the critical path. The most interesting HDS method is Histogram based reuse, as it oscillates between 100–300 MB/s. The oscillation is due to the fact that a region is only reusable for the next time-step. Reusing regions is faster than having to sample a region. Examining the percentage of blocks we reuse, we see the same oscillatory pattern.

From Figure 2 and Figure 3, we elect to use the Importance based sampler because it yields consistent and high performance compared to the other two methods.

C. Reduction Bandwidth

The additive manufacturing process may consume a significant amount of time. With our camera based analysis method, data in constantly being generated and must be reduced. In a production setting, the data is reduced in situ before it is stored or analyzed in real-time. Because not all the reduction methods we evaluate have threaded or accelerator support, we focus on single-threaded performance.

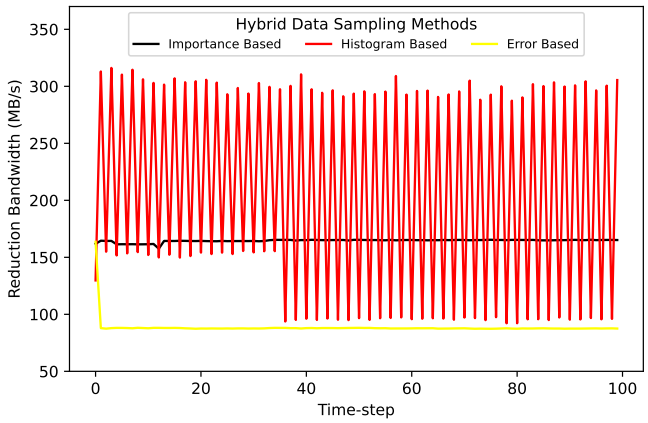


Fig. 3: Reduction Bandwidth for various Hybrid Data Sampling methods with 5% samples. Fluctuations in bandwidth are due to the overhead and savings in the reuse algorithms.

Before we compare all the methods, we explore the reduction bandwidth of the SZ and ZFP lossy compressors because they allow the user to set an error bound to control accuracy in the data. Figure 4 explores the impact of the error bound on reduction bandwidth. We see that as we require the compressor to yield more accurate data (smaller error bound) the bandwidth decreases. However, as the error bound increases allowing the compressor to add in more distortion, bandwidth increases. Here, we select the error bound of 0.01 for our comparison, as it is the closest configuration that achieves a similar level of accuracy as HDS (see Figure 8). We observe that SZ has a slightly lower reduction bandwidth for most of our tested error bounds. This is most likely due to the fact that part of SZ's process is determining the optimal quantization method. This process is not performed in ZFP and therefore could be slightly slowing SZ down.

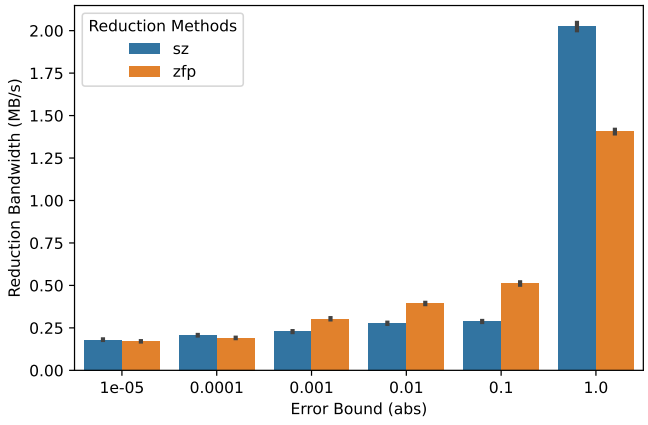


Fig. 4: Impact of absolute error bound value on reduction bandwidth.

Figure 5 shows the reduction bandwidth of our methods over the 100 time-steps. We see that for each method, its reduction bandwidth remains consist over the time-steps. This is due to the high degree of similarity in each image; only the location and shape of the feature differs. When comparing each method, HDS yield the largest bandwidth at 160 MB/s. ZFP and SZ yield 0.4 MB/s and 0.24 MB/s, respectively. Finally, BLOSC obtains the lowest bandwidth with 0.0025 MB/s. Our experimental setup generates 330 MB/s of data. With a single thread, HDS is able to process the data without a backlog or the need for threading. Results on other datasets show that threaded and accelerated versions of SZ and ZFP are more than capable of sufficient bandwidths [25]. These results seem to be fairly constant. All of our test were run on one private node that had no other tasks to handle. This is likely due to this consistency across all time-steps.

