Optimizing Error-Bounded Lossy Compression for Scientific Data with Diverse Constraints

Yuanjian Liu

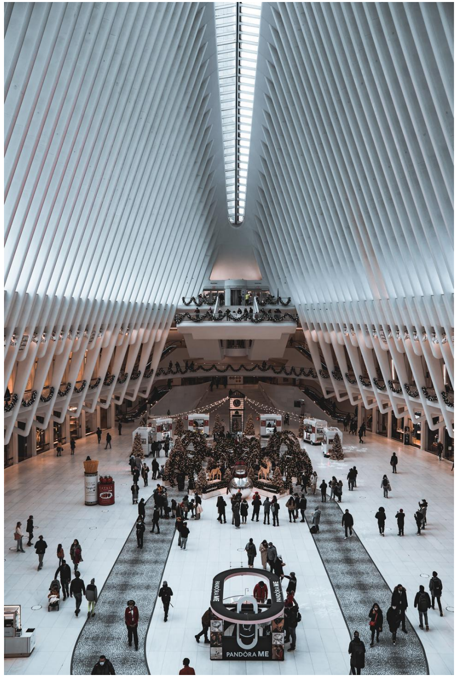
Committee: Kyle Chard, Ian Foster, Sheng Di

Master's Thesis Defense
Mar. 15th, 2022

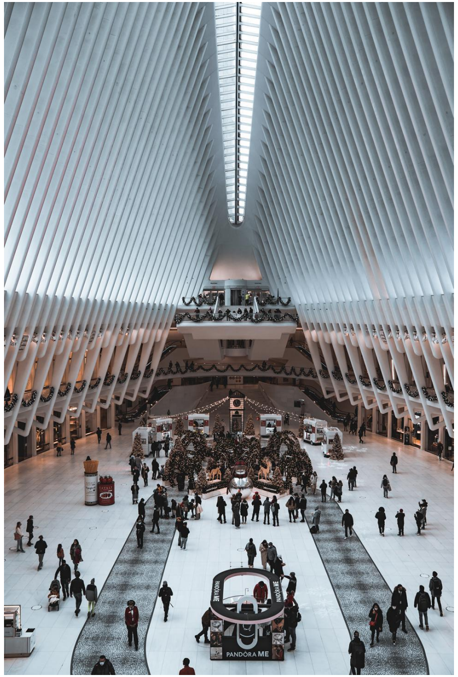
A story of the lossy compression

A story of the lossy compression

A story of the lossy compression



(A) RAW image 25M

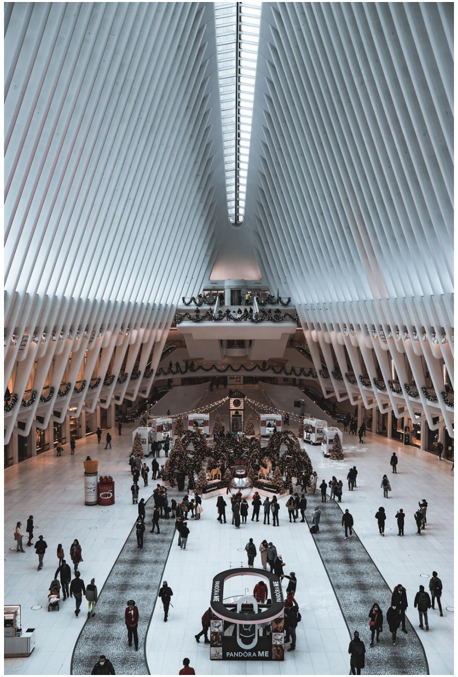


(B) JPEG image ~1M

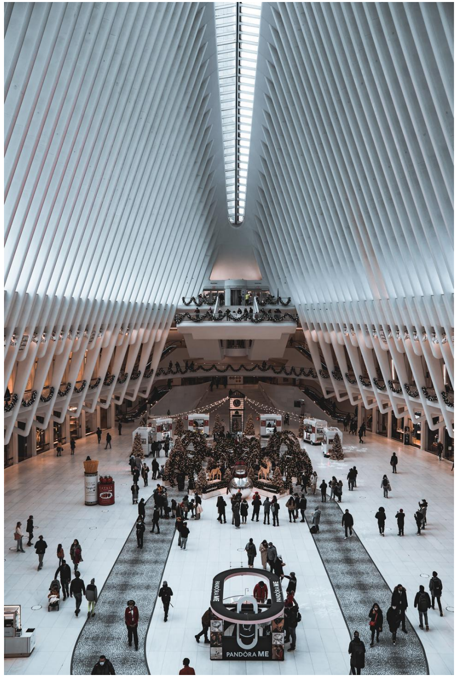
JPEG Compression

1. Video compression algorithms like H.264, which is common format on Youtube, share techniques found in JPEG compression.
2. Compression algorithms such as JPEG save servers Zettabytes of storage, resulting in billions of dollars in cost reduction.
3. JPEG is lossy compression

A story of the lossy compression



(A) RAW image 25M



(B) JPEG image ~1M

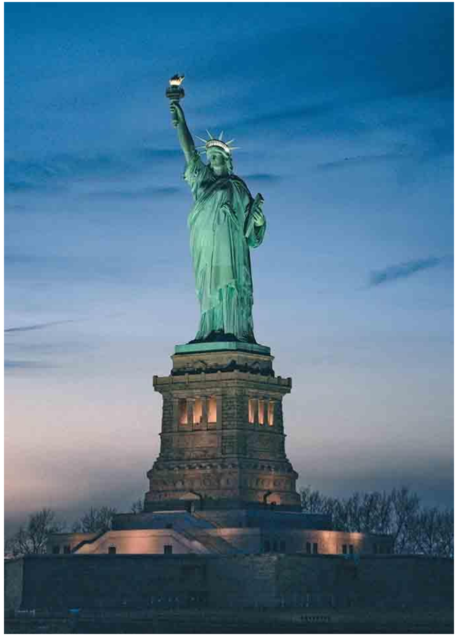
What does a lossy compression do?

JPEG goes through and analyzes each section of an image, finds and removes elements that human eyes cannot easily perceive.
JPEG allows users to set a 'quality' parameter

A story of the lossy compression

(A) Quality=12, size=872KB

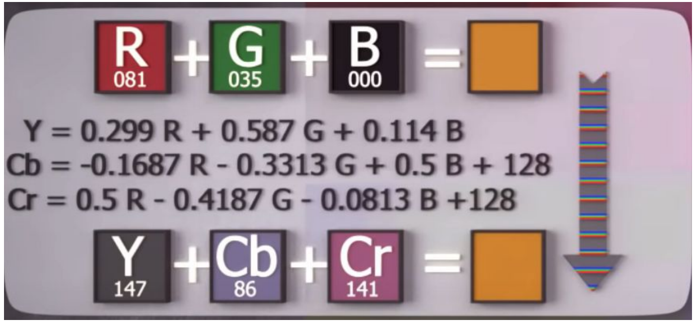
(B) Quality=5, size=80KB

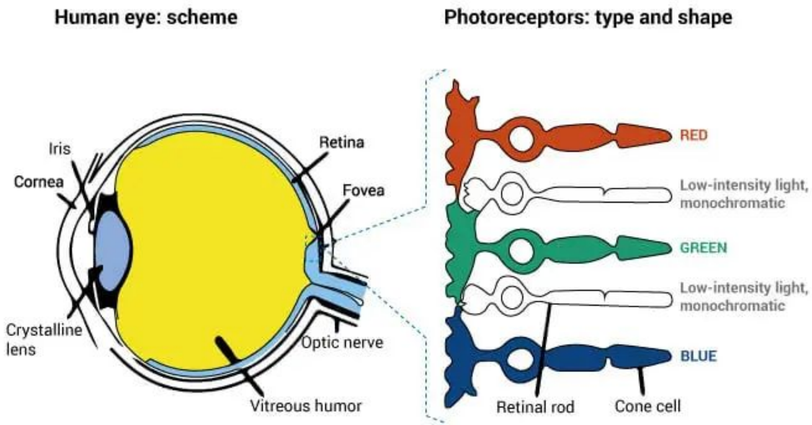


(C) Quality=1,size=32KB

A story of the lossy compression

There are 5 stages in the JPEG compression algorithm:
1. Color Space Conversion
2. Chrominance Downsampling
3. Discrete Cosine Transform
4. Quantization
5. Run Length and Huffman Encoding



