Lossy and Lossless Compression for BioFilm Optical Coherence Tomography (OCT)

Max H. Faykus III
Clemson University
Clemson, South Carolina, USA
mfaykus@g.clemson.edu

Melissa C. Smith
Clemson University
Clemson, South Carolina, USA
mcsmith@clemson.edu

Jon C. Calhoun
Clemson University
Clemson, South Carolina, USA
jonccal@clemson.edu

ABSTRACT

Optical Coherence Tomography (OCT) is a fast and non-destructive technology for bacterial biofilm imaging. However, OCT generates approximately 100 GB per flow cell, which complicates storage and data sharing. Data reduction reduces data complications by reducing overhead and the amount of data transferred. This work leverages the similarities between layers of OCT images to minimize data in order to improve compression. This paper evaluates 5 lossless and 2 lossy state-of-the-art compressors as well as 2 pre-processing techniques to reduce OCT data. Reduction techniques are evaluated to determine which compressor has the most significant compression ratio while maintaining a strong bandwidth and minimal image distortion. Results show SZ with frame before pre-processing is able to achieve the highest CR of 204.6× on its higher error bounds. The maximum compression bandwidth SZ on higher error bounds is $\sim 41MB/s$, for decompression bandwidth, it is able to outperform ZFP achieving $\sim 67MB/s$.

CCS CONCEPTS

• **Information systems → Data compression**; *Data layout*.

KEYWORDS

lossy compression, lossless compression, optical coherence tomography

ACM Reference Format:
Max H. Faykus III, Melissa C. Smith, and Jon C. Calhoun. 2023. Lossy and Lossless Compression for BioFilm Optical Coherence Tomography (OCT). In *Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis (SC-W 2023), November 12–17, 2023, Denver, CO, USA*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3624062.3625125

1 INTRODUCTION

Optical coherence tomography (OCT) is a fast and non-destructive imaging technology that captures 3D morphology of the samples [15]. This method allows generated biofilm to be examined while not requiring any staining or destruction of microorganisms. OCT provides high-resolution depth-resolved images in the mesoscope to macroscopic ranges. This is useful for biological and non-biological contactless non-destructive testing [23]. High throughput OCT measurements, which are generated in a data stream, reaches up to one trillion bits per second [19]. Other methods require cutting and staining the sample to put under a magnifying glass to see. Electron microscopes analyze samples on a 2D plane, however, these lack depth information. OCT solves these obstacles, by viewing the data from refracting light off the surface. A consequence of this method is OCT generates large volumes of data.

Raw data storage accrues a high cost depending on the data center utilized and the need for high-performance servers to analyze data. To solve this problem, data reduction is a technique which is utilized to reduce the size of data in order to lower the footprint of required storage and improve data transmission. Data compression is an effective form of data reduction by helping to solve issues related to I/O bottlenecks and limited storage space on HPC systems [22]. There are two types of data compression, lossless compression and lossy compression. For lossless compression, the data before and after compression is byte for byte precisely the same. The data stored using lossless has less of a storage footprint than the original data. The disadvantage of lossless compression is the overall compressibility of floating point data. Lossy compression is able to achieve a much higher compression ratio when compared to lossless, but this comes at the expense of data distortion. The level of distortion is set by an error bound such as SZ [17, 24] and ZFP [18].

Standard compressors perform a generic algorithm from off-the-shelf compressors that do not leverage the 3D nature of OCT data. This leveraging of images allows improvement in the level of compression. Data reduction is needed for long-term storage and for data transfer to clusters for analysis [21]. For example, storage costs approximately \$0.022 per GB. Reducing overall footprints of the data reduces overhead costs of storing information.

This paper uses OCT biofilm data to determine which compressor and pre-processing method is most effective in compressing the data in a timely and accurate manner. Evaluating different methods of data reduction, we analyze lossless and lossy methods of data compression on OCT biofilm data. This paper contributes the following

- Comparative analysis of 5 lossless and 2 lossy state-of-the-art compressors to reduce biofilm OCT data.
- Lossless methods, zstd provides the best compression ratio 4.5× and lz4 has the best compression bandwidth $73MB/s$.
- In lossy methods, SZ gives the best compression ratio 204.6× , ZFP provides the overall highest compression $40MB/s$ and $48MB/s$ decompression bandwidth.

SC-W 2023, November 12–17, 2023, Denver, CO, USA

ACM ISBN 979-8-4007-0785-8/23/11.
https://doi.org/10.1145/3624062.3625125

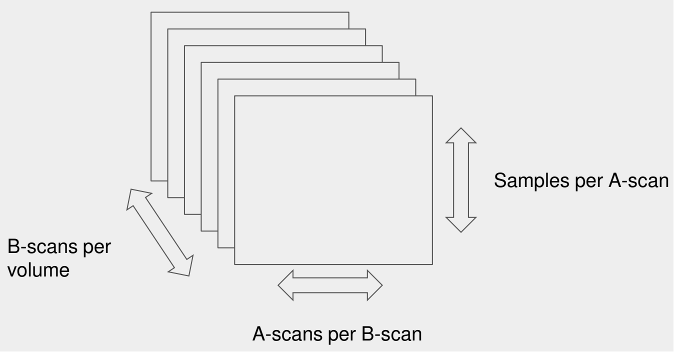


Figure 1: OCT image structure

- Developing a data pre-processing pass that leverages spatial similarity in OCT data to improve the compression ratio by a maximum of 33.98% on the higher error bounds of SZ.

