Report of SZ Lossy Compression on Sample EXAFEL Datasets

SZ compressor development team

Feb. 2017

Introduction

In this report, we evaluate the lossy compressor SZ [1] and compare it with state-of-the-art lossy and lossless compressor on the sample EXAFEL datasets.

Before presenting the evaluation methodology and results, we introduce several metrics that will be used for measuring the compression quality in this report.

- *Compression ratio* is used to evaluate the size reduction as a result of the compression, which is calculated by the original data size divided by the compressed data size.
- *Bit-rate* (bits/value) represents the amortized storage cost of each value after compression.
- *Distortion* evaluates the average error in the compression. The commonly used distortion metrics include root mean squared root (RMSE), normalized RMSE (NRMSE), and peak signal-to-noise ratio (PSNR). In this report, we use PSNR to measure the distortion quality. The PNSR is calculated as $20\log_{10}(R_X/\text{RMSE})$ in decibel, where R_X is the value range of the dataset.
- *Rate-distortion* represents the distortion quality per bit of compressed storage. Rate means bit-rate in bits/value. For a fair comparison, we plot the rate-distortion curves for all the lossy compressors and compare the distortion with the same bit-rate.

- *Compression/decompression Rate* is to evaluate the processing speed. Compression rate, for example, refers to the amount of data (MB) to be compressed per second.
- *Auto-correlation of errors* is to assess the auto-correlation of the compression errors across the data points. In general, the users expect to see close-to-zero auto-correlation of compression errors.
- *Distribution of errors* refers to the probability distribution density of the compression errors in our evaluation.
- *Spectrum analysis* refers to the comparison of the spectrum value (generated by DFT function) between the initial data set and the decompressed data set. Users expect to see little distortion of the spectrum results in-between.

We used three types of error bounds: *absolute error bound*, *value-range based relative error bound* and *point-wise relative error bound*.

- *Absolute error bound* refers to a specific constant given by users to control the maximum compression errors for all data points in the data set. For example, if the absolute error bound is set to 5, then the compression error of each reconstructed data point will be in [-5,5] compared with its initial value.
- *Relative error bound* is used to control the errors compared with the global data value range. Specifically, the compressor will compress the data such that the compression errors for every data point are always bounded within a percentage of the global value range. For example, suppose the value range is 10000 and the relative bound ratio set by users is 0.01, then the actual absolute error bound will be 10000*0.01=100 for each data point. Actually, if users set the value-range-based error bound, SZ compressor will first compute the absolute error bound and then compress the data according to the absolute error bound. Compared with absolute error bound setting, the relative error bound setting allows users to control the errors in terms of the value range, which is particularly helpful when the value ranges change largely over the snapshots/time steps during the simulation.
- SZ compressor also provides another error bound mode, i.e., *point-wise relative error bound*. Unlike the value-range-based relative error bound that is compared with value range, the point-wise relative error bound is compared with each data value.

Visualization of EXAFEL datasets

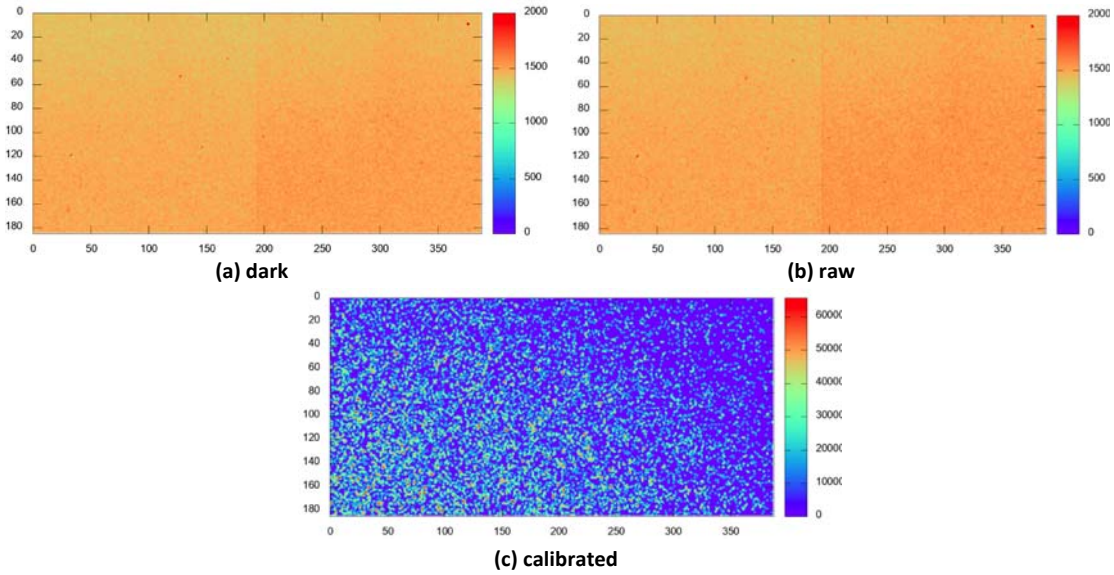

Figure 1. Partial visualization of EXAFEL sample datasets

EXAFEL sample datasets are composed of three subsets, which are dark dataset, raw dataset, and calibrated dataset. The size of each dataset is 10*32*185*388. As described by LCLS-II researcher, the '10' is for the number of images, each with shape of 32*185*388. The '32' is for the separate panels present in one detector. We visualize one panel with the size 185*388 for each dataset. Figure 1 shows the visualization.

Methodology of evaluating SZ on EXAFEL datasets

The EXAFEL datasets are composed of uint_16 or int_16 integer data. While the SZ compressor is designed for floating-point data (single or double), we first convert the uint_16 or int_16 data to the single floating-point data. We then perform the SZ compression on the converted single floating-point data sets. Specifically, we perform 320 (10*32) 2D compression, with each for one 185*388 2D panel. For the decompression, we first perform the SZ decompression on the compressed file and get the single floating-point data. After that, in order to get the same bit-rate as the original uint_16 or int_16 data, we convert the single floating-point data to half-precision floating-point data (16 bit per value) [2] and store them to a decompressed file, which has the same storage size as the original integer file.