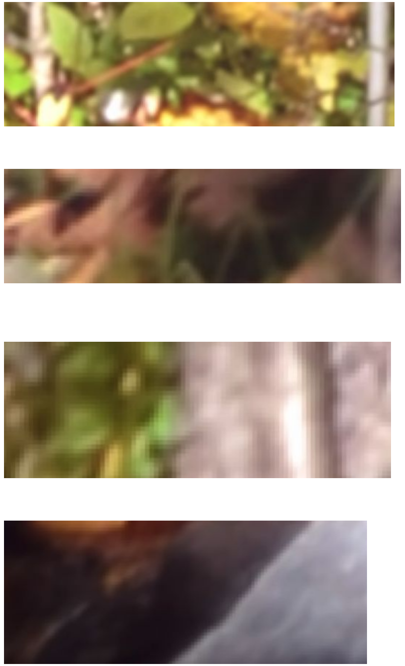


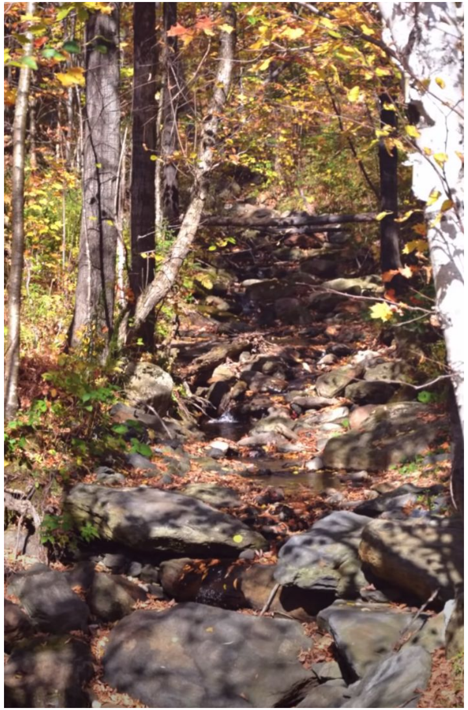
The reason why JPEG works:
- Human eyes are better at perceiving luminance (Ros), far less receptive at chrominance (Cones).

A story of the lossy compression

There are 5 stages in the JPEG compression algorithm:
1. Color Space Conversion
2. Chrominance Downsampling
3. Discrete Cosine Transform
4. Quantization
5. Run Length and Huffman Encoding

Human Eyes are bad at seeing high frequency elements: good at seeing edges, outlines, but bad at distinguishing high-frequency color data such as single blades of grass, individual leaves, etc.

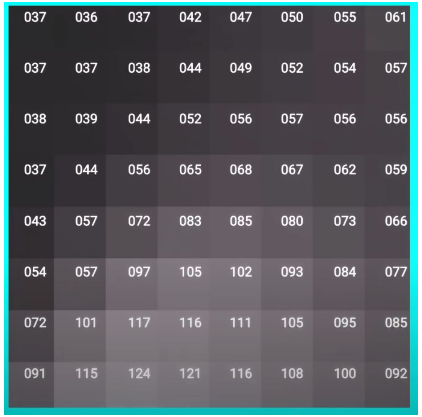


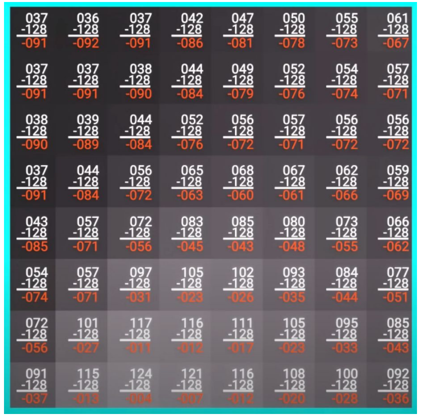


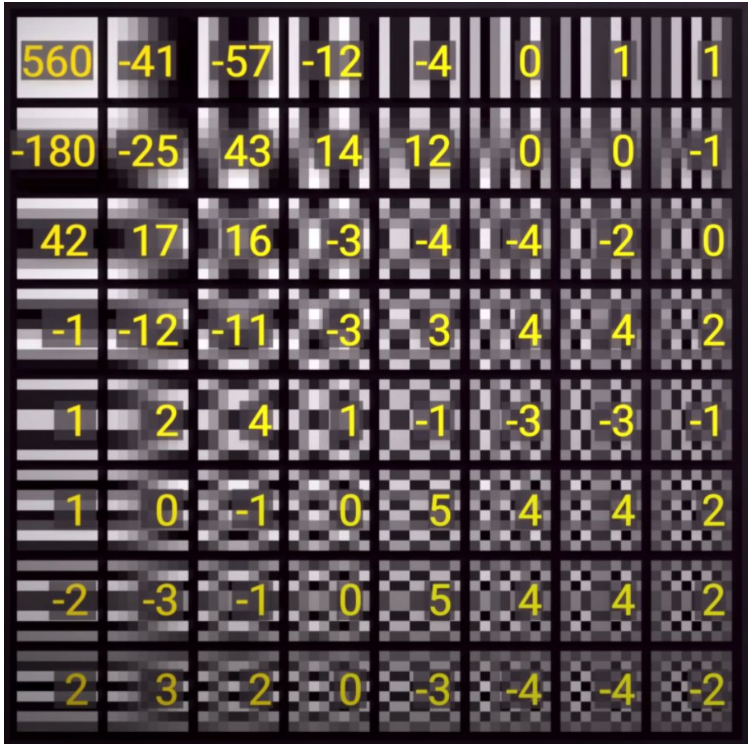
A story of the lossy compression

There are 5 stages in the JPEG compression algorithm:

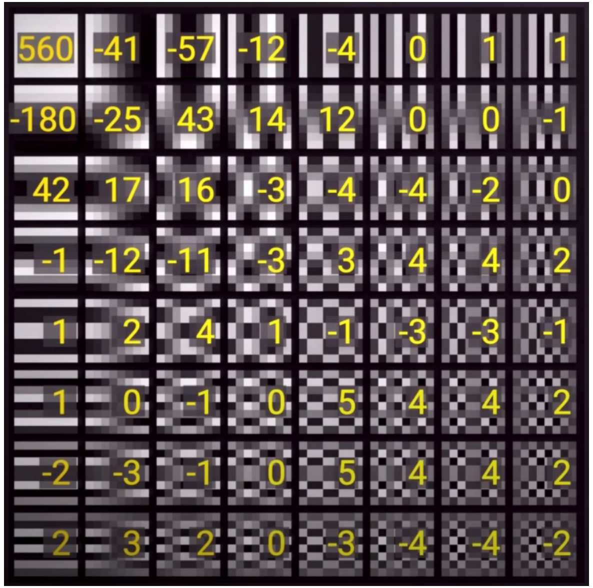
1. Color Space Conversion
2. Chrominance Downsampling
3. **Discrete Cosine Transform**
4. **Quantization**
5. Run Length and Huffman Encoding