2 BACKGROUND

2.1 Dataset

The OCT data we use is generated by a Thorlabs Ganymede commercial OCT system [1]. An OCT 3D image consists of a sequence of 2D images, each of which represents one slice that contains a large volume of information about the Biofilm [8]. An example is illustrated in Figure 1. The data generated from the system is approximately 100GB per biofilm sample which is stored in a flow cell and is broken down into 12 volumes of 8GB each. The combined raw 3D image file format consists of a 12-bit depth and an overall volume of $K \times J \times Z$. This was created with 2048 representing samples per A-scan, 1000 A-scans per B-scan, and 250 B-scans shown in Figure 1. Data from thirteen different biofilm experiments are tested on the compressors and the resulting metrics are averaged. The images were taken on biofilm grown on PVC coupon on 24-well plate and transferred to fresh media every 24 hours.

Biofilms are a growth that naturally occurs by microorganisms, plants, and algae [10]. Microorganisms are ubiquitous in marine environments, and the formation of biofilms is referred to as microfouling [11]. Naturally forming biofilms develop on submerged surfaces, which create a massive drag penalty on ships, causing lower energy efficiency on crafts [10]. These microorganisms are found within sediment formed on ships over time [7]. The overgrowth of organisms on a ship's hull and bacteria causes increased propulsion fuel use and frequency of refueling, which decreases the ship's range and speed [13]. Current strategies to prevent this drag penalty from occurring include a biocidal coating which raises environmental concerns and fouling release coating which requires a sustained speed of (10-15kn) to be effective. Eco-coating addresses this issue with beneficial biofilms. Eco-coating solutions are being developed utilizing natural marine microbes to form smooth, stable biofilms to reduce drag.

2.2 Lossless Compressors

We evaluate four lossless compressors in this paper: BLOSCLZ [2], Zstandard [4], LZ4 [3], and ZLIB [5]. These lossless compressors are loaded in through LibPressio and implemented in BLOSC. BLOSC [2] is a compressor optimized for binary data. BLOSC is a meta-compressor, so it is able to use different compressors and filters. BLOSC is designed to transmit data to the processor cache at a faster rate compared to standard non-compressed direct memory fetch (memcpy) OS calls. It uses a blocking technique to reduce activity on the memory bus, which is accomplished by dividing the datasets into separate blocks that are small enough to fit in caches on modern processors.

The following lossless compressors are utilized in this study through BLOSC. BLOSC handles different compressors to be able to leverage its blocking technique and supports multithreaded executions.

(1) BLOSCLZ: BLOSCLZ [2] is a compressor heavily based on FastLZ [14]. FastLZ is an implementation of the Lempel-ZiV 77 (LZ77) algorithm of lossless data compression [28]. This algorithm is able to achieve compression by encoding future segments of the data by maximum length copying from a buffer that contains a past output. The code word consists of the buffer address.
(2) Zstandard (ZSTD): ZSTD [4] is a lossless compression algorithm that compresses data made up of frames. ZSTD is a combination of dictionary matching LZ77 [28] with a large search and entropy-coding stage. It uses Huffman coding [16] and finite-state entropy. In this scheme, with the set load of buffer and information contained in the code words, data is reconstructed by decoding starting at the end of the process.
(3) LZ4: LZ4 [3] has two sets of API's LZ4 and LZ4HC, where the HC is the high compression ratio [27]. The lz4 compression algorithm breaks data down into a series of groups. Each of these groups begin with a one byte token that is reduced to two 4-bit fields. The first field is the amount of bytes to be copied to the output. The second field is the number of bytes to copy from the decoded output buffer. The compression is completed in blocks of streams, with high CR values occurring when more time is spent finding the best dictionary matches.
(4) ZLIB: ZLIB [5] compression method uses a variant of LZ77 [28] called deflation. Deflation emits compressed data as a sequence of different blocks. The deflation compressor has three modes: 1) no compression – this is done when another compression has already been performed on the data and the deflation compressor stores the data, 2) Compression with first LZ77 and then with Huffman coding. The trees that are created are defined by the deflation, so extra space allocation is not required, and 3) Compression with LZ77, then Huffman codes with the trees the compressor created and stores along with the data [5].

2.3 Lossy Compressors

We evaluate the two leading lossy compressors:

(1) SZ: SZ [6] is a lossy compressor whose HPC data compression method that is composed of four overall steps. 1) SZ divides the dataset into fixed-sized blocks and then based on the results it selects the most appropriate prediction function

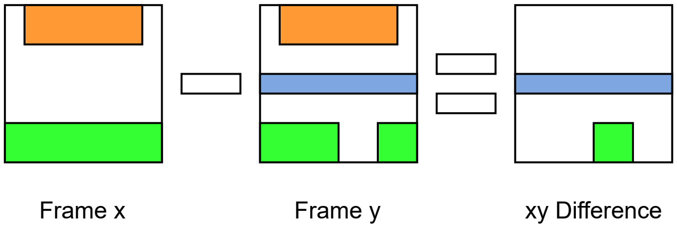


Figure 2: Difference Subtraction for OCT Bscan frames/slices

to predict future values in each block. 2) It performs a linear-scale quantization, with a user-specified error bound SZ, which quantifies the difference between the predicted value and the original data point. This is the quantization index. 3) Encodes the quantization index with a variable–length encoding scheme via Huffman encoding. 4) Lossless compression, improves the compression ratio (CR) by running over the current compressed buffer [6, 17, 22, 24].