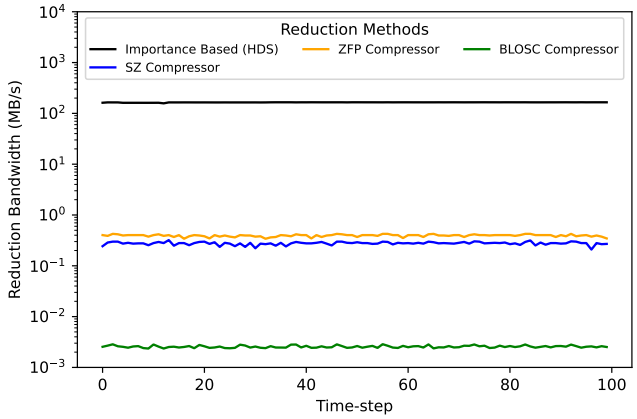


Fig. 5: Evolution of the single-threaded reduction bandwidth of various reduction methods on our additive manufacturing dataset over time.

D. Compressibility

To determine which reduction method gives the best level of reduction, we run each method over our dataset and present the average reduction ratios in Figure 6. For the error-bounded lossy compressors, we vary the absolute error bound between 1e-5 and 1. As SZ's and ZFP's error bound grows, the compression ratio increases, and we see a maximum of 502690× for SZ and 510× for ZFP. In the case of SZ, the reduction ration increases by more than 4 orders-of-magnitude over the error bound range. Comparing the lossy compressors to HDS and BLOSC we see various trade off points. HDS yield a reduction ratio of 20× based on its 5% sampling rate. BLOSC achieves a reduction ratio of 7.05×. For error bounds of less than 1e-3, BLOSC is better at reducing the data's size than ZFP. Overall, we see that SZ is the clear winner, as it yields a reduction ratio 200× more than its closest competitor ZFP at error bound 0.01.

E. Reconstruction Accuracy

The major advantage of lossless reduction methods is that they perfectly preserve the data. As such, we exclude BLOSC from this subsection. Lossy methods trade distortions in the

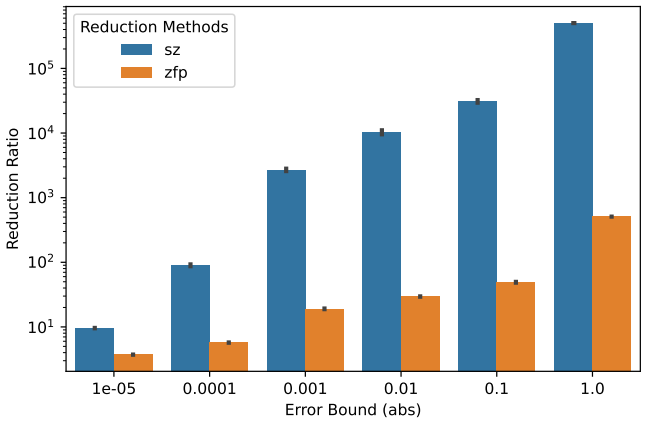


Fig. 6: Impact of absolute error bound value on reduction ratio.

data for larger reduction ratios (see Figure 6). Figure 7 shows the average PSNR for various error bounds for SZ and ZFP. As the error bound decreases, the PSNR increases for both compressors. For all error bounds, ZFP yields a PSNR that is between 2–21 dB larger than SZ. However, considering that SZ reduces the data's size 200× more than ZFP, the small decrease in accuracy is acceptable. These results make sense, because ZFP is optimized for 3D datasets and we are working with 2D image data. Also, the maximum compression error from SZ tends to be extremely close to the user's specified error bound; however, ZFP always over-preserves the compression error compared to the user's specified error bound. This leads to ZFP having a slightly higher accuracy but a slightly lower reduction ratio compared to SZ.

Figure 8 shows the variation in the PSNR over the time-steps for SZ, ZFP, and HDS. For SZ and ZFP, we use an absolute error bound of 0.01. We see that HDS with a 5% sampling rate yield lower PNSR than both SZ and ZFP, but is more consistent over time. Comparing SZ and ZFP, we see that ZFP gives the highest PSNR, except for a few time-steps between time-step 10 and 35.