Comparison of SZ and Gzip on EXAFEL

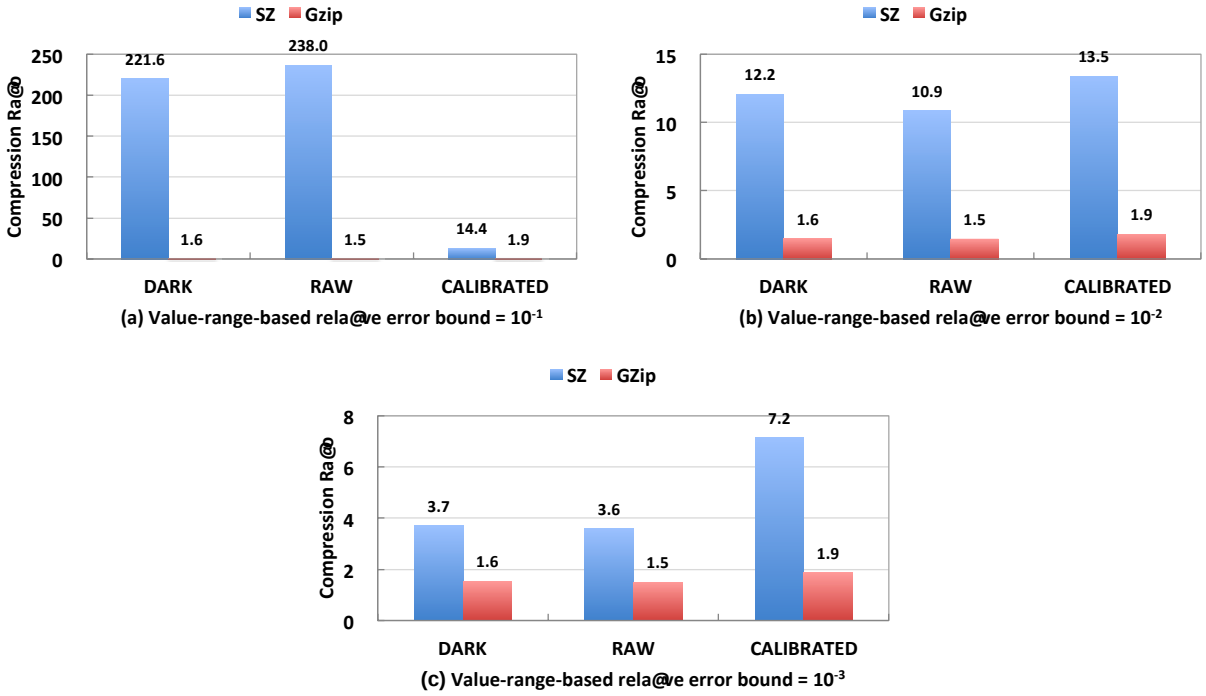

Figure 2. Comparison of compression ratio between SZ with different value-range-based relative error bounds and Gzip on EXAFEL sample datasets

In this section, we compare SZ with the lossless compressor Gzip [3]. We compare the compression ratio of SZ with that of Gzip under the *value-range-based relative error bound*, as shown in Figure 2. As mentioned previously, the value-range-based relative error bound is defined as the absolute error bound divided by the value range of the dataset. Figure 2 shows that SZ's compression ratios are much higher than Gzip's on the whole sample datasets with the value-range-based error bound 10^{-1},10^{-2}, and 10^{-3}.

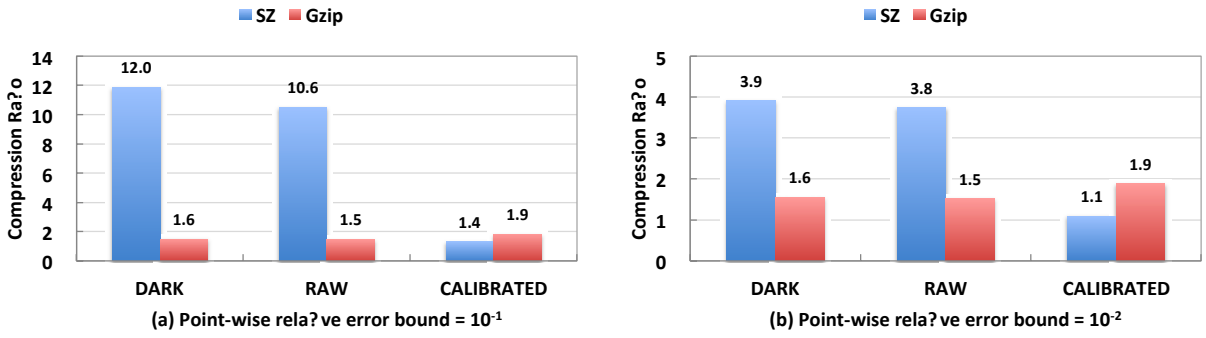

Figure 3. Comparison of compression ratio between SZ with different point-wise error bounds and Gzip on EXAFEL sample datasets

We also evaluate the compression ratio of SZ on the EXAFEL sample data sets under the point-wise relative error bound, as shown in Figure 3. Figure 3 illustrates that using

point-wise error bound is worse than using value-range-based error bound, especially on the calibrated dataset. After looking into the calibrated dataset, the reason is likely that the data values change largely in a small block (6x6) of one panel while SZ needs to compute a uniform compression error bound for all data points in each block based on the smallest value in the block. That is, the compression errors with the point-wise error mode will be over-conservative with regard to the accuracy requirement. In this sense, we think the value-range-based relative error bound makes more sense, so we adopt it in the remaining evaluations.

