

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/390524924>

Assessment of lossless compression algorithms and their performance on near-infrared spectral images

Conference Paper · March 2025

CITATIONS

0

READS

16

6 authors, including:



Abtin Maghmoumi

RWTH Aachen University

10 PUBLICATIONS 206 CITATIONS

SEE PROFILE



Harsh Shekhar

RWTH Aachen University

1 PUBLICATION 0 CITATIONS

SEE PROFILE



Tabea Scherling

RWTH Aachen University

5 PUBLICATIONS 16 CITATIONS

SEE PROFILE



Nils Kroell

STADLER Anlagenbau GmbH

47 PUBLICATIONS 381 CITATIONS

SEE PROFILE

Assessment of lossless compression algorithms and their performance on near-infrared spectral images

Abtin Maghmoumi¹, Harsh Shekhar¹, Tabea Scherling¹, Nils Kroell²,
Alexander Feil¹, Kathrin Greiff¹

¹ Department of Anthropogenic Material Cycles,
RWTH Aachen University,
Wuellnerstr. 2, D-52062 Aachen, Germany
² STADLER Anlagenbau GmbH,
Max-Planck-Str. 2, D-88361 Altshausen, Germany

Abstract Storing and streaming hyperspectral images (HSI) requires significant resources. One suitable strategy of reducing this resource utilizations is the use of compression algorithms. In this work, we benchmarked *Bzip2*, *Gzip*, *LZ4* and *lzf* algorithms on a dataset of HSIs of lightweight plastic packaging and achieved a lossless compression as high as 96.7% compared to the original filesize.

Keywords Near-infrared spectroscopy, Hyperspectral image, Lightweight plastic packaging

1 Introduction

In recent years, the analysis of infrared (IR) spectral data has become significant in waste management. Among the various IR techniques, near-infrared (NIR) spectroscopy has gained particular importance as a reliable solution for sorting different types of plastic [1]. Near-infrared hyperspectral image (HSI) includes more information of the spectral domain as the number of sub-bands they cover goes over 100. This means each pixel shows a spectrum, which enables a detailed spectral analysis and use of sophisticated classification models for inline

material characterization [2]. However, this higher dimension introduces a challenge when the recorded HSIs are transferred or stored. For example, an image with dimension of $1000 \times 312 \times 220$ requires about 130 megabytes of storage space. With an acquisition rate of 300 frames per second (FPS), this will add up to 3.3 terabytes of data for a full day of recording, rendering the acquisition of raw spectral data in industrial processing scenarios infeasible. To tackle the storage challenge, Deore et al. developed a two-stage lossy compression algorithm that uses spectral decorrelation and discrete wavelet transformation (DWT) [3]. Beside lossy compression, lossless compression algorithms such as *Bzip2*, *7zip* and *Gzip* have shown promising results on satellite hyperspectral images [4]. One solution to improve compression performance is taking advantage of properties of near-infrared HSIs like replacing values of pixels in HSI which are not important for further analysis, i.e. spectra of background pixels in HSI of material being transported on a conveyor belt. Here, we benchmark four of the most common lossless compression algorithms *Bzip2*, *Gzip*, *LZ4* and *Lzf* on a dataset of HSIs of lightweight plastic packaging post-consumer waste in two scenarios (with and without background suppression) to find an efficient compression pipeline for such HSIs.

2 Material and Methods

2.1 Dataset generation

The hyperspectral imaging camera used to record the dataset was an EVK HELIOS NIR G2-320 EVK Kerschhaggl GmbH (Raaba, Austria) with a wavelength range of 990 nm – 1678 nm with a spectral resolution of 3.1 nm/band. In total 170 HSIs were recorded, the details about polymer type and number of sample are in Table 1, each HSI has a spatial resolution of 312 and spectral resolution of 220, and was recorded with 300 frames per second. The G2-320 measures and records lines, so to create images we used the GenICam interface of the camera to read the lines from the camera's buffer and append them together to create the hyperspectral images (Fig 1).

Table 1: Dataset structure and number of samples per class

Packaging polymere	# of samples	filesize (GB)
Polyethylene terephthalate (PET)	95	8.896
Polypropylene (PP)	50	4.056
High Density Polyethylen (HDPE)	25	2.445

2.2 Compression algorithms

Here the following lossless compression algorithms have been investigated, *Gzip* [5], *Bzip2* [6], *lzf* [7] and *LZ4* [8]. As these methods are lossless, the reconstructed spectrum remains identical to the original before compression. However, as it is crucial for the compressed file to be decompressed within a reasonable time, the decompression time was also measured. The *Python* implementation of the mentioned algorithms were used on a computer equipped with an Intel® Core i9 14900 and 128 GB of RAM, for storage a 2 TB NVME SSD.

2.3 Background suppression

To improve compression performance, through a preprocessing step, the objects are detected using *scikit-image* [9], and their bounding box is calculated and all the pixels, which lie outside the bounding box are considered as background and were given the value of zero over all channels, the pipeline is illustrated in Fig 2. The time required for object detection was also taken into account as part of compression time.