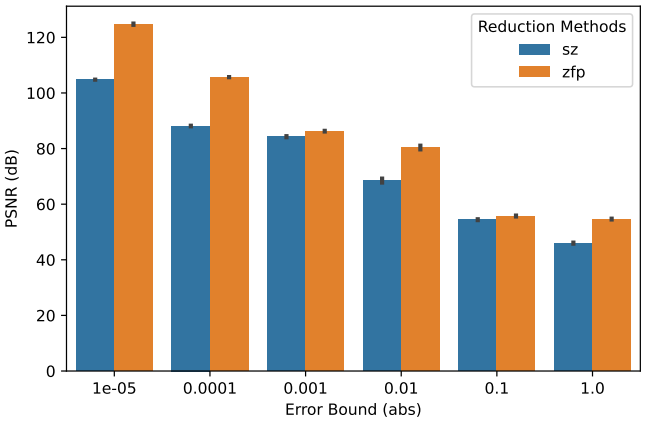


Fig. 7: Impact of absolute error bound value on the PSNR.

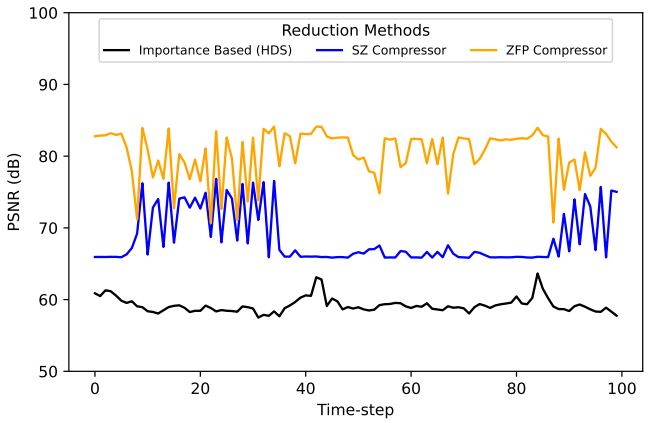


Fig. 8: Evolution of the PSNR of various reduction methods on our AM dataset over time.

V. Related Work

Large-scale scientific instruments and applications generate massive quantities of data. The Community Earth System Model (CESM) can generate over 2.5 PB of data in 18 months [26], but due to I/O and processing issues only 6.6% of it can be analyzed in the next 18 months. Moreover, the Hardware/Hybrid Accelerated Cosmology Code (HACC) [27] can produce 21.2 petabytes of data when simulating 2 trillion particles for 500 time-steps.

To combat the big data deluge, lossy reduction techniques have saw swift development over the last decade [11], [13], [14], [17]–[19], [25], [28]. Moreover, work has been ongoing to integrate lossy compression into applications and workflows [29]–[33]. This paper complements these prior works by exploring lossy data compression for nondestructive AM analysis.

VI. Conclusions

As additive manufacturing continues to grow in importance, improving the process become increasingly vital. One method to analyze AM is to use high-speed cameras, but must confront a data deluge (TBs/day) to be practical. In this paper, we explore four methods of data reduction: BLOSC (lossless), SZ (lossy), ZFP (lossy), and Hybrid Data Sampling (lossy). Results show that BLOSC yield no loss in accuracy, but is slow and does not reduce the size of data as well as the lossy methods. Hybrid Data Sampling gives the best bandwidth and fixed reduction sizes, but the lowest PSNRs. ZFP yields the best PNSR and a comparable reduction bandwidth to SZ. SZ provides the best reduction ratio over the second best ZFP. SZ also yields comparable bandwidth and PSNR to ZFP. Based on the analysis, we conclude that SZ is currently the best compressor for our AM dataset. For future work, we will explore how the inaccuracies from lossy compression impact the quality of the analytics.

ACKNOWLEDGMENTS

Clemson University is acknowledged for generous allotment of compute time on the Palmetto cluster. This material is based upon work supported by the National Science Foundation under Grant No. SHF-1910197 and SHF-1943114. The data is provided by Los Alamos National Laboratory: LA-UR-21-32202. This paper has been approved by Los Alamos National Laboratory: LA-UR-22-29693.

REFERENCES

[1] "What is industry 4.0?" https://mep.purdue.edu/news-folder/what-is-industry-4-0/, accessed: 2022-08-21.

[2] I. Gibson, D. Rosen, B. Stucker, and M. Khorasani, *Additive Manufacturing Technologies*. Springer International Publishing, 2021.

