Improving Statistical Similarity Based Data Reduction for Non-Stationary Data

Dongeun Lee
Texas A&M University-Commerce
Department of Computer Science
Commerce, Texas 75428, USA
dongeun.lee@tamuc.edu

Alex Sim
Lawrence Berkeley National Laboratory
Data Science and Technology Department
Berkeley, California 94720, USA
asim@lbl.gov

Jaesik Choi
Ulsan National Institute of Science and Technology
School of Electrical and Computer Engineering
Ulsan 44919, Korea
jaesik@unist.ac.kr

Kesheng Wu
Lawrence Berkeley National Laboratory
Data Science and Technology Department
Berkeley, California 94720, USA
kwu@lbl.gov

ABSTRACT

We propose a new class of lossy compression based on locally exchangeable measure that captures the distribution of repeating data blocks while preserving unique patterns. The technique has been demonstrated to reduce data volume by more than 100-fold on power grid monitoring data where a large number of data blocks can be characterized as following stationary probability distributions. To capture data with more diverse patterns, we propose two techniques to transform non-stationary time series into locally stationary blocks. We also propose a strategy to work with values in bounded ranges such as phase angles of alternating current. These new ideas are incorporated into a software package named IDEALEM. In experiments, IDEALEM reduces non-stationary data volume up to 100-fold. Compared with the state-of-the-art lossy compression methods such as SZ, IDEALEM can produce more compact output overall.

CCS CONCEPTS

• **Theory of computation** → **Data compression**; • **Mathematics of computing** → Bayesian nonparametric models; Time series analysis; • **Information systems** → Data streaming; Similarity measures;

KEYWORDS

Floating-point data, locally exchangeable measure, lossy compression, online algorithm, time series data

ACM Reference format:
Dongeun Lee, Alex Sim, Jaesik Choi, and Kesheng Wu. 2017. Improving Statistical Similarity Based Data Reduction for Non-Stationary Data. In *Proceedings of SSDBM '17, Chicago, IL, USA, June 27-29, 2017,* 7 pages.
https://doi.org/10.1145/3085504.3085583

SSDBM '17, June 27-29, 2017, Chicago, IL, USA

ACM ISBN 978-1-4503-5282-6/17/06…$15.00
https://doi.org/10.1145/3085504.3085583

1 INTRODUCTION

Lossy compression reduces data storage requirement by dropping some information in the original data. The quality of these techniques is currently measured by the Euclidean distance between the original data and the reconstructed data. Though this approach is typically effective, there are cases where introducing an alternative quality measure would produce better compression. In general, compression takes advantage of repeated patterns or common features in the original data; thus high entropy data is hard to compress. In many applications, only the probability distribution of the high entropy data is important, in which cases, capturing the probability distribution accurately is sufficient for the analysis tasks. More generally, we propose to use the locally exchangeable measure (LEM) [3] to decide whether two data blocks could be used interchangeably. When some blocks are interchangeable, we could keep only a single copy and therefore reduce the storage requirement. Preliminary study of LEM shows that this approach is promising [13], but there are also noticeable shortcomings that we will address.

Our aim is to compress floating-point values produced from large scientific simulations or large experiments. Numerical values, especially floating-point values, are known to be hard to compress [1, 10]. Recently, there has been a flurry of publications on compressing floating-point values [1, 4, 8, 10, 14]. However, all these techniques are designed to reduce the Euclidean distance between the original data and the reconstructed data. Among these, SZ [4] and ZFP [14] are able to reduce the data volume by more than 100-fold while maintaining good quality. In both cases, SZ and ZFP rely heavily on continuity present in data. However, there are many applications where such continuity is not present; instead, the most common feature in the data might be the apparent randomness.

Given two data blocks, LEM defines a statistical similarity measure to decide whether or not the two blocks are exchangeable. Following the general design of a dictionary-based compression approach [16, 18, 23], when an incoming data block is found to be similar to an existing one, we maintain a pointer to the existing block. This approach works well when there are many data blocks with the same empirical probability distributions. When a data

block is different from all others, it is stored precisely as is; thus the unusual features in the original data are preserved.