Comparison of SZ and JPEG2000 on EXAFEL

In this section, we compare SZ with the popular image compressor JPEG2000 [4]. JPEG2000 is designed for a fixed bit-rate, whereas SZ is designed for a fixed maximum compression error. Thus, for a fair comparison, we plot the rate-distortion curves for both SZ and JPEG2000, as shown in Figure 4. We evaluate JPEG2000 on the sample EXAFEL datasets in the following five steps:

1. Partition the initial 10*320*185*388 dataset into 320 two dimensional datasets (i.e., 320 panels)
2. Write each 2D panel to a lossless graphics file in PNG format
3. Use OpenJPEG library [5] to compress these PNG files to the JPEG2000 files
4. Use OpenJPEG library to decompress these compressed JPEG2000 files to PNG files
5. Load the decompressed PNG files and the initial PNG files to calculate the PSNR.

We have verified that the conversion between 2D panel dataset and PNG graphics file is lossless. Figure 4 shows that SZ compressor's rate-distortion curve is a bit worse than JPEG2000's rate-distortion curve on the dark and raw dataset but is much better than JPEG2000's rate-distortion curve on the calibrated dataset. It illustrates that SZ compressor uses bit in the compressed storage more efficiently than JPEG2000 on the calibrated dataset. Note that we only plot the rate-distortion curves of JPEG2000 within 8 bits/value since the lowest compression ratio JPEG2000 allows to set is 2.

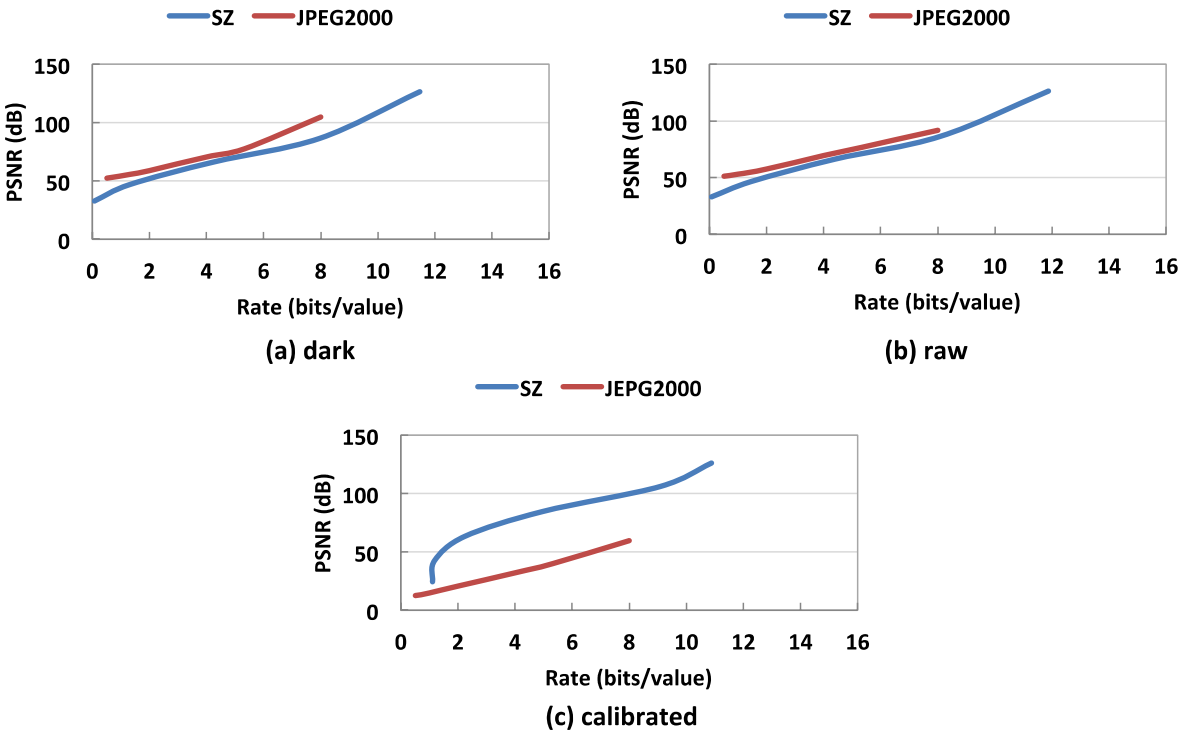

Figure 4 Comparison of rate-distortion between SZ and JPEG2000 on EXAFEL datasets

Autocorrelation analysis of compression errors

In this section, we analyze the autocorrelation of the compression errors since some applications require the compression errors to be uncorrelated or white noise. When computing the autocorrelation of errors, we treat the dataset as 1D data set, which can be considered as 1D signal. The autocorrelation is defined with respect to delays and the delay is the index of the 1D dataset. Figure 5 shows the first 100 autocorrelation coefficients of the compression errors with increasing lags using SZ compressor (under the value-range-based relative error bound 10^{-2}) on the sample EXAFEL datasets. Theoretically, smaller autocorrelation coefficient value indicates less correlation of compression errors across data points in the data set. The dark dataset is a noise data and Figure 5 (a) shows that the autocorrelation coefficients of compression errors are smaller than 0.01, which means SZ compression introduces little autocorrelation to the original noise data.