2.4 Performance assessment

To assess their performance, each near-infrared HSI was compressed once with and once without background suppression using all the four algorithms. To compare the performance of benchmarked compression algorithms, the compression ratio (CR) (1) and compression speed for each HSI were calculated, and then the mean and standard deviation (std) over the whole dataset were derived.

$$\text{Compression Ratio (CR)} = \frac{\text{Original File Size} - \text{Compressed File Size}}{\text{Original File Size}}$$

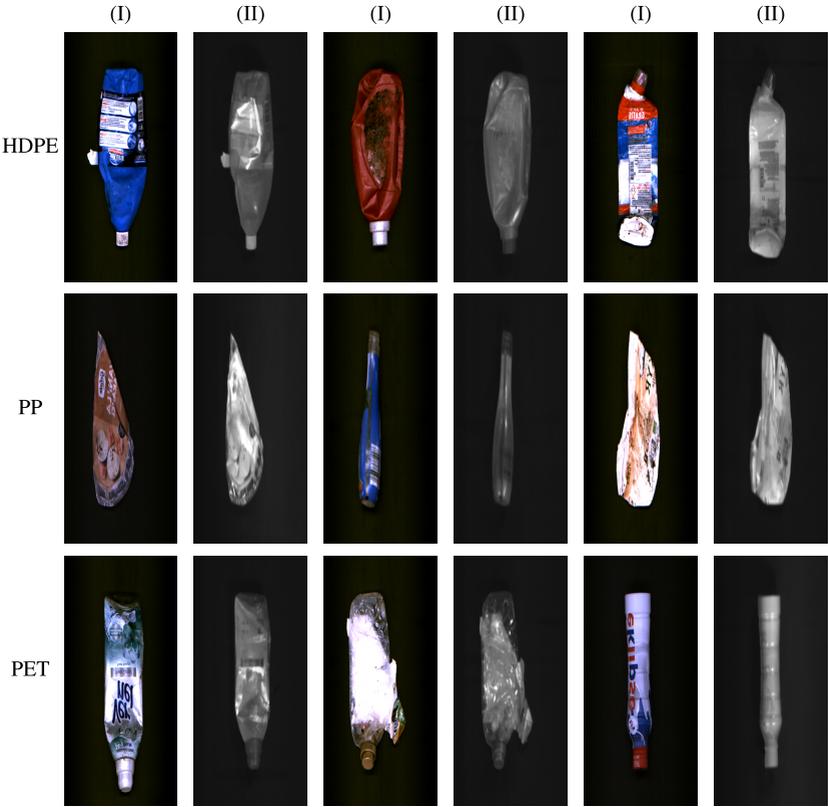


Figure 1: Sample images from dataset, (I) columns illustrate the RGB image and (II) columns visualize the HSI as a grayscale image (for each pixel an average was calculated)

(1)

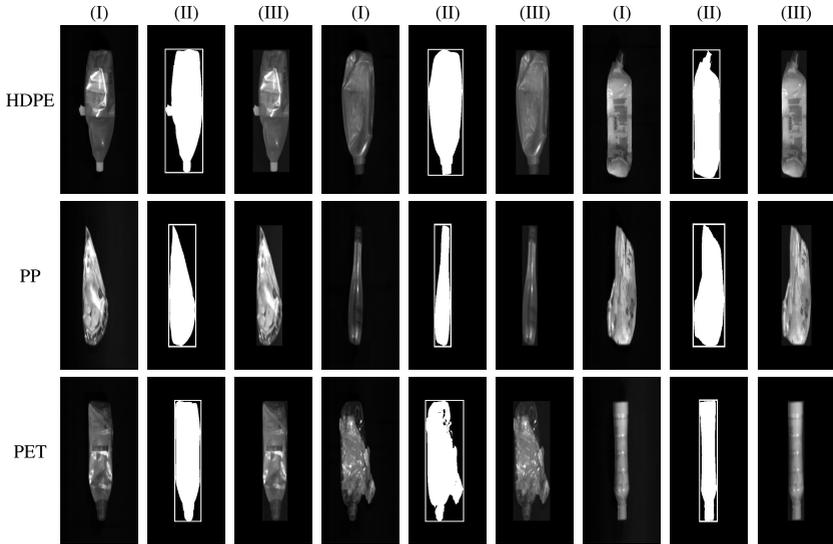


Figure 2: The pipeline for suppressing background pixels. (I) columns visualize the HSI as a grayscale image (for each pixel an average was calculated), (II) columns are binary images of the grayscale image with detected objects and their bounding boxes and (III) columns depict the grayscale images of the HSIs with suppressed background.