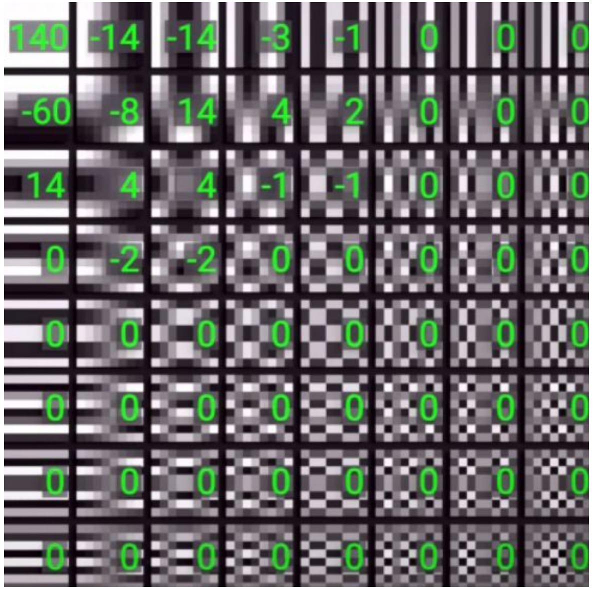




A story of the lossy compression



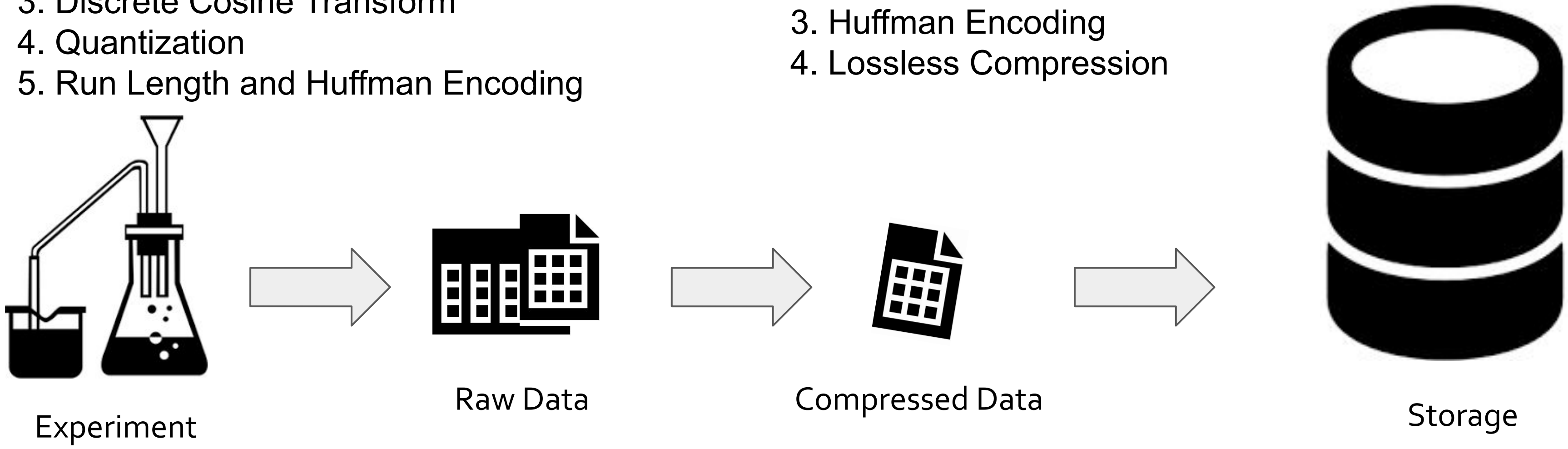
04	03	04	04	04	06	11	15
03	03	03	04	05	08	14	19
03	04	04	05	08	12	16	20
04	05	06	07	12	14	18	20
06	06	09	11	14	17	21	23
09	12	12	18	23	22	25	21
11	13	15	17	21	23	25	21
13	12	12	13	16	19	21	21



Apply similar ideas to scientific data

There are 5 stages in the JPEG compression algorithm:
1. Color Space Conversion
2. Chrominance Downsampling
3. Discrete Cosine Transform
4. Quantization
5. Run Length and Huffman Encoding

There are 4 stages in the SZ compression algorithm:
1. Prediction
2. Quantization
3. Huffman Encoding
4. Lossless Compression



The procedure of SZ compression

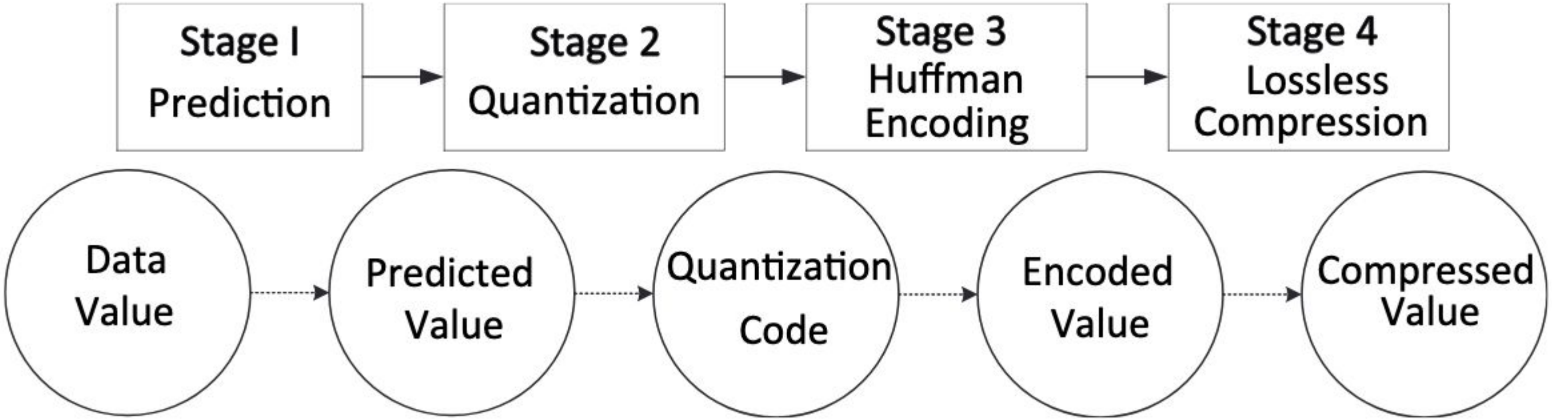
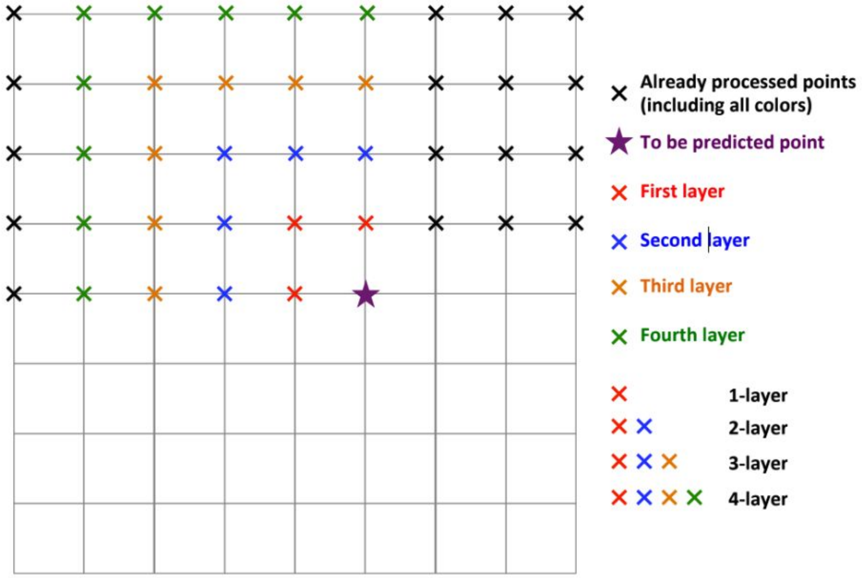


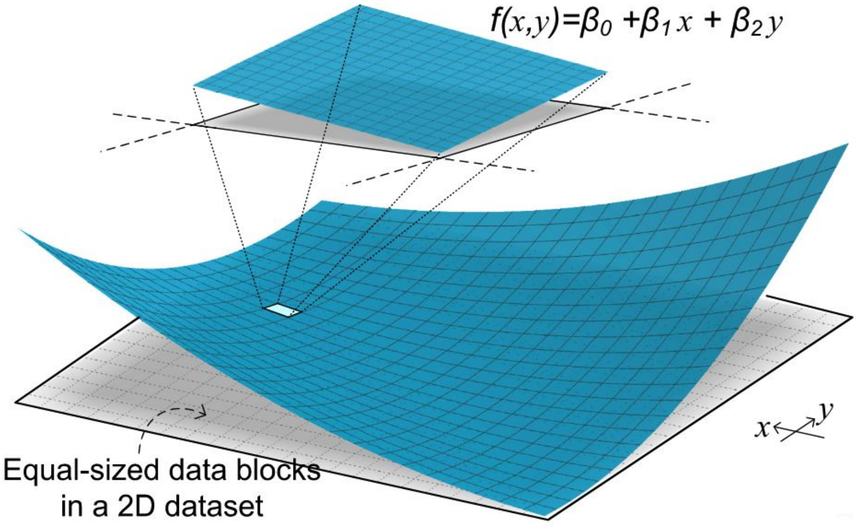
Figure 2.1: General procedure of prerequisite-preserving error-bounded lossy compression: Constraint (A) is handled before the prediction step; constraint (B) is handled primarily in both the prediction and quantization stage by replacing data points with Lorenzo-predicted values; constraints (C), (D), and (E) are addressed by designing a new quantization method.