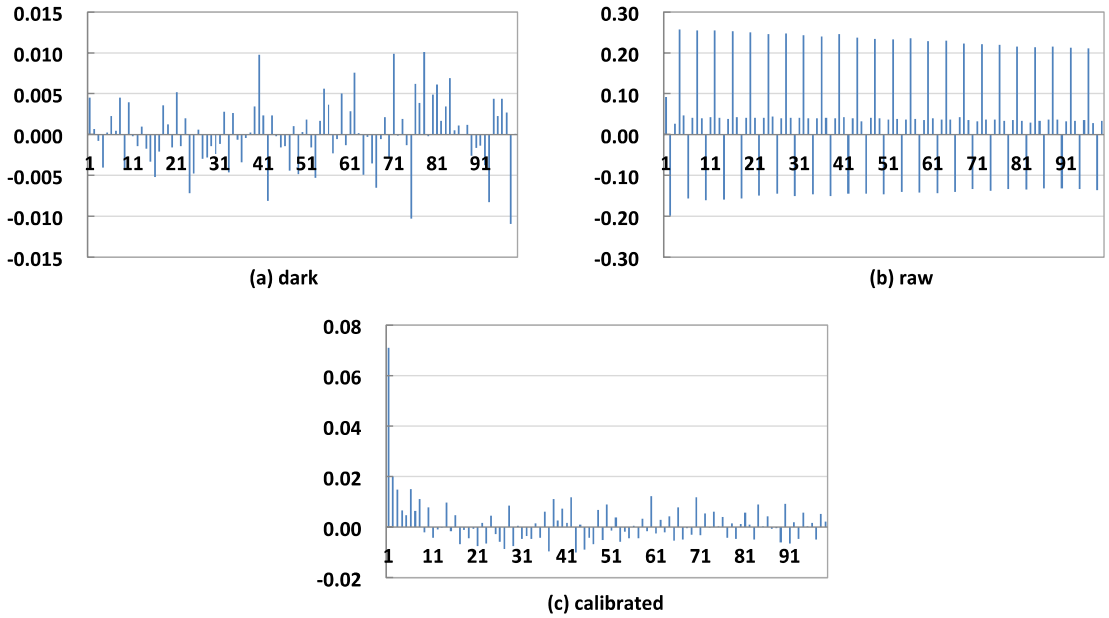

Figure 5 Autocorrelation analysis (first 100 coefficients) of compression errors with increasing delays using SZ on EXAFEL datasets

Distribution of compression errors

In this section, we evaluate the distribution of compression errors for SZ and JPEG2000 on the EXAFEL datasets. Figure 6 -11 present the distribution of compression errors of SZ and JPEG2000 on one panel of the dark, raw, and calibrated data set separately. Figure 6 - 8 shows the error distribution of SZ using the value-range-based relative error bound 10^{-2} and JPEG2000 using the same compression ratios of SZ since JPEG2000 does not have error bound mode. Similarly, Figure 9 - 11 present the error distribution of SZ and JPEG2000, but the error bound of SZ is set to the value-range-based relative error bound 10^{-3} and the compression ratios of JPEG2000 are set to the same ones of SZ under such error bound.

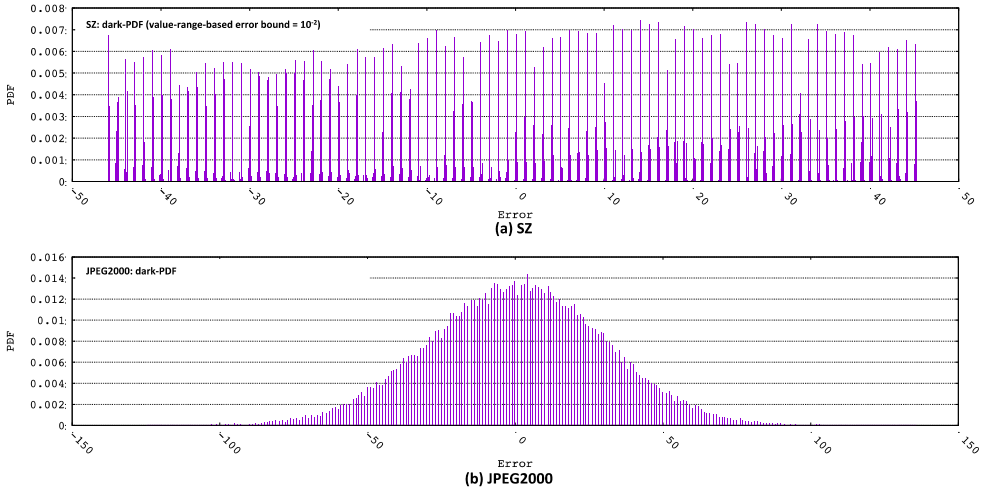

Figure 6 Distribution of compression errors of (a) SZ with value-range-based relative error bound 10^{-2} and

(b) JPEG2000 on the dark dataset

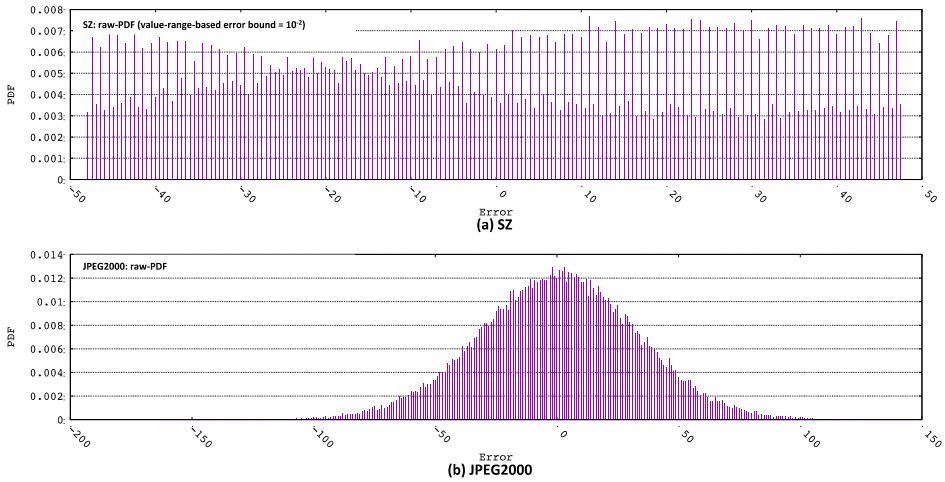

Figure 7 Distribution of compression errors of (a) SZ with value-range-based relative error bound 10^{-2} and (b) JPEG2000 on the raw dataset