A simple way to decide whether two blocks are exchangeable is to compare their distributions. When their distributions match, we say they are similar enough to be exchanged. In other words, for our LEM based compression method to be effective, the probability distribution of original data must be stationary. However, many real-world data sets are not stationary. For example, outdoor temperature measurements have seasonal trends, and the phase angle of electricity data is carefully managed so that it can only drift slowly over time. In our preliminary tests, our implementation of the compression software named IDEALEM was found to be able to reduce the storage requirement by a factor of more than 100 for a set of voltage and current recorded by μPMUs[1] [13]. However, we also noted that IDEALEM was not effective in compressing the phase angles in the same data set. One key motivation of this work is to explore options for IDEALEM to compress some forms of non-stationary values as well as stationary random values. In particular, we propose to explore a number of transformations that could turn some non-stationary time series into stationary time series.

The main contributions of this work are as follows:

- We propose two different transformation methods, named residual transformation and delta transformation, to capture long-range trends in data and allow local variations to be compared through LEM.
- Values in bounded ranges, such as angles between 0° and 360°, are also considered.
- We carefully modify the IDEALEM software to implement the above methods [20], and conduct extensive tests of the resulting software on the phase angles known to be hard to compress for the original IDEALEM software. Tests show that the new IDEALEM can reduce the storage requirement by nearly 100 fold in many cases. Furthermore, the new methods significantly outperform the state-of-the-art lossy compression methods.

2 RELATED WORK

Since lossless compression methods are typically not able to significantly compress floating-point values, we focus on lossy compression techniques. Conventional lossy compression schemes typically quantize or threshold data to adjust quality and reduce storage requirement [18]. They measure compression quality using ℓ_2 norms (Euclidean distance), such as mean squared error (MSE) and signal-to-noise ratio (SNR) [2, 11, 12, 18]. These compression methods are unable to effectively reduce the data size when incoming data have high entropy or are floating-point values. Intuitively, high entropy data can be regarded as random and floating-point values often have unpredictable variations in their lower order bits. Since the existing compression methods could not find patterns among the values in these cases, we decided to explore the patterns in probability distributions instead. Following a statistical concept known as exchangeability, we define a statistical similarity measure named locally exchangeable measure (LEM) [3].

[1]More information about the data could be found at http://powerdata.lbl.gov/.

Overall, our LEM-based compression technique resembles a dictionary-based methods [16, 18] in lossless compression. In IDEALEM, data values are reconstructed from learned probability distributions during the encoding process, not from the encoded (quality-adjusted) data itself. The encoded output is not a direct representation of the original data; instead, the encoder informs the decoder how to regenerate them. Thus, this approach can be also regarded as one of the analysis/synthesis schemes [9, 18, 21], in the sense that they rely on the self-similarity of data, suggesting parts of data often resemble other parts of the same data.

Research on the compression of floating-point data has been very active. Here, we briefly review *ZFP* [14, 15], *ISABELA* [7, 10], and *SZ* [4, 5]. ZFP is designed to work with multi-dimensional floating-point arrays commonly used in scientific simulations. ZFP can reduce the data size by 100 fold or more on some three-dimensional arrays while maintaining high reconstruction quality. However, on one-dimensional arrays ZFP do not perform very well [13].

ISABELA [10] sorts the incoming data blocks and uses the B-spline to approximate the sorted values. Since the splines can be represented compactly, the overall storage requirement may be reduced. However, it also needs to store a representation of the sorting order, which limits its compression performance.

SZ (squeeze) [4] is a recently developed lossy coding scheme with superior performance. Similar to ISABELA, SZ employs curve fitting; but it provides three curve-fitting models and chooses the one that best predicts each target value. The authors of SZ have demonstrated it to be more effective than all known lossy compression methods [4].