Prediction Methods

(1) Lorenzo Prediction



(2) Linear Regression Prediction



Prediction Methods

(3) Interpolation Prediction

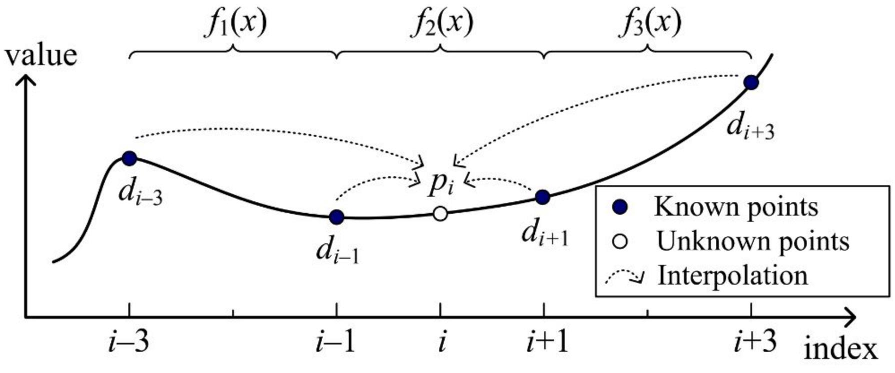


Figure 2.4: Illustration of Cubic Spline Interpolation

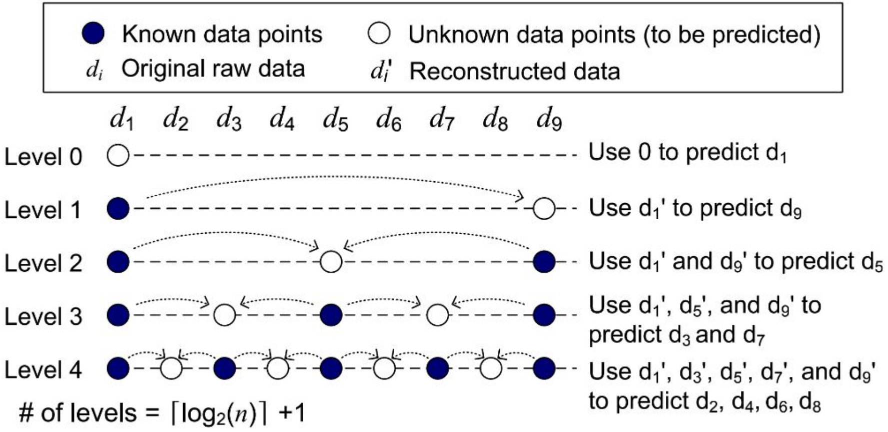
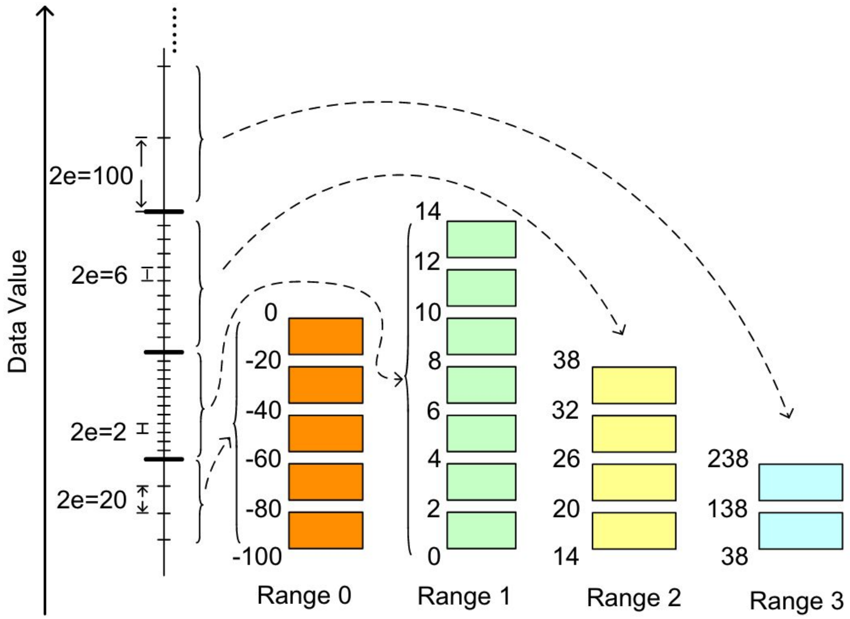
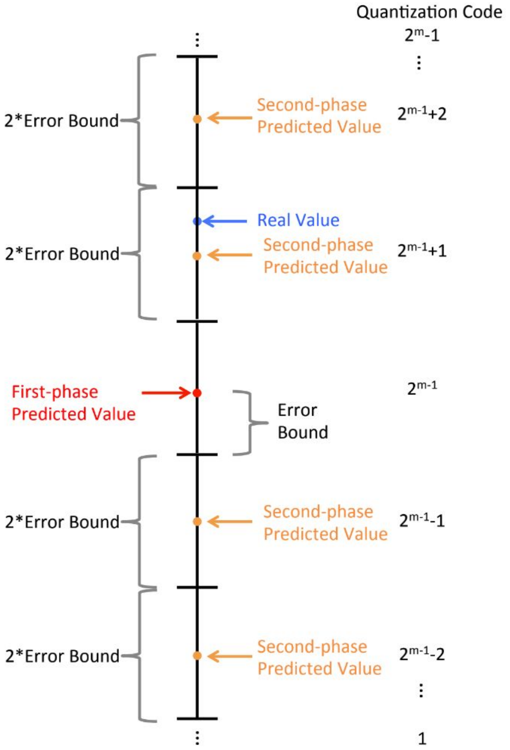


Figure 2.5: Illustration of Multilevel Linear Spline Interpolation

Quantization Methods



(A) Multi-range Quantization



(B) Single-range Quantization

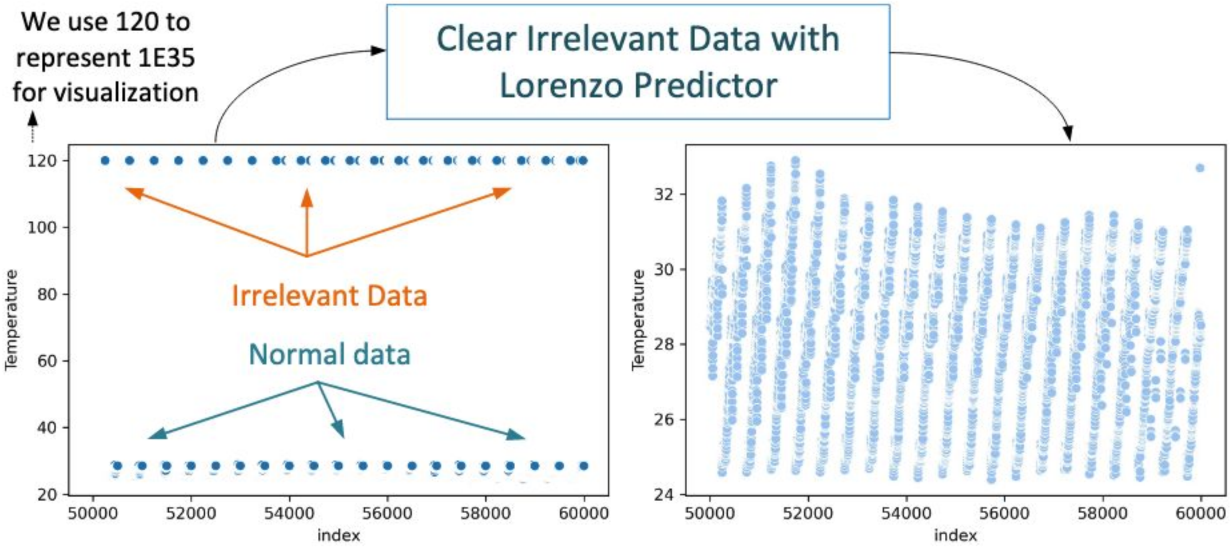
Diverse Constraints in Scientific Data

Table 3.1: Examples of user-required constraints applied to scientific simulation datasets

No.	User-Required Constraints	Science Domains
(A)	Isolating irrelevant value	Climate, Weather, etc.
(B)	Preserving global value range	Climate, etc.
(C)	Preserving value-interval-based error bounds	Weather, Cosmology, etc.
(D)	Preserving multiregion-based error bounds	Weather, Seismic imaging, etc.
(E)	Preserving irregularly shaped regions	Hydrodynamics, Weather, etc.