(2) ZFP: ZFP [18] is a lossy compressor that uses a block scheme that takes 3D double-precision data and divides the array into small fixed-sized blocks. These blocks have the dimensions of 4 × 4 × 4 and stored using a user-specified amount of bits. Their method compresses these blocks in 5 steps. 1) It aligns the values in the block to a common exponent. 2) Converts the floating-point values to a fixed-point representation. 3) Applies an orthogonal block transform to decorrelate the values. 4) Orders the transform coefficients by their expected magnitude. 5) Encodes the resulting coefficients, one bit plane at a time [18]. ZFP has three modes: fixed rate (set number of bits), fixed accuracy (variable number of bits with fixed number of bit planes), and fixed precision (within set absolute error tolerance) [12].

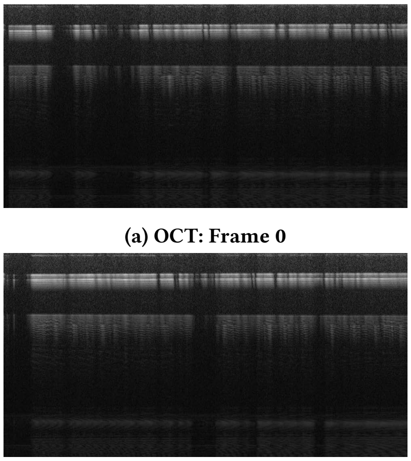
3 ANALYSIS OF BIOFILM COMPRESSION

In this work, the compression and OCT biofilm preprocessing techniques are evaluated on how well the data is reduced and their bandwidth, which is the speed of compressing and decompressing the data. B-scans per volume (slices) are defined as frames for the preprocessing techniques.

For lossy compression, SZ and ZFP both leverage floating point data to improve compression. SZ maps the floating-point prediction error to an integer in quantization, and ZFP puts data in a fixed point representation and utilizes transforms to decorrelate the data. To be able to leverage these features, OCT data is normalized from 0-255 `uint8` to 0-1 `float32` for the compression and back to 0-255 `uint8` after decompression. SZ and ZFP both leverage floating point data to improve compression. The compression ratio (CR) values are evaluated with respect to the original data size before the data factor is increased in the conversion to `float32`.

3.1 Frame 0 Difference

Leveraging the 3D nature of OCT images is tested with multiple pre-processing steps to help transform the data closer to zero to improve the CR. First, the difference is taken between the very first frame of the OCT image and every subsequent image frame. This attempt looks at the initial frame and subsequent frames to find similar data. An example of two frames from the same experiment is found in Figure 3.



(a) OCT: Frame 0

(b) OCT: Frame 49

Figure 3: OCT Bscan Frame Similarity

Figure 2 depicts how the difference is performed. This is done to leverage the similarity in background noise between frames and turn the data to zeros. Both lossy and lossless compression algorithms further compress data that has more repeating zero's stored together. After this is performed, the diff values are passed to the compressors. The formula for this operation is as follows:

$$diff(i,j,k) = (Frame(i,j,k)) - (Frame(0,j,k)) \quad (1)$$

In this operation, every frame following the first frame has its data diffed with the first frame. When decompressing the data this causes a post-processing step where the resulting diffed data needs to be converted back to its original form. This is done by taking the unmodified first frame and adding it to all the subsequent decompressed data. This method is referred as 0diff in graphs/tables.

3.2 Frame Before Difference

With the scanning nature of OCT data, similarities between nearby frames is taken into effect. Looking at the similar background data from each of the frames, this redundancy in the 3D nature is leveraged to improve CR. This attempt looks at frames next to each other to find the most similar data found. An example of two frames from the same experiment is shown in Figure 3. The difference of each frame is taken from the frame before it. This is the same process as Frame 0 diff when compressing, so the compressors are given a 3D OCT image of a rolling diff of each frame. The formula for this is as follows.

$$diff(i,j,k) = (Frame(i,j,k)) - (Frame(i-1,j,k)) \quad (2)$$

Again, for this method, the first frame of the image is not diffed due to being the point of reference in the decompression. Then the decompressed frame is used for the next frame's decompression. This method is referred as Ldiff in graphs/tables.

3.3 Evaluation Metrics

These metrics are used to evaluate compression algorithms. Not all metrics are valid for every compressor, for example, lossless methods have perfect accuracy, so methods that measure accuracy are not needed. There is also a specific scenario where the configuration of lossy methods may preserve all the data, acting similar to a lossless compressor.

To determine which compressor provides the best level of data reduction, compression ratio is used. Compression ratio CR is the efficiency of compression algorithms in the form of comparing original data size to compressed size.

$$CompressionRatio = \frac{UncompressedSize}{CompressedSize} \tag{3}$$

CR shows the efficiency of the relative reduction in size of the data. The higher the CR value, the better the relative reduction of data achieved.