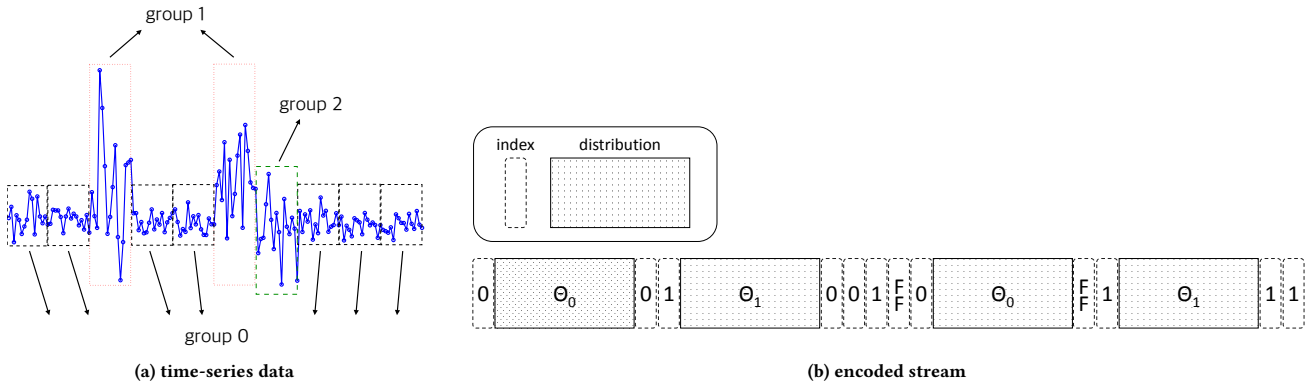
3 IDEALEM COMPRESSION ALGORITHM

Various application scenarios might be considered as generating random numbers. For example, internet traffic might be regarded as random and sensors monitoring many phenomena might be recording background noise while waiting for interesting events. For such noisy data blocks, a compression method that reproduces the same distribution would be sufficient for many analysis operations. To explore the potential of this approach, we developed Implementation of Dynamic Extensible Adaptive Locally Exchangeable Measures (IDEALEM) [20]. Our preliminary tests showed that IDEALEM was able to significantly reduce the storage requirement while preserving the "interesting" features in the electric power grid monitoring data [13]. Next, we outline the key concepts in IDEALEM.

3.1 IDEALEM Data Blocks

IDEALEM treats an incoming data series as a sequence of fix-sized blocks. The basic approach of IDEALEM follows the dictionary-based compression, where each dictionary entry represents a user-specified block of incoming data. Let B denote the block size, y_i the incoming values ($i = 0, 1, 2, \ldots$). Then the block j ($j = 0, 1, 2, \ldots$), denoted by b_j, would include values $(y_{jB}, y_{jB+1}, \ldots, y_{jB+B-1})$. When two blocks are similar, the newer block would be replaced with the existing block.

Fig. 1a shows 160 values broken into ten 16-element blocks, and the ten blocks are split into three groups based on their empirical



(a) time-series data

(b) encoded stream

Figure 1: (a) An example of 160 data values broken in blocks of 16 values each. IDEALEM treats blocks with a similar probability distribution as "exchangeable" and only records one copy of each group of "exchangeable" blocks. (b) The encoded stream structure of (a) with $B = 16$ and $D = 2$ ($j = 0, 1$): a dotted box represents an index (pointer) in 1 byte; a solid box with pattern represents a source distribution (dictionary entry) in $8B$ bytes (128 bytes); `0xFF` denotes a special marker for overwriting signal.

probability distributions. The IDEALEM software records a representative from each group as a dictionary entry, and a pointer to the dictionary entry for each of the remaining blocks. IDEALEM creates a new dictionary entry when the incoming data block is different from all dictionary entries. During the reconstruction process, it copies the dictionary entry when a new block is encountered, which allows to keep unique blocks exactly as in the original data, while reconstructing the repeating blocks through a random process.

3.2 Dictionary Size

IDEALEM software keeps a small number of dictionary blocks in memory during encoding and decoding process. We limit the number of dictionary entries to be no more than 255, which allows us to use a byte as the pointer to a dictionary entry.

In Fig. 1a, we have identified three groups of blocks. Ideally, we would like to store a dictionary entry for each of the groups in memory. Fig. 1b illustrates how we might record the blocks if only two dictionary blocks are allowed to be kept in memory ($D = 2$). In this case, the seventh block, i.e., group 2, overwrites an existing dictionary entry. IDEALEM uses a special marker `0xFF` to denote that the oldest dictionary entry has been overwritten. Since the value `0xFF` is used for this special purpose, the maximum dictionary size is 255 ($j = 0, 1, \ldots, 254$).