- This thesis proposes five constraints and the goal is to increase compression ratio by dropping some information while still respecting constraints.
- **Ideas to acheiving the goal given the five constraints**
  - (A) allows us to smoothen data by picking out irrelevant data;
  - (B) is a simple but necessary requirement for post-hoc analysis
  - (C)(D)(E) all allow us to lossen the error bound of a subset of the data

The Importance of the Constraints



Without cleaning the irrelevant data, the smoothness of data values will likely be distorted.

The irrelevant data constraint allows us to pick out them and use methods to substitute their value during compression so that we can deal with data with better continuity.

The Importance of the Constraints

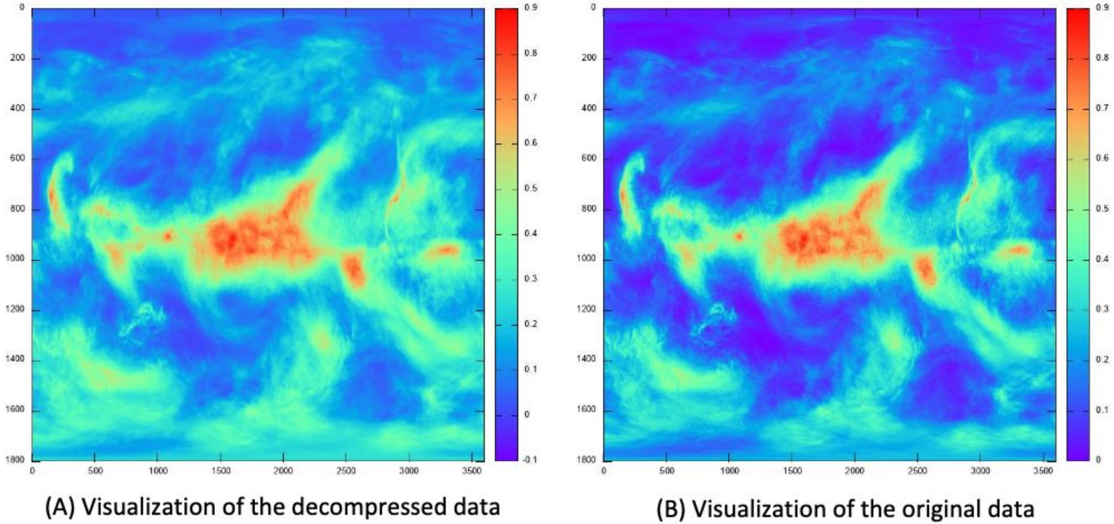


Figure 3.1: Without preserving the global range, the color mapping shifts, causing significant different visualization result compared to the original image.

The global range constraint is quite necessary in some analysis but not supported by existing compressors.

Without preserving the global range, the generated heat map will look very different compared to the original data because during compression, some points will have values lower than the global minimum or higher than the maximum.

Problem Formulation

(A) and (B) are the actual constraints we need to comply.

(C) (D) (E) give the possibilities to vary the error bounds during compression.

This thesis is the first to address the possibilities to vary error bounds during error-bounded lossy compression by proposing three different approaches in different circumstances.

$$\rho = \frac{N \cdot sizeof(dataType)}{Size_{compression}}, \tag{4.1}$$

$$\begin{aligned} &\text{Maximize } \rho \\ &\textit{subject to} \;\; \textit{user-required constraint} \end{aligned} \tag{4.2}$$

$$\text{CONSTRAINT (A): } \textit{Preserve and isolate } d_i \notin [R_{min}, R_{max}] \tag{4.3}$$

$$\text{CONSTRAINT (B): } \textit{Preserve} \begin{cases} \max(\widehat{d_i}) = high(r(D)) \\ \min(\widehat{d_i}) = low(r(D)) \end{cases} \tag{4.4}$$

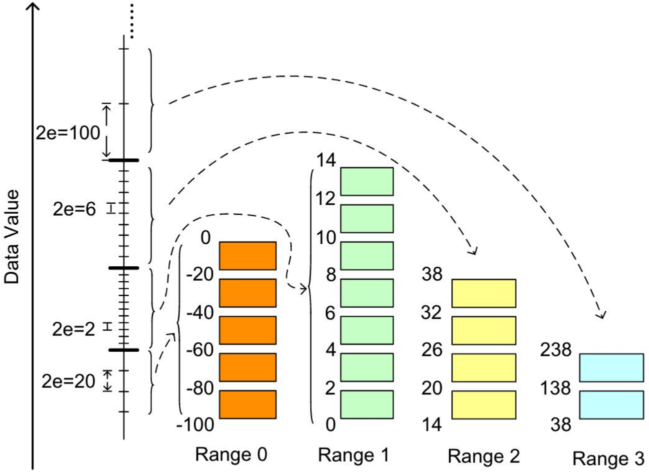
$$\text{CONSTRAINT (C): } |d_i - \widehat{d_i}| \leq e(d_i) \tag{4.5}$$

$$\text{CONSTRAINT (D, E): } |d_i - \widehat{d_i}| \leq e(LOC(d_i)), \tag{4.6}$$

Some Challenges

- Challenges exclusive to varying error bound compression
  - prediction methods like Linear Regression require blockwise compression. When the data are separated to blocks, if the neighboring blocks have different error bounds, the block boundary will be obvious, casuing artifacts in visualization results.
  - quantization code is just an integer. Without careful design, the compression and decompression stage can easily desynchronize and cause compression error.
  - to users, it is not always clear how to vary the error bounds.
- Solutions
  - This thesis proposes using interpolation prediction without the requirement to build data blocks to solve the blockwise artifact problem.
  - This thesis designs three algorithms to adapt to different use cases, and proposes a method to extract meaningful varying error bounds from the data.

How to vary the error bounds?



Method 1: Multiple value intervals. Each has its own error bound. The compressor uses different error bound according to data value.

Algorithm 1 MULTI-INTERVAL QUANTIZATION IN COMPRESSION STAGE

Input: user-specified intervals and error bounds ε
Output: compressed data stream in form of bytes

for each data point d_i **do**
  Use the composed prediction that combines Lorenzo predictor and linear regression predictor to obtain a prediction value p_i.
  $I_p \leftarrow r(p_i)$. /*Obtain interval index of p_i*/
  $I_d \leftarrow r(d_i)$. /*Obtain interval index of d_i*/
  if $I_d == I_p$ **then**
    $q \leftarrow round(\frac{(d_i - p_i)}{2e(I_d)})$./*Quantized distance between d_i & p_i.*/
  else if $I_d > I_p$ **then**
    $t = \sum_{i=I_p+1}^{I_d-1} \frac{l(i)}{2e(i)}$. /*Count bins for middle intervals.*/
    $t_p = round(\frac{high(I_p)-p_i}{2e(I_p)})$. /*Get quantized distance for I_p.*/
    $t_d = round(\frac{d_i - low(I_d)}{2e(I_d)})$. /*Get quantized distance for I_d.*/
    $q = t + t_p + t_d$. /*Get the logic quantization code.*/
  else
    $t = \sum_{i=I_d+1}^{I_p-1} \frac{l(i)}{2e(i)}$. /*Count bins for middle intervals.*/
    $t_p = round(\frac{high(I_d)-d}{2e(I_d)})$. /*Get quantized distance for I_d.*/
    $t_d = round(\frac{p_i - low(I_p)}{2e(I_p)})$. /*Get quantized distance for I_p.*/
    $q = t + t_p + t_d$. /*Get the logic quantization code.*/
  end if
  $q_s \leftarrow q + R$. /*Shift quantization code.*/
end for

How to vary the error bounds?

Method 2: Multiple regions. Each region has its own error bound. The compressor uses different error bound according to data indexes.

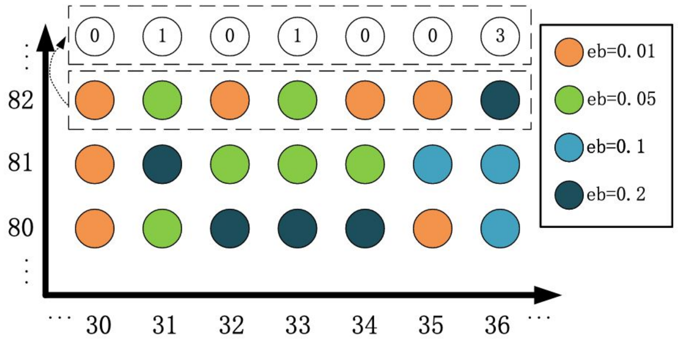


Figure 5.4: Illustration of bitmap error bound setting: Use an index to represent the error bound for each data point, and use a separate array to store all possible error bounds.