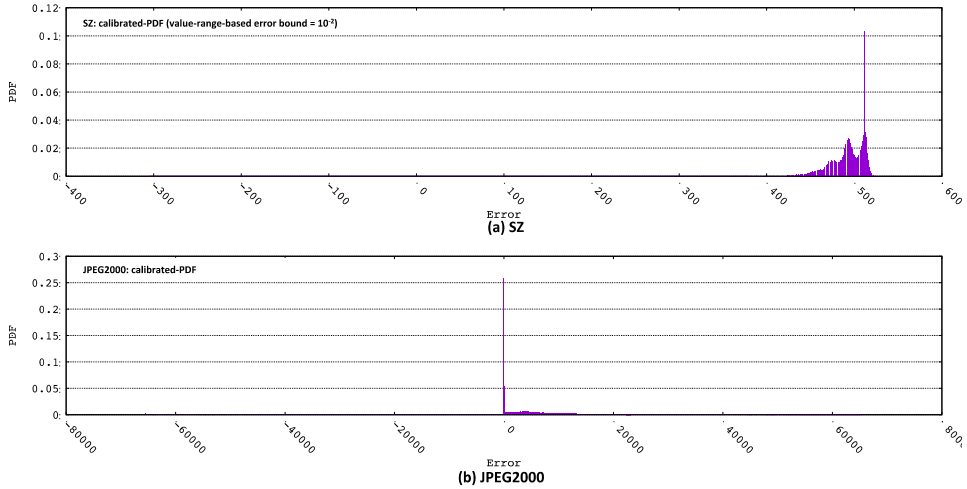

Figure 8 Distribution of compression errors of (a) SZ with value-range-based relative error bound 10^{-2} and (b) JPEG2000 on the calibrated dataset

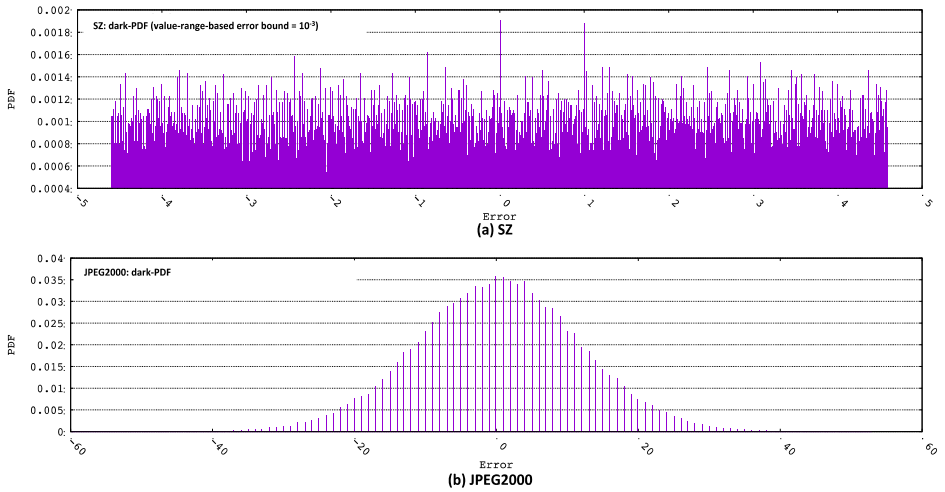

Figure 9 Distribution of compression errors of (a) SZ with value-range-based relative error bound 10^{-3} and (b) JPEG2000 on the dark dataset

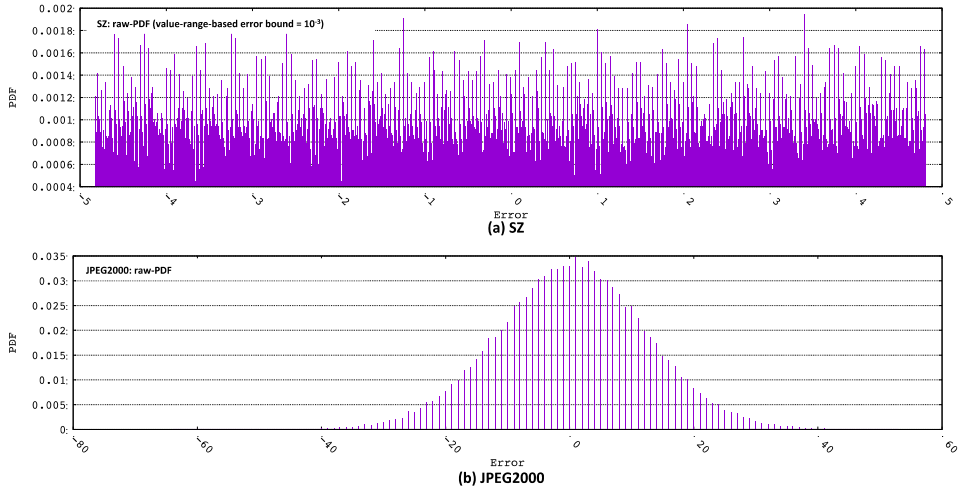

Figure 10 Distribution of compression errors of (a) SZ with value-range-based relative error bound 10^{-3} and (b) JPEG2000 on the raw dataset