[3] W. E. Frazier, "Metal additive manufacturing: A review," *Journal of Materials Engineering and Performance*, vol. 23, no. 6, pp. 1917–1928, Jun 2014. [Online]. Available: https://doi.org/10.1007/s11665-014-0958-z

[4] I. Gibson, D. Rosen, B. Stucker, and M. Khorasani, "Introduction and basic principles," in *Additive Manufacturing Technologies*. Springer International Publishing, nov 2020, pp. 1–21.

[5] B. Blakey-Milner, P. Gradl, G. Snedden, M. Brooks, J. Pitot, E. Lopez, M. Leary, F. Berto, and A. du Plessis, "Metal additive manufacturing in aerospace: A review," *Materials & Design*, vol. 209, p. 110008, 2021.

[6] C. Culmone, G. Smit, and P. Breedveld, "Additive manufacturing of medical instruments: A state-of-the-art review," *Additive manufacturing*, vol. 27, pp. 461–473, 2019.

[7] M. Akbari and N. Ha, "Impact of additive manufacturing on the vietnamese transportation industry: An exploratory study," *The Asian Journal of Shipping and Logistics*, vol. 36, no. 2, pp. 78–88, 2020.

[8] G. V. Research, "Gvr report cover additive manufacturing market size, share & trends analysis report by component, by printer type, by technology, by software, by application, by vertical, by material, by region, and segment forecasts, 2022 - 2030," Grand View Research, Tech. Rep., 2022.

[9] Z. A. Young, Q. Guo, N. D. Parab, C. Zhao, M. Qu, L. I. Escano, K. Fezzaa, W. Everhart, T. Sun, and L. Chen, "Types of spatter and their features and formation mechanisms in laser powder bed fusion additive manufacturing process," *Additive Manufacturing*, vol. 36, p. 101438, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214860420308101

[10] "What is blosc?" https://www.blosc.org/pages/blosc-in-depth/, accessed: 2022-08-21.

[11] S. Di and F. Cappello, "Fast error-bounded lossy hpc data compression with sz," in *2016 ieee international parallel and distributed processing symposium (ipdps)*. IEEE, 2016, pp. 730–739.

[12] F. Cappello, S. Di, S. Li, X. Liang, A. M. Gok, D. Tao, C. H. Yoon, X.-C. Wu, Y. Alexeev, and F. T. Chong, "Use cases of lossy compression for floating-point data in scientific data sets," *The International Journal of High Performance Computing Applications*, vol. 33, no. 6, pp. 1201–1220, 2019. [Online]. Available: https://doi.org/10.1177/1094342019853336

[13] M. H. Fulp, A. Biswas, and J. C. Calhoun, "Combining spatial and temporal properties for improvements in data reduction," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 2654–2663.

[14] P. Lindstrom, "Fixed-rate compressed floating-point arrays," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2674–2683, Dec 2014.

[15] A. du Plessis, S. G. le Roux, J. Els, G. Booysen, and D. C. Blaine, "Application of microct to the non-destructive testing of an additive manufactured titanium component," *Case Studies in Nondestructive Testing and Evaluation*, vol. 4, pp. 1–7, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214657115000155

[16] Y. Collet and M. Kucherawy, "Zstandard Compression and the application/zstd Media Type," RFC 8478, Oct. 2018. [Online]. Available: https://rfc-editor.org/rfc/rfc8478.txt

[17] 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 2017, Orlando, FL, USA, May 29 - June 2, 2017*, 2017, pp. 1129–1139. [Online]. Available: https://doi.org/10.1109/IPDPS.2017.115

[18] X. Liang, S. Di, D. Tao, S. Li, S. Li, H. Guo, Z. Chen, and F. Cappello, "Error-controlled lossy compression optimized for high compression ratios of scientific datasets," in *IEEE International Conference on Big Data, Big Data 2018, Seattle, WA, USA, December 10-13, 2018*, 2018, pp. 438–447. [Online]. Available: https://doi.org/10.1109/BigData.2018.8622520

[19] X. Liang, K. Zhao, S. Di, S. Li, R. Underwood, A. M. Gok, J. Tian, J. Deng, J. C. Calhoun, D. Tao, Z. Chen, and F. Cappello, "Sz3: A modular framework for composing prediction-based error-bounded lossy compressors," *IEEE Transactions on Big Data*, pp. 1–14, 2022.