Method 3: Bitmap. Each data point corresponds to a position in the bitmap. The compressor uses the designated error bound for each data point.

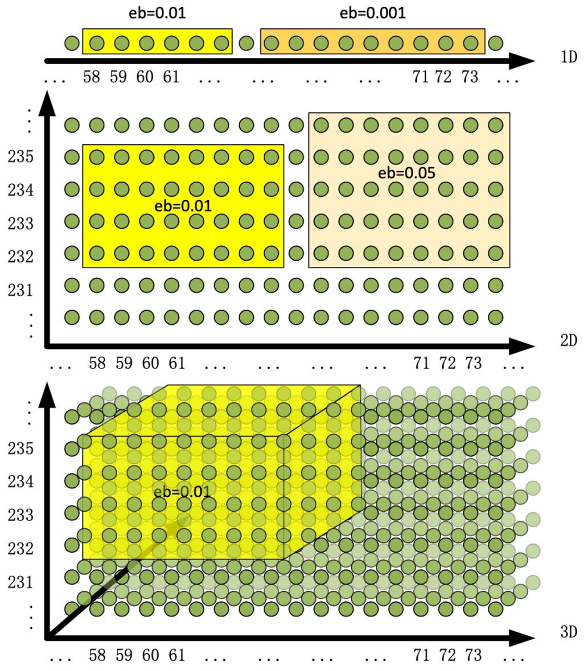


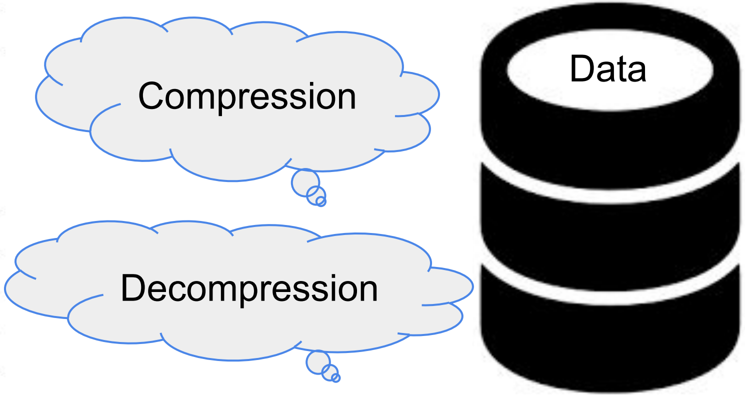
Figure 5.3: Constraint(D) region selection for 1D, 2D, and 3D data: In 3D cases, each region can be specified with seven parameters: the starting positions (3 parameters), the length of each direction (3 parameters), and the error bound (1 parameter).

Evaluations

Table 6.1: Basic dataset information

Dataset	# Fields	Dimensions	Science
QMCPACK	1	33120×69×69	electronic structure of atoms, molecules, and solids
RTM	1	449×449×235	Electronic
Miranda	7	256×384×384	hydrodynamics code for large turbulence simulations
CESM	79	1800×3600	Climate
Nyx	6	512×512×512	Cosmology
Hurricane Isabel	13	100×500×500	Weather
Hurricane Katrina	1	162×417642	Weather

All time evaluations are performed on Argonne Bebop Machine. There are 664 nodes in the cluster, each having 36 cores with 128GB DDR4 memory. This cluster uses Intel Xeon E5-2695v4 CPU.

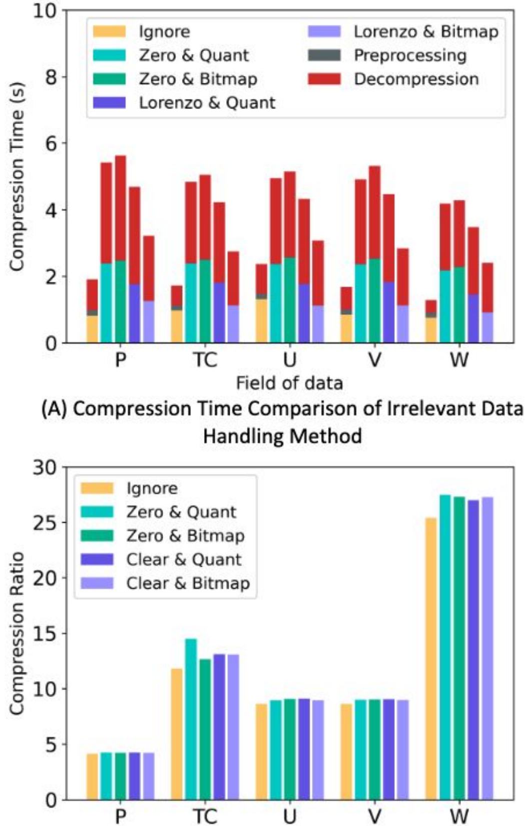


Isolating Irrelevant Data

Table 6.3: The 5 fields tested in the hurricane dataset

Field	Description	Value Range
P	Pressure (weight of atmosphere above a grid point)	-5471.8579/3225.4257
TC	Temperature (Celsius)	-83.00402/31.51576
U	X wind speed (positive means winds from west to east)	-79.47297/85.17703
V	Y wind speed (positive means winds from south to north)	-76.03391/82.95293
W	Z wind speed (positive means upward wind)	-9.06026/28.61434

This thesis uses either zero or Lorenzo predicted data to substitute the irrelevant data and uses either one quantization bin or a bitmap to record the indexes of the irrelevant data.



(A) Compression Time Comparison of Irrelevant Data Handling Method

(B) Compression Ratio Comparison of Irrelevant Data Handling Method

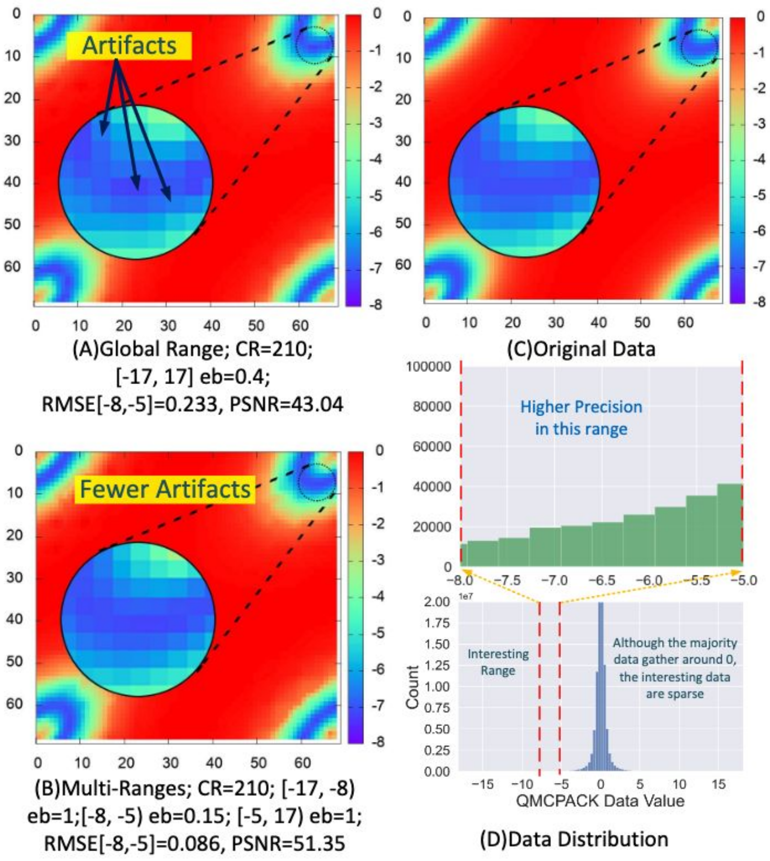
Multiple Value Interval

Table 6.4: QMCPACK RMSE & PSNR Comparison

Method	Range	eb	RMSE	PSNR
Global Range CR=210	[-17, -8]	0.4	0.232	43.067
	[-8, -5]		**0.233**	**43.041**
	[-5, 17]		0.051	56.159
Multi-Intervals CR=210	[-17, -8]	1.0	0.538	35.747
	[-8, -5]	0.15	**0.086**	**51.623**
	[-5, 17]	1.0	0.089	51.354

Table 6.5: Miranda density RMSE & PSNR Comparison

Method	Range	eb	RMSE	PSNR
Global Range CR=206	[0.5, 1.4]	0.07	0.012	44.804
	[1.4, 2]		**0.036**	**34.801**
	[2, 3.5]		0.015	42.379
Multi-Intervals CR=207	[0.5, 1.4]	0.1	0.013	43.5813
	[1.4, 2]	0.05	**0.027**	**37.193**
	[2, 3.5]	0.1	0.018	40.682