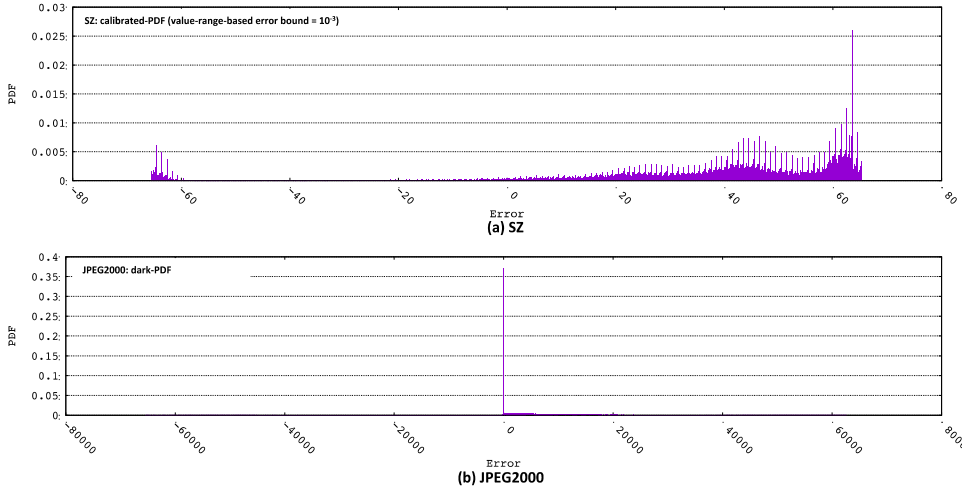

Figure 11 Distribution of compression errors of (a) SZ with value-range-based relative error bound 10^{-3} and (b) JPEG2000 on the calibrated dataset

Figure 6, 7, 9, and 10 illustrate that the compression errors of SZ are basically uniform distribution on the dark and raw dataset, while the compression errors of JPEG2000 are normal distributions on the dark and raw dataset. However, the compression errors of SZ and JPEG2000 on the calibrated dataset are totally different from the other two datasets. The reason why the compression errors follow such a distribution on the calibrated dataset still needs to be studied in the future. We also note the maximum absolute compression errors of JPEG are much higher than those of SZ. For example, in Figure 11, the maximum absolute compression error of JPEG2000 is about 80,000, whereas the maximum absolute compression error of SZ is only about 80. Note that the distribution of compression errors in Figure 11 is not uniform and skewed to one side compared with 0 is due to the fact that the prediction method in the current version of SZ seems not very effective for the calibrated data set. In order to get the compression errors located around 0, we can either reduce the compression error bound or improve the prediction method (to be studied in the future).

Spectrum analysis: Discrete Fourier transform on original and decompressed datasets

Due to information loss, lossy compression affects the spectral properties of the dataset. In this section, we compare the results of DFT on original and reconstructed EXAFEL datasets. In other words, we evaluate the effects that lossy compressors bring to DFT on the EXAFEL datasets. We generate two sets of DFT results on the original and reconstructed EXAFEL data sets at the same time and compare the amplitudes of these two DFT results from low frequency to high frequency. Figure 12 - 14 shows the difference of these two DFT's amplitudes using SZ and JPEG2000 on one panel in the dark, raw, and calibrated dataset. For SZ, we test two error bounds, which are the value-range-based relative error bound 10^{-2} and 10^{-3}. For JPEG2000, similar to evaluations of error distribution, we use the compression ratios as same as SZ's. Note that the difference is normalized to each amplitude of the DFT performed on the original datasets, hence, this difference can be considered as point-wise relative error of DFT's amplitudes. Figure 12 - 14 illustrate that the amplitude differences of SZ with the value-range-based relative error bound 10^{-3} are smaller than the amplitude differences of JPEG2000 with the corresponding compression ratios. Specifically, the normalized difference of amplitudes using SZ can be within 1% on the calibrated dataset, whereas, the normalized difference of amplitudes using JPEG2000 can only be within 500% with most of normalized differences are within 100%).

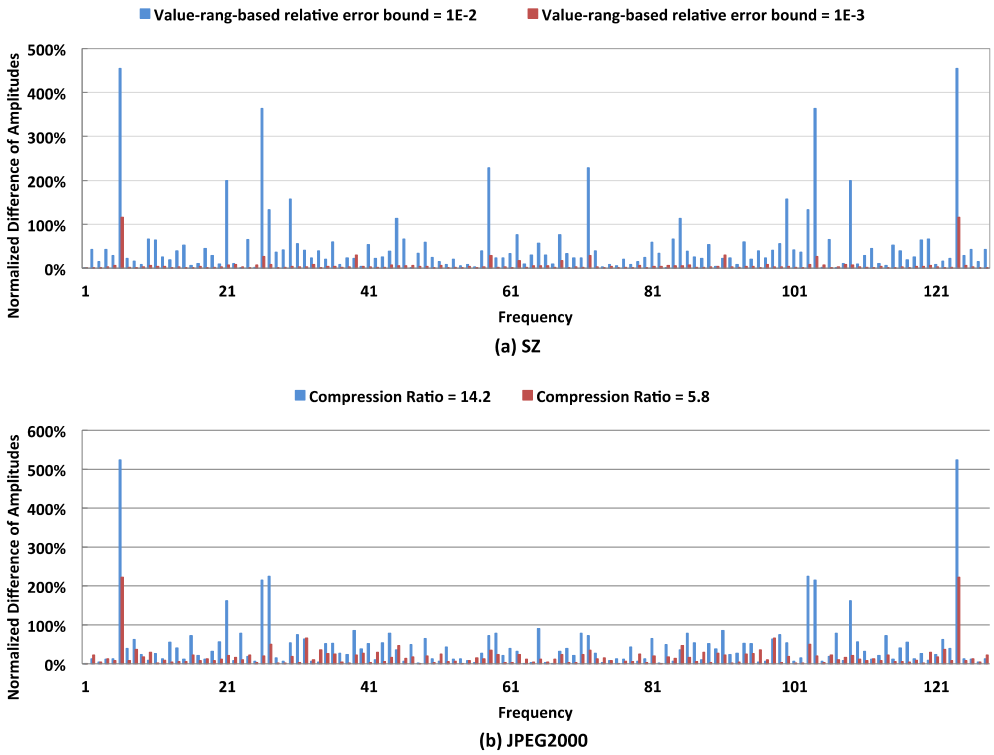

Figure 12 Normalized difference of DFT's amplitudes on the original and reconstructed dark dataset