The time required to reduce a dataset is important and very dependent on the configuration setup on the compressors. On average, lossless compressors take more time than lossy compressors and as their level of compression setting increases so does the time it takes to reduce the dataset. That being said, lossy methods on average run much faster than lossless compressors, but that comes at the cost of image quality. Compression bandwidth cBW is the total time required for the data to be fully reduced. This includes pre-processing and compression time. Decompression time is the full time it takes to decompress and post-process the data. Timing does not include loading from the disk due to us looking at real time applications.

$$CompressionBandwidth = \frac{UncompressedSize}{t_{compression}} \tag{4}$$

Decompression bandwidth is the time it takes to decompress the data. This includes time to decompress the compressed data and post-processing steps to bring the data back to its correct form.

$$DecompressionBandwidth = \frac{UncompressedSize}{t_{Decompression}} \tag{5}$$

Lossy compression is capable of generating much higher CR than lossless methods, To be able to achieve much higher CR it comes at the cost of image quality by introducing image distortion. The effectiveness of lossy compression is evaluated by its accuracy. Error-bound methods like ZFP and SZ provide a precise control on the error bounding value to examine the error. In this paper, we use SSIM to evaluate the performance of lossy methods. SSIM is a metric for lossy compression that evaluates the structural degradation of image quality during compression [26].

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{6}$$

μ_x = pixel sample mean of x, μ_y = pixel sample mean of y, σ_x^2 = variance of x, σ_y^2 = variance of y, σ_{xy} = covariance of x and y, $c_1 = (k_1L)^2, c_2 = (k_2L)^2$ = variables to stabilize the division, L is the dynamic range of pixel values, $k_1 = 0.01$ and $k_2 = 0.03$ are default settings.

SSIM is used over other accuracy metrics such as peak signal-to-noise ratio (PSNR) because the OCT images we are looking at have substantial background noise, which would skew the PSNR value. Important data in the image includes biofilm structure that is analyzed by OCT. Since the structure of the image is important to keep intact, SSIM is implemented.

4 RESULTS

4.1 Testing Environment

Tests are performed on Clemson University's Palmetto Cluster. The node requested for the experiment contains 2× 20-core Intel(R) Xeon(R) Gold 6258R CPUs with a clock frequency of 2.70GHz and 384 GB of RAM. Software for the compressors and environment is defined in the following Table.

Software	Version
GCC	12.1.0
SZ	2.1.12
ZFP	1.0.0
ZSTD	1.5.5
LZ4	1.9.4
ZLIB	1.2.13
BLOSC	1.21.2
LibPressio	1.21.2

The dataset is stored on the Palmetto Cluster's scratch directory, which is an Indigo file system with SSD disk type and an Infiniband (Mellanox Technologies MT28908 Family) and Ethernet network connection. Tests are run on 13 different biofilm experiments and results are averaged. Thus, in total, each data point is the average of these data files. Compression algorithms are evaluated with LibPressio [25], which is a compression library that provides a common interface to various lossless and lossy compressors. Each experiment is stored as a series of tiff files, which are the frames of the OCT image. These files are loaded into memory and combined into a single OCT image when the calculations are performed.

4.2 Compressor Configuration

Configurations for lossless compressors are handled by setting the compression level's (1–9) and testing each mode over the data. SZ and ZFP both allow the user to bound the level of distortion in the data in an error bounding mode. When reducing, the error bounding value precisely controls the distortion level of the data. Configurations for lossy compressors are tested over a series of error bounds (1E-7 – 1E-1). For SZ and ZFP, as the error bound configuration increases the CR, bandwidth, and SSIM are reduced.

4.3 Lossless Compression

To determine which lossless method gives the best reduction level, each of the compressors are run over the entire dataset and the average compression ratios are presented. From lossless compressors, ZSTD gives the best CR ∼ 4.5× as seen in Figure 4

Blosclz has the highest average compression bandwidth, $\sim 78MB/s$ which is closely followed by LZ4, as shown in Figure 5a. As the compression level for the lossless compressors increases, the compression bandwidth decreases. For decompression, LZ4/LZ4hc performs slightly better comparatively with decompression bandwidth $\sim 27MB/s$ and is shown in Figure 5. The decompression bandwidth

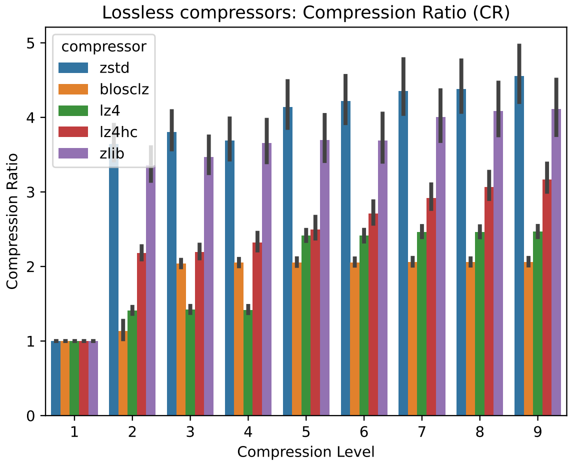


Figure 4: Lossless Compression Level on Compression Ratio

is still affected by the increasing compression level, but much less when compared to the compression bandwidth.