QMCPACK data

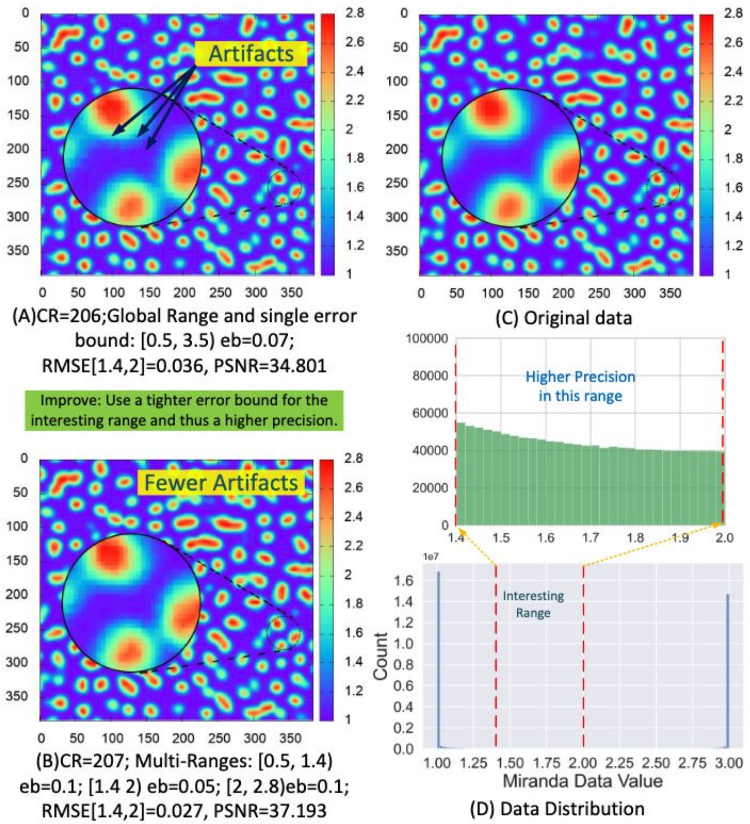
Multiple Value Interval

Table 6.4: QMCPACK RMSE & PSNR Comparison

Method	Range	eb	RMSE	PSNR
Global Range CR=210	[-17, -8]	0.4	0.232	43.067
	[-8, -5]		**0.233**	**43.041**
	[-5, 17]		0.051	56.159
Multi-Intervals CR=210	[-17, -8]	1.0	0.538	35.747
	[-8, -5]	0.15	**0.086**	**51.623**
	[-5, 17]	1.0	0.089	51.354

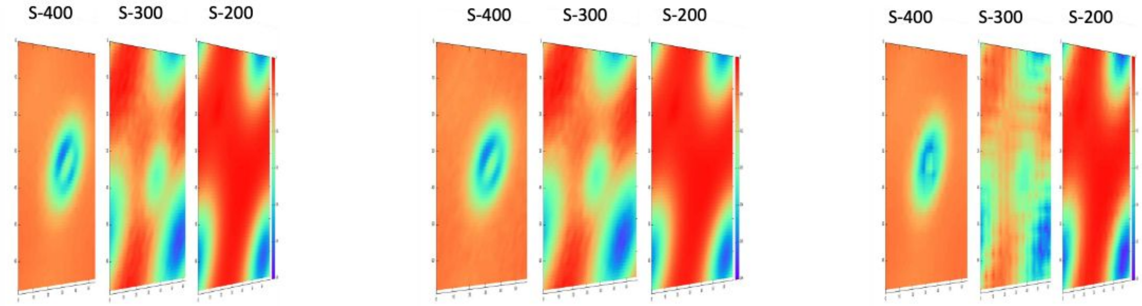
Table 6.5: Miranda density RMSE & PSNR Comparison

Method	Range	eb	RMSE	PSNR
Global Range CR=206	[0.5, 1.4]	0.07	0.012	44.804
	[1.4, 2]		**0.036**	**34.801**
	[2, 3.5]		0.015	42.379
Multi-Intervals CR=207	[0.5, 1.4]	0.1	0.013	43.5813
	[1.4, 2]	0.05	**0.027**	**37.193**
	[2, 3.5]	0.1	0.018	40.682



Miranda data

Multiple Regions

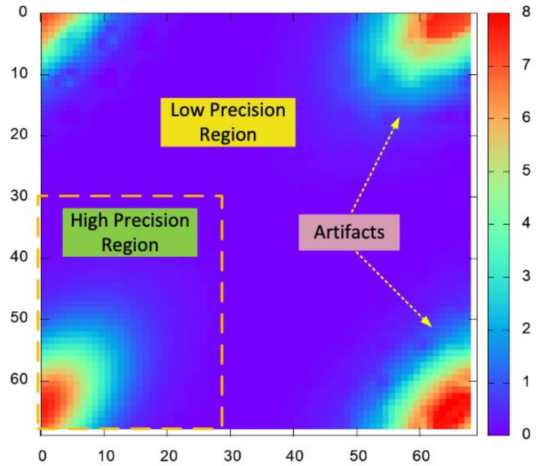


(A) Oursol (CR=54): multiple regions, create a small region-box for each significant region [190 0 0:20 69 69], [290 0 0:20 69 69], [390 0 0:20 69 69], and give each region a dedicated error bound.

(B) Original Data: the value ranges for the demonstrated regions are different, and each region requires a different precision to have a good visualization result.

(C) SZ3 All-0.01 (CR=27): the old method cannot take care of all regions. Even when giving a quite tight error bound 0.01, some regions will be hugely distorted.

Figure 6.8: QMCPACK visual quality comparison: Each slice has 69×69 pixels. We select slice 200, 300, and 400 to observe the visual distortion because each has a different range: slice 200 has range [-0.06, 0], slice 300 has range [-0.0016, 0], and slice 400 has range [-0.0025, 0.0005].



Compression Time Compression

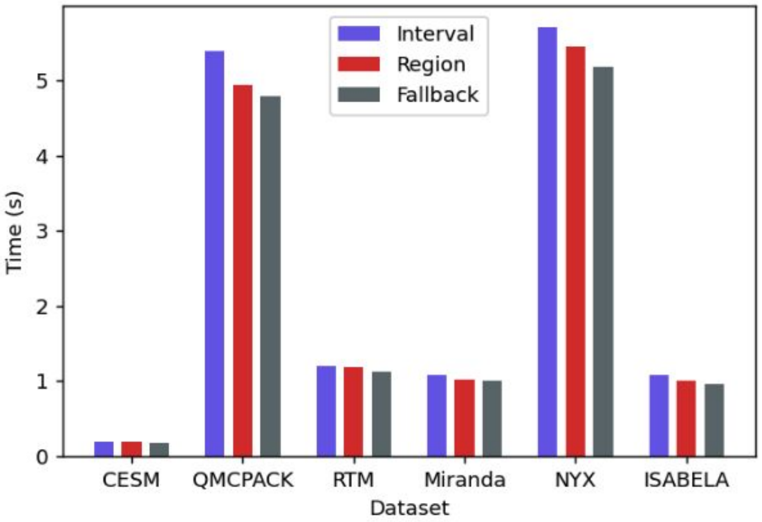
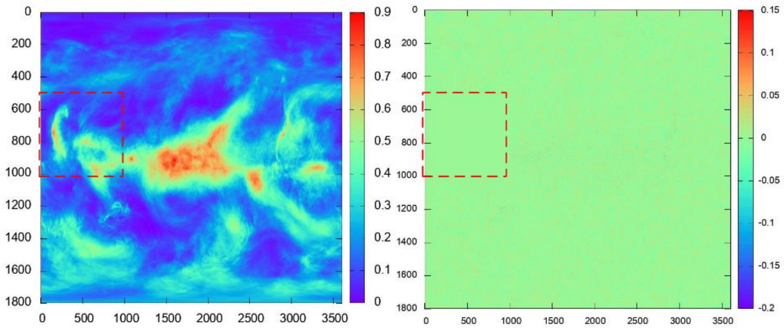


Table 6.7: Compression Time and Overhead of Interval/Region/Fallback Methods

Method	CESM	QMC	RTM	MIRAN	NYX	ISAB
Interval(s)	0.20	5.39	1.20	1.08	5.70	1.08
Region(s)	0.19	4.94	1.18	1.03	5.46	1.01
Fallback(s)	0.18	4.80	1.12	1.00	5.18	0.96
Interval%	8.9%	12.3%	7.1%	6.8%	10.0%	13.0%
Region%	3.3%	3.0%	5.4%	1.9%	5.4%	5.7%



(A) CLDHGH Data(1800×3600): The climate data map, the shape of which corresponds to the geolocations on earth.