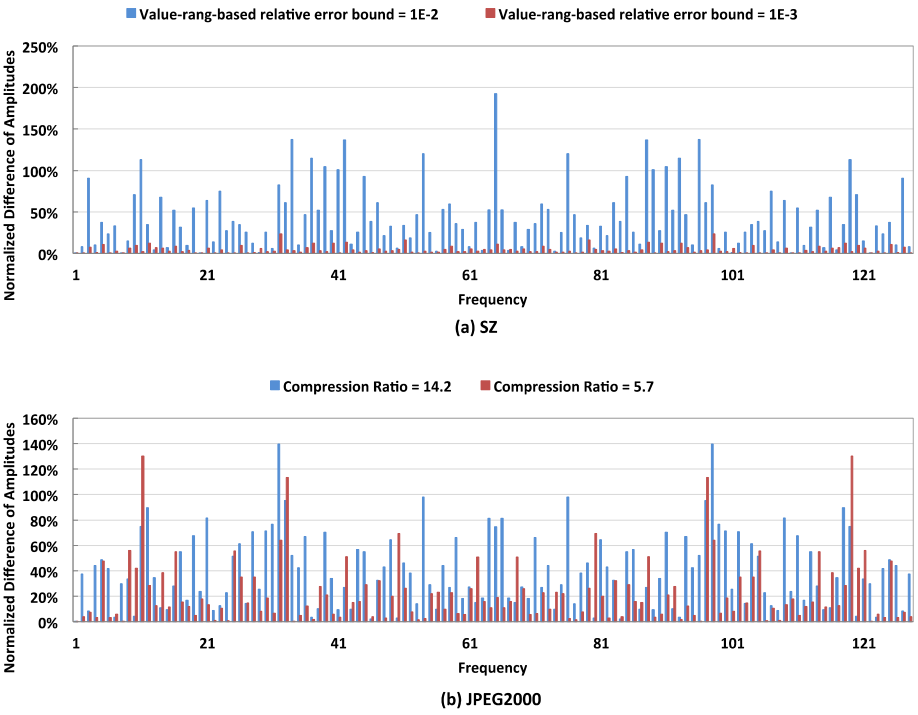

Figure 13 Normalized difference of DFT's amplitudes on the original and reconstructed raw dataset

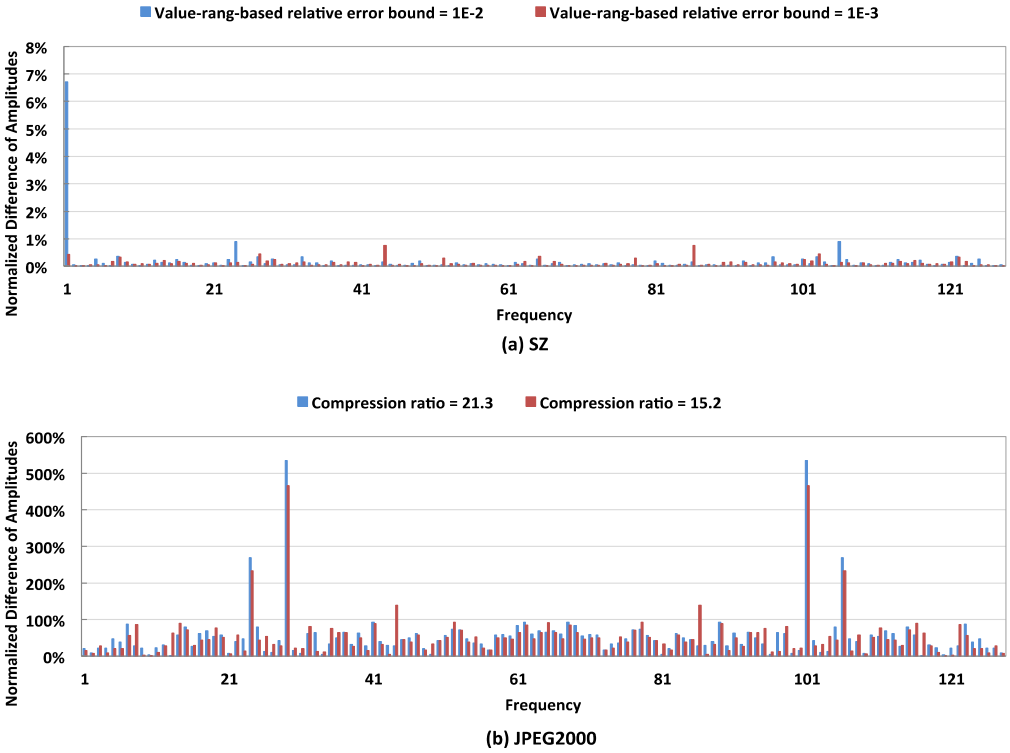

Figure 14 Normalized difference of DFT's amplitudes on the original and reconstructed calibrated dataset

Compression rate and decompression rate

In this section, we evaluate the compression/decompression speed of SZ compressor on

the EXAFEL datasets. Specifically, we use compression rate and decompression rate to assess how fast the compression/decompression is performed. Compression rate is defined as the number of bytes in the original data stream (or data file) to be processed per second. Decompression rate is defined as the number of bytes to be reconstructed per second. Figure 15 shows the compression and decompression rate on the dark, raw, and calibrated dataset with the value-range-based relative error bound 10^{-2} and 10^{-3}. Note that we count the compression/decompression time without I/O and we use the size of single floating-point data rather than the uint_16 or int_16 data size.