The lossless compressors did not benefit from pre-processing, the unaltered base performance obtains the highest CR as seen in Figure 6. The base implementation provides the highest CR values on each of the error bounds.

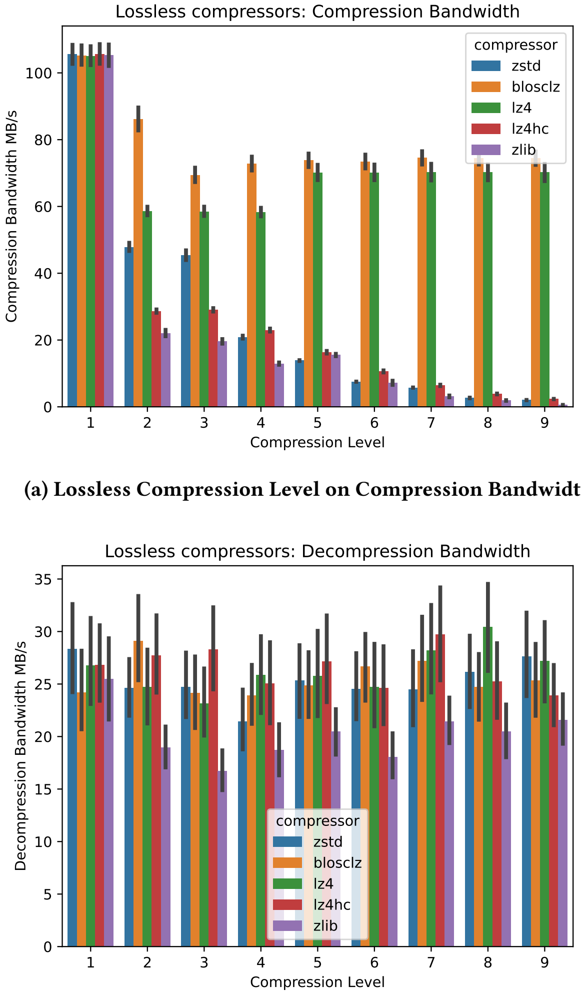
4.4 Lossy Compression

The lossy compressors are evaluated over a set error bound. The error bound is used to set the accuracy of the data to be able to analyze the effect it has on compression Ratio, Compression Bandwidth, and Decompression Bandwidth. The absolute error bound is varied between 1E-7 to 1E-1. As the error bound for the compressors increases, so does the compression ratio. As shown in Figure 7, average compression ratio achieved on the highest error bound by SZ is 204.6× on 1E-1 and by ZFP is 4.1×. When comparing the two lossy compressors there are different trade-offs between them. For error bounds less than 1E-4 the blosc lossless compressors are able to outperform the lossy compressors CR.

The frame before pre-processing is able to achieve up to an 33.98% CR increase on lossy SZ at error bound 1E-1. On error bounds lower than 1E-2 the base method is able to outperform or stay consistent with the pre-processing methods. As the error bound increase, the frame before pre-processing is able to outperform the base methods. For lossless methods, pre-processing caused on average a 12.82% decrease in CR, as seen in Figure 8.

Figure 9 shows the reduction bandwidth for lossy methods. For each of the time steps, bandwidth stays mostly consistent. Bandwidth does slightly increase as error bounds increase. When comparing the two lossy compressors, SZ is behind ZFP for compression and decompression bandwidths. ZFP on average obtains a higher compression bandwidth averaging $\sim 40MB/s$ compared to SZ averaging to $\sim 32MB/s$ on the highest error bound 1E-1.

As the error bound for SZ and ZFP increase, compression and decompression bandwidths also increase. For error bounds 1E-2 and lower, ZFP again outperforms SZ in compression. On that error bound and higher SZ is able to achieve a higher decompression bandwidth as seen in Figure 9b.



(a) Lossless Compression Level on Compression Bandwidth

(b) Lossless Compression Level on Decompression Bandwidth

Figure 5: Lossless Bandwidth

4.5 Image Distortion

Lossless compressors perfectly preserve the data, while lossy compressors loses information to improve CR. Lossy compression achieves a higher CR at the cost of image distortion. Figure 10 shows the SSIM over various error bounds. SZ begins degrading at 1E-2 while ZFP remains consistent. SZ achieves a higher CR when compared to ZFP, but its SSIM degrades more than ZFP when error bounds increase. This trade off between CR, bandwidth against accuracy is the limiting factor for lossy compressors. Data pre-processing also has an effect on accuracy of the data when it is compressed with a lossy compressor. On the lower error bounds, SSIM stays relatively consistent, but as the error increases, the minimized data begins to drop off in accuracy when compared to base unaltered data. The