The dictionary size D plays an important role in compression performance: more entries in general promise higher compression ratios because there is a higher chance of finding a similar distribution stored in the dictionary when we encounter a new data block. Increasing D may also have drawbacks: (i) it increases memory usage; (ii) it increases the number of similarity tests needed since the new data block is compared against each dictionary entry.

3.3 Measuring Similarity with Kolmogorov-Smirnov Test

We measure the similarity between two data blocks with Kolmogorov-Smirnov test (KS test) [17, 19, 22]. KS test is a popular choice for testing similarity and is also relatively simple and fast to compute as compared to other well-known statistical tests. It compares empirical distributions of two time series, computes the maximum distance between the two distributions, and then normalizes the distance into the p-value. A large p-value means the two time series are more likely to be produced from the same probability distribution. In our work, when the p-value is larger or equal to a user-specified parameter α, i.e., the KS test threshold, we declare the two data blocks to be similar.

4 IMPROVING IDEALEM

This work continues our development of IDEALEM. In the previous study of IDEALEM [13], we have identified a number of cases where it does not work very well: when data values contain trends (e.g., cyclostationary process) and the values are in bounded ranges. Next, we outline our approach to address these challenges.

4.1 Transforming Data Blocks

The similarity test in IDEALEM declares two data blocks to be the "same" when their empirical distributions are close to each other. Assuming data blocks are generated by random processes, compressible time series must be generated from stationary processes. For those time series from non-stationary processes, we seek to transform the data blocks so that we can reuse the basic mechanism of IDEALEM. Our basic design is to capture the long-range trends explicitly by keeping one value from each block, which we call the base value. We transform the rest of values of the block relative to this base value. Next, we describe two different transformation strategies named residual transformation and delta transformation.

4.2 Residual Transformation

Recall that we use the notation b_j to represent the block with the following values: $y_{jB}, y_{jB+1}, \ldots, y_{jB+B-1}$, where B is the block size. The residual transformation records a base value, say y_{jB} of b_j and transforms the remaining values of the block as follows: $y^r_{jB+k} = y_{jB+k} - y_{jB}$, where $k = 1, 2, \ldots, B-1$. In this case, the similarity

comparison is performed on $b_j^r \equiv (y_{jB+1}^r, y_{jB+2}^r, \ldots, y_{jB+B-1}^r)$. In most non-stationary processes, we expect the probability distribution within $B-1$ values to be relatively stable; therefore, it is possible that b_j^r could be similar to b_k^r.

4.3 Delta Transformation

It is possible that probability distribution changes rather quickly, in which case, the successive difference among y_i might follow a stationary process. Again, using y_{jB} as the base value of block b_j, the delta transformation computes a new block consisting of $y_{jB+k}^d = y_{jB+k} - y_{jB+k-1}$, where $k = 1, 2, \ldots, B-1$. In this case, the similarity comparison is performed on $b_j^d \equiv (y_{jB+1}^d, y_{jB+2}^d, \ldots, y_{jB+B-1}^d)$.

Fig. 2 shows the residual and delta transformations of 256 sample values. The original values in Fig. 2a are a small subset of the angular values that are difficult to compress using the original IDEALEM [13]. After the transformations (see in Fig. 2b and Fig. 2c), the new data blocks appear more likely to be similar to each other, and therefore could be more compressible.

4.4 Values in Bounded Ranges

Values such as the phase angles of alternating current have bounded ranges; the encoding and decoding procedures of a compression method should respect the ranges. For instance, the phase angle of electricity data has a range of 0° to 360°, which is a periodic variable.

In order to handle the values in bounded ranges, IDEALEM has a provision for controlling the range of encoded and decoded data in the residual/delta transformation. Range minimum *rmin* and range maximum *rmax* are two parameters that define the minimum and maximum values of a variable. Using these values, the encoding process assures that residual/delta values are within the range of *rmin* to *rmax*; similarly, the decoding process also wraps all values outside of the range to be within the range.

5 EXPERIMENTAL RESULTS

We employ a set of power grid monitoring data from μPMUs installed on-site at Lawrence Berkeley National Laboratory (LBNL) for experiments.[1] In this test, we only use the phase angles that were found to be hard to compress for the original IDEALEM software. This data set contains 12 variables representing phase angles of current (C) and line voltage (L) of three phases of the alternating electricity (marked with 1, 2, 3) captured by two different μPMUs (A6BUS1 and BANK514). The recorded values are about half a month's recording, and contain nearly 1 GB for each time series.