[20] P. Deutsch, "Gzip file format specification version 4.3," United States, Tech. Rep., 1996.

[21] F. Yates, "Systematic sampling," *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, vol. 241, no. 834, pp. 345–377, 1948.

[22] A. Biswas, S. Dutta, J. Pulido, and J. Ahrens, "In situ data-driven adaptive sampling for large-scale simulation data summarization," in *Proceedings of the Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization - ISAV '18*. ACM Press, 2018, p. 13–18. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3281464.3281467

[23] A. Biswas, S. Dutta, E. Lawrence, J. Patchett, J. C. Calhoun, and J. Ahrens, "Probabilistic data-driven sampling via multi-criteria importance analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 12, pp. 4439–4454, 2021.

[24] R. Underwood, V. Malvoso, J. C. Calhoun, S. Di, and F. Cappello, "Productive and performant generic lossy data compression with libpressio," in *2021 7th International Workshop on Data Analysis and Reduction for Big Scientific Data (DRBSD-7)*. IEEE, 2021, pp. 1–10.

[25] J. Tian, S. Di, K. Zhao, C. Rivera, M. H. Fulp, R. Underwood, S. Jin, X. Liang, J. Calhoun, D. Tao, and F. Cappello, "Cusz: An efficient gpu-based error-bounded lossy compression framework for scientific data," in *Proceedings of the ACM International Conference on Parallel Architectures and Compilation Techniques*, ser. PACT 2020. New York, NY, USA: Association for Computing Machinery, 2020, p. 3–15. [Online]. Available: https://doi.org/10.1145/3410463.3414624

[26] S. Mickelson, A. Bertini, G. Strand, K. Paul, E. Nienhouse, J. Dennis, and M. Vertenstein, "A new end-to-end workflow for the community earth system model (version 2.0) for the coupled model intercomparison project phase 6 (cmip6)," *Geoscientific Model Development*, vol. 13, no. 11, pp. 5567–5581, 2020.

[27] S. Habib, V. Morozov, N. Frontiere, H. Finkel, A. Pope, K. Heitmann, K. Kumaran, V. Vishwanath, T. Peterka, J. Insley *et al.*, "HACC: Extreme scaling and performance across diverse architectures," *Communications of the ACM*, vol. 60, no. 1, pp. 97–104, 2016.

[28] M. Ainsworth, O. Tugluk, B. Whitney, and S. Klasky, "Multilevel techniques for compression and reduction of scientific data—the multivariate case," vol. 41, pp. A1278–A1303.

[29] A. Fox, J. Diffenderfer, J. Hittinger, G. Sanders, and P. Lindstrom, "Stability Analysis of Inline ZFP Compression for Floating-Point Data in Iterative Methods," vol. 42, no. 5, pp. A2701–A2730, 2020. [Online]. Available: https://epubs.siam.org/doi/10.1137/19M126904X

[30] D. M. Hammerling, A. H. Baker, A. Pinard, and P. Lindstrom, "A collaborative effort to improve lossy compression methods for climate data," in *2019 IEEE/ACM 5th International Workshop on Data Analysis and Reduction for Big Scientific Data (DRBSD-5)*, 2019, pp. 16–22.

[31] X.-C. Wu, S. Di, E. M. Dasgupta, F. Cappello, Y. Alexeev, H. Finkel, and F. T. Chong, "Full state quantum circuit simulation by using data compression," in *IEEE/ACM 30th The International Conference for High Performance computing, Networking, Storage and Analysis (IEEE/ACM SC2019)*, 2019, pp. 1–12.

[32] J. Calhoun, F. Cappello, L. N. Olson, M. Snir, and W. D. Gropp, "Exploring the feasibility of lossy compression for pde simulations," *The International Journal of High Performance Computing Applications*, vol. 33, no. 2, pp. 397–410, 2019. [Online]. Available: https://doi.org/10.1177/1094342018762036

[33] A. H. Baker, H. Xu, D. M. Hammerling, S. Li, and J. P. Clyne, "Toward a multi-method approach: Lossy data compression for climate simulation

data," in *High Performance Computing*, ser. Lecture Notes in Computer Science, J. M. Kunkel, R. Yokota, M. Taufer, and J. Shalf, Eds. Springer International Publishing, 2017, vol. 10524, pp. 30–42.