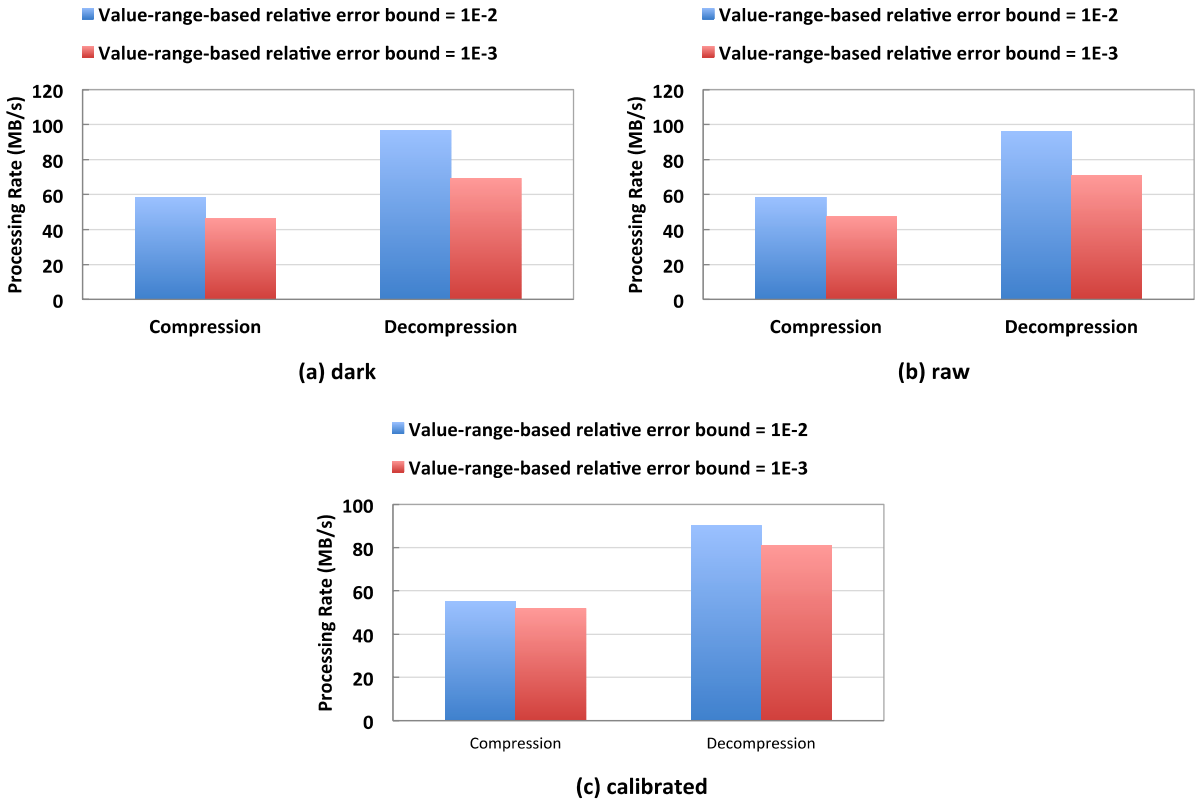

Figure 15 Compression and decompression rate on the (a) dark, (b) raw, and (c) calibrated dataset with value-range-based relative error bound 10^{-2} and 10^{-3}

Conclusion

In this report, we present the compression results of SZ compressor working on EXAFEL data sets, as compared with some other compressors such as Gzip and JPEG2000. We have the following findings:

1. The lossless compressor Gzip can only get a compression factor of 1.5-1.9, while SZ can get the compression factor to 3.6-238, depending on the compression error bounds.
2. SZ's rate-distortion is slightly worse than that of JPEG2000 on dark and raw data sets,

while its rate-distortion is far better than that of JPEG2000 on the calibrated data set. Specifically, for the calibrated data set, SZ requires only 1.7 bits per data point to get PSNR of 50 while JPEG2000 needs about 7 bits per data point.

3. Auto-correlation of the compression errors under SZ are very small (around 0.01) for the dark and calibrated data sets, when the error bound is set to 0.01 during the compression (as shown in Figure 5). However, the auto-correlation of errors is relatively large (around 0.1 or so) for the raw data set. In fact, the auto-correlation of the compression errors will decrease with decreasing error bounds, depending on users' requirement.

4. Distribution of compression errors of SZ generally is uniform in most cases. However, there are some exceptions as shown in Figure 8(a) and Figure 11(a). The reason why the compression errors do not follow uniform distribution is that the compression error bound set by users there (abs_err=63) seems too huge for most of the data, such that the quantization step may not have random compression errors in the quantization bins. This could be fixed by reducing the compression error bound or improving the prediction accuracy, to be studied in our future work.

5. Spectrum analysis shows that the FDT results of SZ between initial data sets and decompressed data sets change less than that of JPEG2000. Specifically, for calibrated data set, SZ leads to less than 1% distortion, while JPEG2000 suffers from 500% distortion, as shown in Figure 14.

6. As shown in Figure 15, decompression speed (60-90MB/s) is much faster than compression speed (50-60MB/s).

References

[1] SZ library. Online available at https://collab.mcs.anl.gov/display/ESR/SZ.

[2] Half-precision floating-point format. Online available at https://en.wikipedia.org/wiki/Half-precision_floating-point_format

[3] L. P. Deutsch. Gzip file format specification version 4.3.

[4] C. Christopoulos ; A. Skodras ; T. Ebrahimi, "The JPEG2000 still image coding system: an overview," in IEEE Transactions on Consumer Electronics, Volume: 46, Issue: 4, Nov 2000.

[5] OpenJPEG library. Online available at http://www.openjpeg.org/