Experiments were conducted on a server equipped with Intel Xeon X3450 (2.66 GHz) CPU, 8 GB RAM, and 256 GB SSD, which runs Ubuntu 10.04.4 LTS (Linux kernel 2.6.32-57-server). We compare the performance of IDEALEM with the state-of-the-art compression algorithms described in Section 2 including ZFP [15], ISABELA [7], and SZ [5]. We also present the results of *gzip* [6] for reference, which is a popular lossless coding scheme that can handle general data types [18, 23].

With either the residual or delta transformation, IDEALEM can theoretically achieve a compression ratio up to $(8/9)B$. In the best case, if all data blocks have the same empirical distribution, then we can represent each data data block with a pointer (1 byte) plus a corresponding base value (8 bytes), which reduces the storage requirement for a block from $8B$ bytes to 9 bytes. In general, a larger block size B increases the compression ratio; but a large B also increases the difficulty of passing the KS test due to the sensitivity with the number of samples [13], which reduces the compression ratio. Since there is no theoretical analysis telling us the optimal block sizes to use, we plan to explore the choice of B through an empirical study.

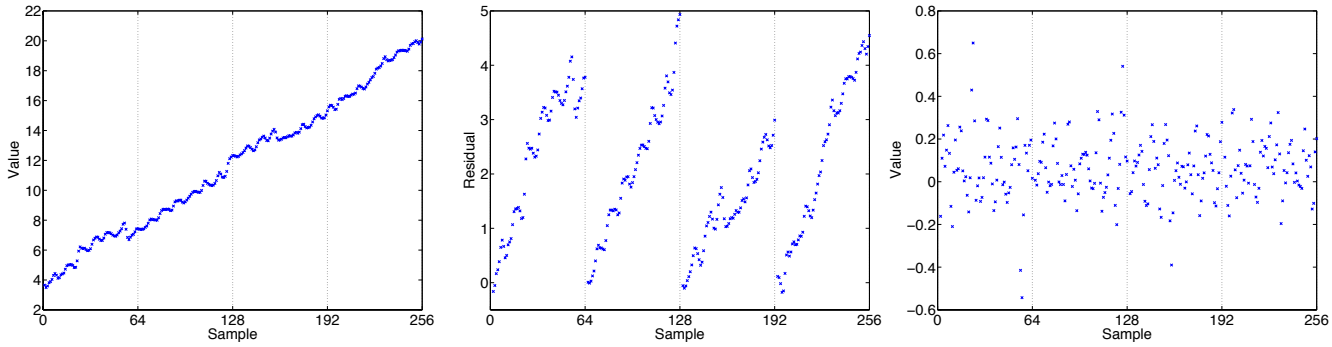
Fig. 3 shows the compression ratios of IDEALEM in the residual and delta transformations. The optimal choice of B depends on each time series. When block sizes are small, compression ratios of all time series are almost identical to the maximum theoretical compression ratio for given B. As B grows, these compression ratios diverge from the maximum compression ratio and eventually reach their maximum values. However, these curves show different shapes depending on time series and transformation methods, though they are similar among three-phase measurements within each transformation method. For instance, compression ratios are relatively similar to each other in the residual transformation in Fig. 3a; whereas they are dramatically different from each other in the delta transformation in Fig. 3b. In particular, six current (C) time series in Fig. 3b have very high compression ratios, which keep increasing till $B \sim 688$ where compression ratios become higher than 400.

Overall, the angle values change relatively slowly over time, which allows methods such as SZ to work quite well. Nevertheless, as illustrated in Fig. 2, there are small "random" fluctuations in the angle values and the delta transformation produces values that look more "random" especially for current (C) data, which makes them somewhat easier to pass the KS test and increases the compress ratio. The KS test with the delta transformation mode is susceptible to the noisiness (variance) of data and produces totally different results depending on time series.[2] On the contrary, the KS test with the residual transformation mode shows more robust results in compression ratios, regardless of data characteristics.