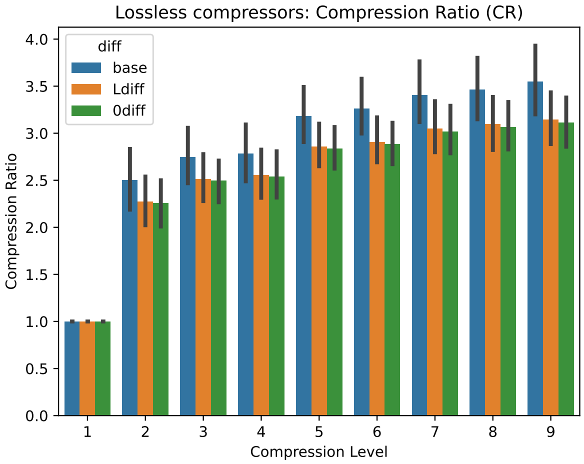


Figure 6: Lossless Pre-processing Compression Level on Compression Ratio (Note: Ldiff refers to the frame before difference, 0diff refers to the frame 0 difference and base is unaltered data)

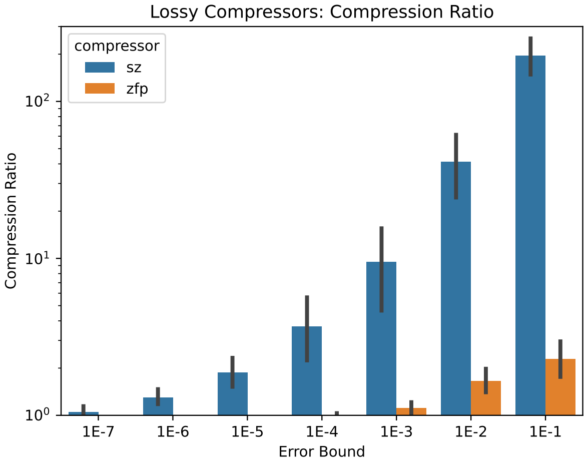


Figure 7: Lossy absolute error bound for different compressors

frame before diff on the highest error bound has ~ 0.2 difference compared to the 0 diff technique, which is slightly lower than the base data.

5 RELATED WORK

High-Resolution Wavelet-Fractal Compressed Optical Coherence tomography images [8]

3D OCT images are compressed by proposing a 3D extension of the wavelet-fractal coding algorithm. The authors use 3D fractal approximation to encode 3D wavelet coefficients to exploit the inter and intra redundancy. Their encoding method includes 1) calculating the N-level Haar wavelet transform 2) partitioning the components into various block domains 3) finding the best matching domain block 4) saving the map. When decoding, the following procedures are implemented: 1) calculate the N-level Haar wavelet transform 2) for each N-level, calculate the components for each of the coded information 3) calculate the N-level inverse transform to reconstruct at that level 4) repeat until the entire image is reconstructed. In their wavelet-fractal method, the authors are able to achieve an average CR of 21.49 with a PSNR ranging between 25 and 27. They use a different method for evaluating the OCT data and use PSRN instead of SSIM to evaluate the lossy compression.

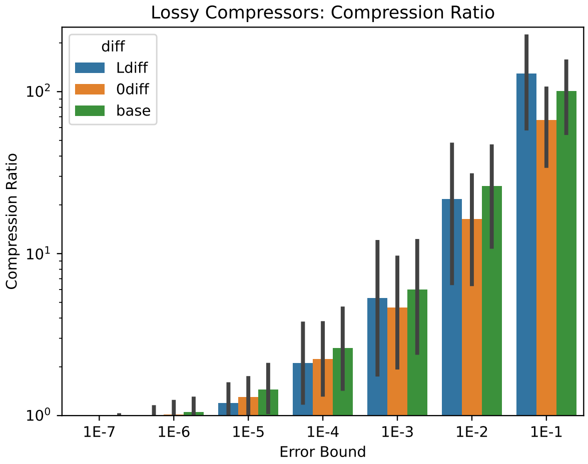


Figure 8: Lossy absolute error bound for different preprocessing. (Note: Ldiff refers to the frame before difference, 0diff refers to the frame 0 difference and base is unaltered data)

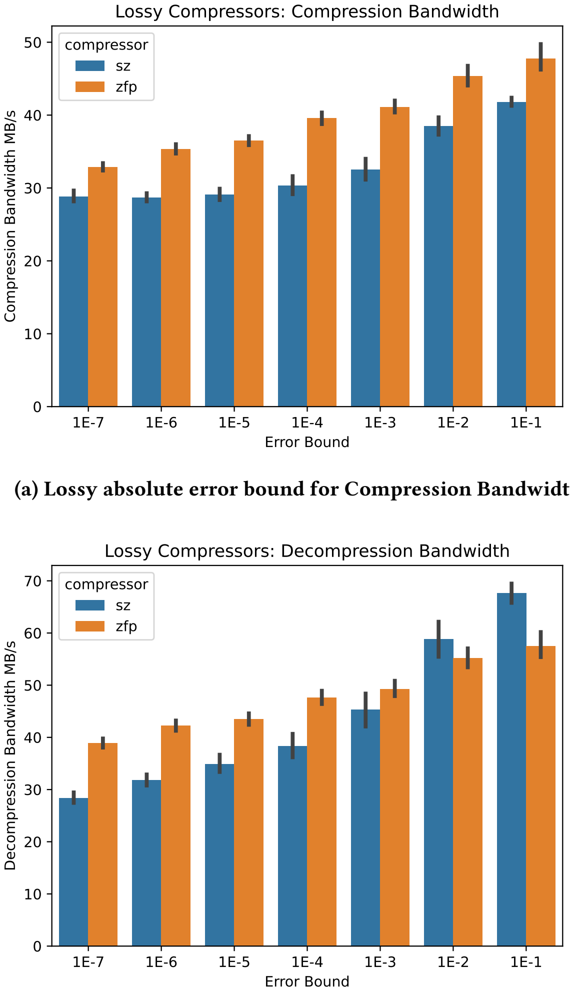
A Digital Method for Lossless and Lossy Compression of High Definition Optical Coherence Tomography Data [20]