(B) The difference image using region-based compression

Extract Irregular Regions from data

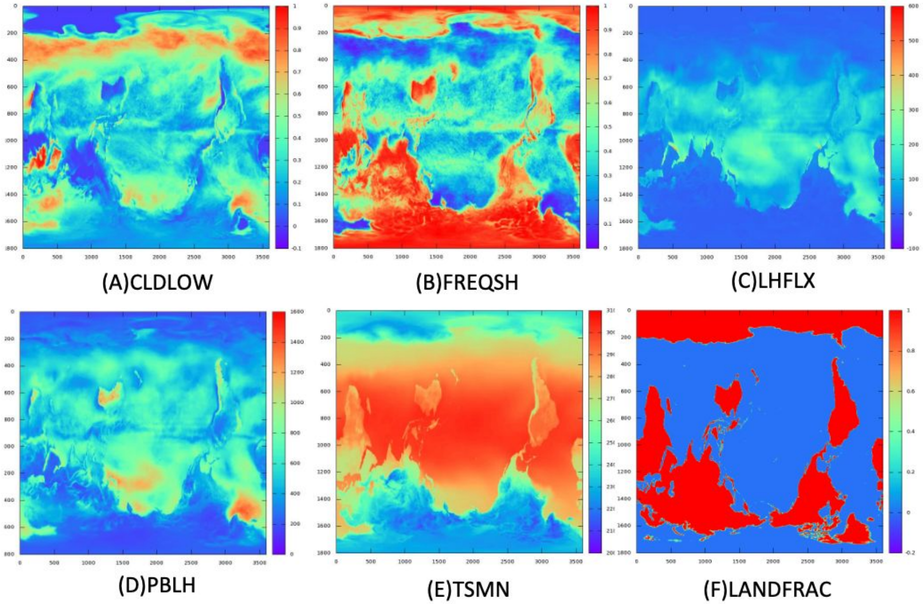


Figure 6.12: Six Fields in CESM: the visualization indicates that bitmap-separated precisions may be suitable to compress these fields.

In some datasets, geospatial information can be mapped from data indexes.

Land and ocean naturally may have different scientific significance and we can extract such information from the data and build a bitmap to set different error bounds to each part.

Users do not need to set the error bounds themselves in this scenario.

Extract Irregular Regions from data

Table 6.8: Compression Setting Definition

Setting	Description
A	SZ2.1 [Liang et al., 2018a]: Lorenzo & Linear Regression Predictor with one global error bound
B	Use SZ2.1's predictor, but adopt two error bounds set by a bitmap array
C	Interpolation based Compression with one uniform error bound [Zhao et al., 2021]
D	The proposed region based error bounded compressor with two error bounds set by a bitmap

Data Field	Setting	CR	CR'	PSNR	P_0	P_1
CLDLOW	A: eb=0.01	21	-	44.94	46.74	49.59
min=-0.1	B: eb=0.01, 0.1	30	29.0	29.71	46.74	29.73
max=1	C: eb=0.01	138	-	47.14	49.23	51.26
	D: eb=0.01, 0.1	224	176.6	32.31	49.22	32.34
FREQSH	A: eb=0.01	16	-	44.73	46.76	48.97
min=0	B: eb=0.01, 0.1	22	21.4	28.67	46.76	28.67
max=1	C: eb=0.01	88	-	46.79	48.83	50.99
	D: eb= 0.01, 0.1	126	109.5	32.10	48.83	32.13
LHFLX	A: eb=1	30	-	60.27	62.28	64.55
min=-100	B: eb=1, 10	48	45.4	49.36	62.28	49.55
max=600	C: eb=1	106	-	62.41	64.58	66.40
	D: eb= 1, 10	216	171.6	47.81	64.63	47.84
PBLH	A: eb=5	37	-	53.04	55.20	57.07
min=0	B: eb=5, 15	45	42.7	47.72	55.20	48.55
max=1600	C: eb=5	107	-	55.03	57.24	58.99
	D: eb= 5, 15	169	140.5	49.23	57.26	49.93
TSMN	A: eb=1	66	-	44.78	47.04	48.64
min=200	B: eb=1, 10	191	155.4	36.19	47.04	36.51
max=310	C: eb=1	292	-	47.14	49.41	50.99
	D: eb= 1, 10	812	411.5	31.64	49.24	31.66

Scalability Test

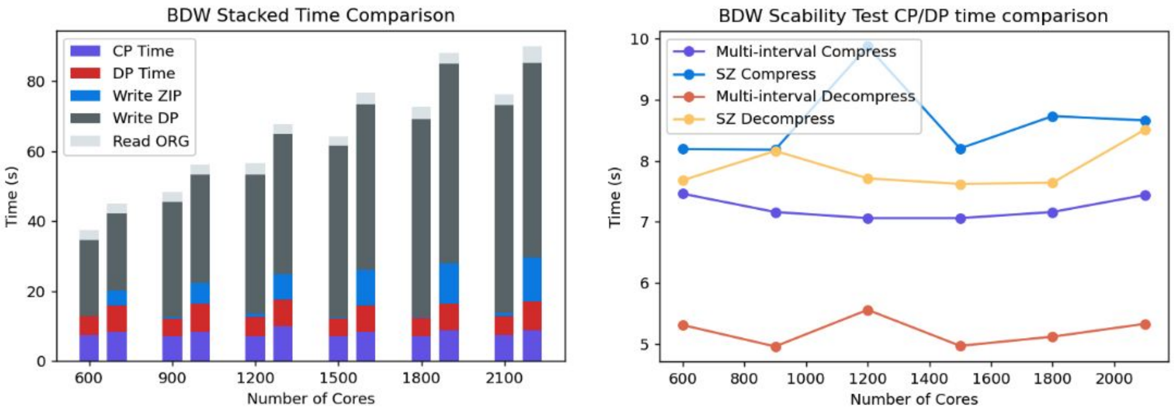


Figure 6.13: BDW partition: for each pair of bars, the left side is multi-interval solution's result, and the right side is SZ's result. CP/DP Time are compression/decompression time respectively. Write ZIP/Write DP are the I/O time to write the compressed/decompressed file respectively. Read ORG is the time to read the original file.

Summary

- ❖ Multi-interval/region error-bound-based compression can significantly improve the visual quality for users with the same or even higher compression ratios.
- ❖ Evaluation for the bitmap-based solution shows that the cost to satisfying a customized complex region requirement is acceptable and the proposed solution can possibly be generalized to suit all kinds of fine-grained error bound settings.
- ❖ Experiments on a supercomputer - Argonne Bebop with up to 3500+ cores show that the proposed multi-precision lossy compressors have a very good scalability.

Future Work

- Predict the (de)compression time
  - Using machine learning algorithms to predict the (de)compression time on a specific machine.
  - The main challenge is to extract effective features from data in a short amount of time.
- Predict the compression ratio
  - Using some mathematical deduction to estimate the prediction accuracy, quantization bin distribution, and huffman coding size to predict the compression ratio.
- If compression time and ratio can be relatively accurately predicted, it is possible to design efficient methods to transfer data.

Relevant Publications

[1] Y. Liu et al., "Optimizing Multi-Range based Error-Bounded Lossy Compression for Scientific Datasets," 2021 IEEE 28th International Conference on High Performance Computing, Data, and Analytics (HiPC), 2021, pp. 394-399, doi: 10.1109/HiPC53243.2021.00036.

[2] Y. Liu et al., "Understanding Effectiveness of Multi-Error-Bounded Lossy Compression for Preserving Ranges of Interest in Scientific Analysis," 2021 7th International Workshop on Data Analysis and Reduction for Big Scientific Data (DRBSD-7), 2021, pp. 40-46, doi: 10.1109/DRBSD754563.2021.00010.

Acknowledgements

Advisors

Ian Foster

Kyle Chard

Committee

Sheng Di

Contributors

Kai Zhao

Sian Jin

Dingwen Tao

Franck Cappello

Cheng Wang

Thank you!

Questions?

yuanjian@uchicago.edu