We present the results of the compression ratio and the execution time of IDEALEM along with other compression algorithms in Table 1 and Table 2, respectively. We selected a set of parameters for each compression method that performed *best* on our data set of first phase (1) measurements, in both compression ratio and reconstruction quality. Specifically, gzip uses the compression level `-5`; ZFP uses the tolerance parameter `-a 8`; ISABELA uses the window size 512, the number of coefficients 15, the error rate 5, and the BSplines switch; SZ uses the error bound `errorBoundMode=REL` and the ratio `relBoundRatio=0.001`; the original IDEALEM [13] uses $B = 32$, $D = 255$, and $\alpha = 0.01$; both the residual and delta transformations use $B = 112$, $D = 255$, and $\alpha = 0.01$.

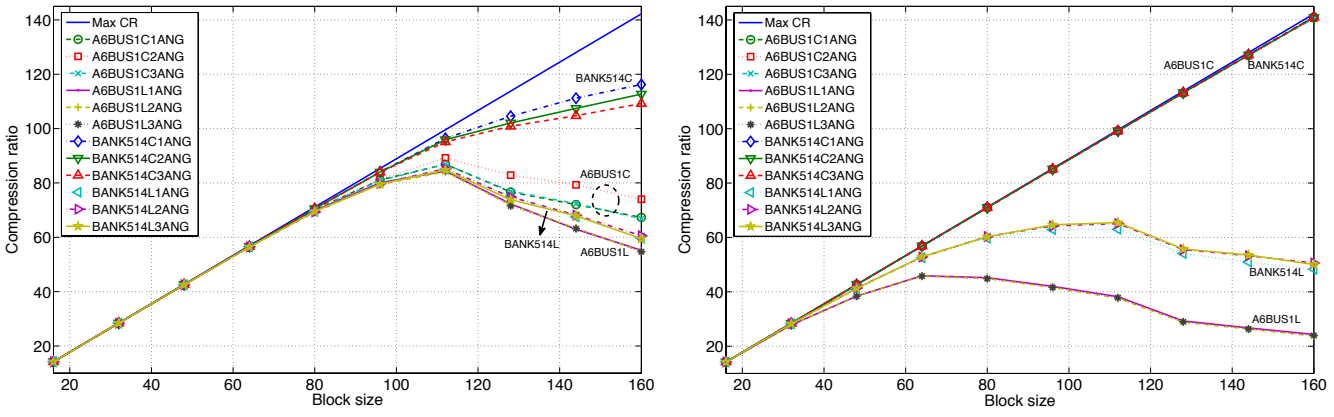
In Table 1, the compression performance of IDEALEM has been improved a lot from the original version with the addition of the residual/delta transformation, which is far beyond the performance of gzip, ZFP, and ISABELA. On the other hand, SZ (version 1.4.9.3-hacc) shows the best performance on voltage (L) data that are cleaner than current (C) data; whereas IDEALEM, especially delta

[2] Current (C) data has higher variances than voltage (L) data has.



(a) original (b) residual (c) delta

Figure 2: Scatter plot of 256 samples of angular values from an electric power grid data (BANK514C1ANG) with $B = 64$. Residual values of data blocks are shown in (b), and delta values in (c), both without base values (63 values for each data block).



(a) residual (b) delta

Figure 3: Compression ratios of IDEALEM with μPMU phase angle (ANG) data in residual and delta transformations. Max CR (compression ratio), the fundamental limit $(8/9)B$, is shown together for comparison. Overall, phase angles of current (C) data show higher compression ratios than those of voltage (L) data. Compression ratios with the delta transformation increase up to higher than 400 when $B \sim 688$.

Table 1: Compression Ratio

Data/Compression	gzip	ZFP	ISABELA	SZ(1.2)	SZ(1.4.9.3-hacc)	original	residual	delta
A6BUS1C1ANG	2.42	8.76	5.36	45.84	58.95	2.04	86.89	99.19
A6BUS1L1ANG	2.45	8.78	5.38	159.34	147.77	1.68	84.32	38.21
BANK514C1ANG	2.39	8.76	5.36	49.18	63.76	2.45	96.39	99.21
BANK514L1ANG	2.45	8.78	5.38	148.22	185.34	1.69	85.05	62.99
Overall	2.43	8.77	5.37	72.50	92.67	1.92	87.91	64.30

transformation, shows the best performance on the noisier current (C) data.