This work studies OCT data and its extensive storage cost and need for compression tools. Examining both lossy and lossless compressors, the tools analyzing are zip, gzip, bzip2, lzma, 7-zip. The best resulting compression ratio achieved is 2.78 CR on Bzip2. Utilizing background removing, re-sampling, and BZip2 the authors are able to achieve a CR value of 5.47.

This work is similar to ours, but we implement more state-of-the-art compressors for the lossless and lossy compressors. They also implement a different method for preprocessing by performing a background subtraction algorithm instead of slice/frame based.

3-D Adaptive Sparsity Based Image Compression with Applications to Optical Coherence Tomography [9]

Improving the performance of sparse representation for compressing of OCT images is examined. The authors identify that nearby slices of the OCT image are similar and apply slice-based compression methods. Their 3D framework for OCT compression utilizes high correlation between nearby slices. The method is broken down into three separate parts, 1) 3D adaptive sparse representation 2) 3D adaptive encoding 3) decoding and reconstruction.



(a) Lossy absolute error bound for Compression Bandwidth

(b) Lossy absolute error bound for Decompression Bandwidth

Figure 9: Lossy Bandwidth

Metrics used for evaluation on the performance of the compression are PSNR and FSIM. Experiments ran on 7 different set compression ratios ranging from 10 up to 40. A 2D and 3D version of the algorithm is implemented and show PSNR/FSIM improvements between their two versions and other standard compressors. The authors conclude the 3D methods which leverage slices are improvements over the 2D version. This is similar to our study in levering the data structure but uses a different metrics in accuracy evaluation and have set CR values.

6 CONCLUSION

To effectively analyze biofilm OCT images, data reduction is essential. These results show that although lossless methods are perfectly preserved and reduce the data, they have much less achievable CR compared to lossy methods. When looking at lossy methods at error bounds above 1E-4, it is able to outperform lossless methods.

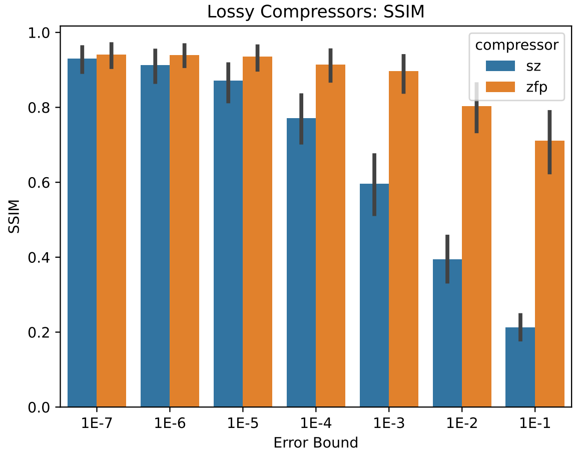


Figure 10: Lossy Absolute Error Bound on SSIM

We also explored leveraging the 3D nature of the OCT by taking the difference between frames. Improvements in higher error bounds show how volume data can be leveraged. Frame before difference was able to achieve the highest CR on higher error bounds with SZ. ZSTD yields the best CR for lossless compressors, while SZ is able to provide the best compression for lossy compressors. ZFP is seen to have higher SSIM, compression bandwidth, and decompression bandwidth when compared to SZ. Lossless compressors trade compression time for smaller file sizes, and lossy compressors trade data distortion for smaller file sizes. In conclusion, this study shows that SZ with frame before difference pre-processing is the best compressor for the biofilm OCT dataset with higher CR's and bandwidths, and it also outperforms lossless methods.

SSIM is an important component for OCT evaluation to determine usability of lossy data in biofilm analysis. The structural integrity of SSIM provides insight into finding the biofilms ability to reduce ship drag penalty. Future studies should address SSIM values that are needed for an accurate analysis of biofilm OCT data.

ACKNOWLEDGMENTS

Clemson University is acknowledged for their generous allotment of compute time on the Palmetto Cluster. This work is sponsored by the United States Defense Advanced Research Projects Agency under agreement HR00112320008. The content of the information does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred. This material is based upon work supported by the National Science Foundation under Grant No. SHF-1943114.

REFERENCES

[1] [n. d.]. Thorlabs Inc. - Your Source for Fiber Optics, Laser Diodes, Optical Instrumentation and Polarization Measurement &amp; Control. https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=8214
[2] Francesc Alted. 2017. Blosc, an extremely fast, multi-threaded, meta-compressor library.
[3] Yann Collet. [n. d.]. LZ4 lossless compression algorithm. https://lz4.org/
[4] Yann Collet and Murray Kucherawy. 2018. *Zstandard Compression and the application/zstd Media Type.* Technical Report.

[5] Peter Deutsch and Jean-Loup Gailly. 1996. *Zlib compressed data format specification version 3.3.* Technical Report.