The improved compression performance of IDEALEM also affects the encoding and decoding time in Table 2, because IDEALEM can now easily find a similar distribution stored in the dictionary and this search finishes quickly, which is not the case for the original version where searching attempts generally fail after a full search. The high variance in the compression ratio of the delta transformation in Table 1 is closely connected with high variance

Table 2: Execution Time (s)

	Data/Compression	gzip	ZFP	ISABELA	SZ(1.2)	SZ(1.4.9.3-hacc)	original	residual	delta
Encoding	A6BUS1C1ANG	45.84	10.79	106.91	13.07	8.27	167.28	66.25	17.09
	A6BUS1L1ANG	48.48	11.34	105.02	11.44	9.04	172.24	42.14	77.68
	BANK514C1ANG	46.82	10.56	107.50	13.17	9.50	153.01	48.14	17.85
	BANK514L1ANG	49.57	10.75	104.23	11.56	8.83	169.98	43.56	79.32
	Total	190.71	43.44	423.66	49.24	35.64	662.51	200.09	191.94
Decoding	A6BUS1C1ANG	20.11	11.43	39.09	140.51	4.46	12.76	2.51	2.65
	A6BUS1L1ANG	18.08	12.32	39.13	138.34	4.32	18.21	6.55	10.73
	BANK514C1ANG	24.65	12.17	39.22	143.76	8.33	17.27	10.05	9.38
	BANK514L1ANG	20.71	11.07	39.19	138.13	5.98	17.92	7.28	10.45
	Total	83.55	46.99	156.63	560.74	23.09	66.16	26.39	33.21

in its encoding time: higher compression ratios lead to shorter encoding time. In contrast to IDEALEM, the execution time of ZFP does not vary much between encoding and decoding, as it is not based upon dictionary searching. In Table 2, the most notable improvements are those of SZ. In the previous SZ (version 1.2) shows the shortest encoding time and the longest decoding time. As the previous SZ targets for in-memory data compression, its decoding time including file I/O using the actual implementation [5] was slower than the decoding time of memory-only computation reported in the paper [4], and the previous file I/O had outputs in text format only by default which contributes to more I/O time. The latest SZ follows the same data format as the input data, and the file I/O has been improved.

6 CONCLUSIONS

We reported a new lossy compression method based on statistical similarity, called IDEALEM, which has been demonstrated to compress many variables by more than 100-fold. This paper proposed new transformation methods that enabled to compress non-stationary data which were hard to compress for the original IDEALEM software. We also considered values in bounded ranges which are commonly found in periodic variables. Experimental results showed that new IDEALEM could compress non-stationary data by up to 100-fold.

ACKNOWLEDGMENTS

The authors gratefully acknowledge helpful discussions with Kristofer Bouchard, David Donofrio, Emma Stewart, Sean Peisert, Chuck McParland, Reinhard Gentz, Mahdi Jamei, Ciaran Roberts, and Sebastian Ainslie. This work was supported by the Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. This work was also supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science, ICT & Future Planning (MSIP) (NRF-2014R1A1A1002662, NRF-2014M2A8A2074096). This research used resources of the National Energy Research Scientific Computing Center. Dongeun Lee wishes to express his gratitude to Texas A&M University-Commerce for support.

REFERENCES

[1] Martin Burtscher and Paruj Ratanaworabhan. 2009. FPC: a high-speed compressor for double-precision floating-point data. *IEEE Trans. Comput.* 58, 1 (January 2009), 18–31.
[2] Emmanuel J. Candès and Michael B. Wakin. 2008. An introduction to compressive sampling. *IEEE Signal Process. Mag.* 25, 2 (March 2008), 21–30.
[3] Jaesik Choi, Kejia Hu, and Alex Sim. 2013. *Relational dynamic Bayesian networks with locally exchangeable measures.* Technical Report LBNL-6341E. Lawrence Berkeley National Laboratory.
[4] Sheng Di and Franck Cappello. 2016. Fast error-bounded lossy HPC data compression with SZ. In *Proc. Int'l Parallel Distrib. Process. (IPDPS '16).* 730–739.
[5] Sheng Di, Dingwen Tao, and Franck Cappello. 2017. SZ: fast error-bounded floating-point data compressor for scientific applications. (January 2017). Retrieved February 3, 2017 from https://collab.mcs.anl.gov/display/ESR/SZ
[6] Jean-Loup Gailly and Mark Adler. 2003. The gzip home page. (July 2003). Retrieved February 3, 2017 from http://www.gzip.org
[7] NCSU Big Data Analytics Group. 2013. ISABELA: effective in-situ compression of scientific data. (2013). Retrieved February 3, 2017 from http://freescience.org/cs/ISABELA/ISABELA.html
[8] Jeremy Iverson, Chandrika Kamath, and George Karypis. 2012. Fast and effective lossy compression algorithms for scientific datasets. In *Proc. Int'l Conf. Parallel Process. (Euro-Par '12).* 843–856.
[9] Arnaud E. Jacquin. 1992. Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Trans. Image Process.* 1, 1 (January 1992), 18–30.
[10] Sriram Lakshminarasimhan, Neil Shah, Stephane Ethier, Scott Klasky, Rob Latham, Rob Ross, and Nagiza F. Samatova. 2011. Compressing the incompressible with ISABELA: in-situ reduction of spatio-temporal data. In *Proc. Int'l Conf. Parallel Process. (Euro-Par '11).* 366–379.
[11] Dongeun Lee and Jaesik Choi. 2014. Low complexity sensing for big spatio-temporal data. In *Proc. Int'l Conf. Big Data (BigData '14).* 323–328.
[12] Dongeun Lee, Jaesik Choi, and Heonshik Shin. 2015. A scalable and flexible repository for big sensor data. *IEEE Sensors J.* 15, 12 (December 2015), 7284–7294.
[13] Dongeun Lee, Alex Sim, Jaesik Choi, and Kesheng Wu. 2016. Novel data reduction based on statistical similarity. In *Proc. Int'l Conf. Scient. Stat. Database Manag. (SSDBM '16).* 21:1–21:12.
[14] Peter Lindstrom. 2014. Fixed-Rate Compressed Floating-Point Arrays. *IEEE Trans. Vis. Comput. Graphics* 20, 12 (December 2014), 2674–2683.
[15] Peter Lindstrom. 2016. zfp & fpzip: floating point compression. (February 2016). Retrieved February 3, 2017 from http://computation.llnl.gov/projects/floating-point-compression
[16] Mark Nelson and Jean-Loup Gailly. 1996. *The Data Compression Book* (second ed.). Vol. 2. M&T Books.
[17] Céline Quinsac, Adrian Basarab, Jean-Marc Girault, and Denis Kouamé. 2010. Compressed sensing of ultrasound images: sampling of spatial and frequency domains. In *Proc. Int'l Workshop Signal Process. Syst. (SiPS '10).* 231–236.
[18] Khalid Sayood. 2012. *Introduction to Data Compression* (fourth ed.). Morgan Kaufmann.
[19] José Seabra and Joao Sanches. 2008. Modeling log-compressed ultrasound images for radio frequency signal recovery. In *Proc. Int'l Conf. Eng. Med. Biol. Soc. (EMBC '08).* 426–429.
[20] Alex Sim, Dongeun Lee, Kesheng Wu, and Jaesik Choi. 2016. IDEALEM. (November 2016). Retrieved February 3, 2017 from http://datagrid.lbl.gov/idealem

[21] Brendt Wohlberg and Gerhard De Jager. 1999. A review of the fractal image coding literature. *IEEE Trans. Image Process.* 8, 12 (December 1999), 1716–1729.
[22] Jiangsheng Yu, Stefano Ongarello, R. Fiedler, Xuewen Chen, Gianna Toffolo, Claudio Cobelli, and Zlatko Trajanoski. 2005. Ovarian cancer identification based on dimensionality reduction for high-throughput mass spectrometry data. *Bioinform.* 21, 10 (May 2005), 2200–2209.
[23] Jacob Ziv and Abraham Lempel. 1977. A universal algorithm for sequential data compression. *IEEE Trans. Inf. Theory* 23, 3 (May 1977), 337–343.