[6] Sheng Di and Franck Cappello. 2016. Fast error-bounded lossy HPC data compression with SZ. In *2016 ieee international parallel and distributed processing symposium (ipdps)*. IEEE, 730–739.

[7] Lisa A Drake, Martina A Doblin, and Fred C Dobbs. 2007. Potential microbial bioinvasions via ships' ballast water, sediment, and biofilm. *Marine pollution bulletin* 55, 7-9 (2007), 333–341.

[8] Mekhalfa Faiza, Saba Adabi, Berkani Daoud, and Mohammad RN Avanaki. 2017. High-resolution wavelet-fractal compressed optical coherence tomography images. *Applied Optics* 56, 4 (2017), 1119–1123.

[9] Leyuan Fang, Shutao Li, Xudong Kang, Joseph A Izatt, and Sina Farsiu. 2015. 3-D adaptive sparsity based image compression with applications to optical coherence tomography. *IEEE transactions on medical imaging* 34, 6 (2015), 1306–1320.

[10] Andrea Farkas, Soonseok Song, Nastia Degiuli, Ivana Martić, and Yigit Kemal Demirel. 2020. Impact of biofilm on the ship propulsion characteristics and the speed reduction. *Ocean Engineering* 199 (2020), 107033.

[11] Eugene Georgiades, Chris Scianni, and Mario N Tamburri. 2023. Biofilms associated with ship submerged surfaces: implications for ship biofouling management and the environment. *Frontiers in Marine Science* 10 (2023), 1197366.

[12] Pascal Grosset, Christopher M. Biwer, Jesus Pulido, Arvind T. Mohan, Ayan Biswas, John Patchett, Terece L. Turton, David H. Rogers, Daniel Livescu, and James Ahrens. 2020. Foresight: Analysis That Matters for Data Reduction. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. 1–15. https://doi.org/10.1109/SC41405.2020.00087

[13] Elizabeth G Haslbeck and Gerard S Bohlander. 1992. Microbial Biofilm Effects on Drag No. 3A–Lab and Field. In *1992 Ship Production Symposium*.

[14] A Hidayat. 2019. Fastlz, free, open-source, portable real-time compression library. *URL http://www. fastlz. org* (2019).

[15] David Huang, Eric A Swanson, Charles P Lin, Joel S Schuman, William G Stinson, Warren Chang, Michael R Hee, Thomas Flotte, Kenton Gregory, Carmen A Puliafito, et al. 1991. Optical coherence tomography. *science* 254, 5035 (1991), 1178–1181.

[16] David A Huffman. 1952. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE* 40, 9 (1952), 1098–1101.

[17] Xin Liang, Sheng Di, Dingwen Tao, Sihuan Li, Shaomeng Li, Hanqi Guo, Zizhong Chen, and Franck Cappello. 2018. Error-controlled lossy compression optimized for high compression ratios of scientific datasets. In *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 438–447.

[18] Peter Lindstrom. 2014. Fixed-rate compressed floating-point arrays. *IEEE transactions on visualization and computer graphics* 20, 12 (2014), 2674–2683.

[19] Chaitanya K. Mididoddi, Fangliang Bai, Guoqing Wang, Jinchao Liu, Stuart Gibson, and Chao Wang. 2017. High-Throughput Photonic Time-Stretch Optical Coherence Tomography with Data Compression. *IEEE Photonics Journal* 9, 4 (2017), 1–15. https://doi.org/10.1109/JPHOT.2017.2716179

[20] OO Myakinin, VP Zakharov, IA Bratchenko, DV Kornilin, and AG Khramov. 2014. A digital method for lossless and lossy compression of high definition optical coherence tomography data. In *Biomedical Optics*. Optica Publishing Group, BT3A–67.

[21] Coleman Nichols, Megan Hickman Fulp, Nathan DeBardeleben, and Jon C. Calhoun. 2022. Exploring Data Reduction Techniques for Additive Manufacturing Analysis. In *2022 IEEE/ACM 8th International Workshop on Data Analysis and Reduction for Big Scientific Data (DRBSD)*. 21–28. https://doi.org/10.1109/DRBSD56682.2022.00008

[22] Ruiwen Shan and Jon C. Calhoun. 2022. Exploring Data Corruption Inside SZ. In *2022 IEEE International Conference on Big Data (Big Data)*. 3172–3178. https://doi.org/10.1109/BigData55660.2022.10020891

[23] David Stifter. 2007. Beyond biomedicine: a review of alternative applications and developments for optical coherence tomography. *Applied Physics B* 88 (2007), 337–357.

[24] Dingwen Tao, Sheng Di, Zizhong Chen, and Franck Cappello. 2017. 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, 1129–1139.

[25] Robert Underwood, Victoriana Malvoso, Jon C Calhoun, Sheng Di, and Franck Cappello. 2021. 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, 1–10.

[26] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.

[27] Zhe Zhang and Brian Bockelman. 2017. Exploring compression techniques for ROOT IO. In *Journal of Physics: Conference Series*, Vol. 898. IOP Publishing, 072043.

[28] Jacob Ziv and Abraham Lempel. 1977. A universal algorithm for sequential data compression. *IEEE Transactions on information theory* 23, 3 (1977), 337–343.