3 Results and Discussion

3.1 Compression performance without background suppression

The achieved CRs for no background suppression scenario varies between 58.9 % and 90.1 % with *Bzip2* performing the best, while *LZ4* only achieved an average CR of 58.9 %. As seen in Table 2, *Bzip2* achieves the highest CR with *Gzip*, *lzf* and *LZ4* following. For both *Gzip* and *Bzip2* a compression level parameter can be defined: for *Gzip* it defines the search window and the intensity of Huffman tree optimization and for *Bzip2* it changes the block size used for compression. Varying the compression level parameter of both *Gzip* and *Bzip2* algorithms indicates that for *Bzip2* on average only 2.1 % higher CR are achieved, for *Gzip* on the other hand a 6.2% increase in CR is observed.

Table 2: Compression results without background suppression (highest mean values are bolded)

		CR		Speed [MB/s]	
		mean	std	mean	std
Algorithm Compression level					
Bzip2	1	88.0%	2.0%	15.3	1.8
	2	88.8%	1.8%	13.3	1.9
	3	89.2%	1.7%	12.7	2.1
	4	89.5%	1.6%	12.2	2.1
	5	89.7%	1.6%	11.5	2.1
	6	89.8%	1.5%	11.0	2.0
	7	90.0%	1.5%	10.5	2.0
	8	90.0%	1.4%	10.2	2.0
	9	90.1%	1.4%	9.7	1.9
Gzip	1	76.4%	2.6%	85.3	11.3
	2	77.7%	2.5%	75.0	9.6
	3	79.2%	2.6%	55.0	6.3
	4	80.9%	2.6%	56.2	7.1
	5	81.7%	2.6%	41.5	5.0
	6	82.4%	2.7%	27.6	2.7
	7	82.6%	2.7%	19.3	1.9
	8	82.6%	2.7%	10.0	1.4
	9	82.6%	2.7%	8.9	1.3
lzf	-	64.5%	3.3%	135.6	29.2
LZ4	-	58.9%	2.8%	157.0	38.9

Beside CR, the compression speed should also be evaluated. Though *Bzip2* achieves higher CR, it is much slower in comparison to other algorithms, which raises the concern for its real-time implementation.

3.2 Compression performance with background suppression

Suppressing the background in HSIs improves CR, on average the CR attained by algorithms were improved by 11.1 %, the maximum improvement in CR was for *LZ4* with a 30 % increase of CR. As for compression speed, both *Gzip* and *Bzip2* achieve more throughput, but *lzf* and *LZ4* slow down. Table 3 illustrates the results for compression with background suppression. The first observation is an evident increase in CR for all algorithms, with *Bzip2* achieving CR as high as 96.70 %, an explanation would be the decreased variance in HSI with background suppression and existence of repetitive zero values in the HSI, these enable the compression algorithms to achieve higher compression ra-

Table 3: Compression results with background suppression (highest mean values are bolded)

		CR		Speed [MB/s]	
		mean	std	mean	std
Algorithm Compression level					
Bzip2	1	95.6%	2.4%	32.6	11.0
	2	96.0%	2.2%	29.7	10.6
	3	96.2%	2.0%	27.7	10.3
	4	96.4%	1.9%	26.4	10.1
	5	96.5%	1.9%	25.4	9.8
	6	96.6%	1.8%	24.5	9.5
	7	96.6%	1.8%	22.5	9.3
	8	96.7%	1.8%	22.0	9.1
	9	96.7%	1.7%	21.0	9.2
Gzip	1	91.7%	3.9%	100.3	29.1
	2	92.0%	3.8%	100.4	26.9
	3	92.2%	3.7%	99.7	29.2
	4	92.7%	3.6%	84.6	18.4
	5	92.8%	3.6%	78.8	19.6
	6	92.8%	3.6%	72.3	19.4
	7	92.8%	3.6%	72.9	20.6
	8	92.8%	3.6%	66.6	21.2
	9	92.9%	3.6%	60.7	20.7
lzf	-	88.2%	5.5%	96.5	35.7
LZ4	-	89.1%	5.3%	103.3	32.0

tios. When it comes to compression speeds, for *gzip* and *Bzip2* a subtle boost in speed can be seen, but for *lzf* and *LZ4* a decrease is observed. Also, changing the compression level parameter for *Bzip2* and *gzip* leads to minimal improvement of CR, so it is not necessary to sacrifice compression speed to achieve lower sized HSIs. Interestingly, *lzf* and *LZ4* achieve a CR close to 90 % while maintaining a fast compression speed.

3.3 Decompression performance

When compressing a file, it is also important to evaluate the decompression performance, specially how fast the original HSI can be reconstructed. In Table 4 the mean and standard deviation of decompression time and decompression speed are shown for the dataset in case of no background suppression. *LZ4*, *lzf* and *gzip* achieve much higher decompression speeds compared to *Bzip2*, it is also worth not-

Table 4: Decompression results for compressed HSIs without background suppression.

		Speed [MB/s]	
		mean	std
Algorithm	Compression level		
Bzip2	1	87.90	16.97
	2	98.81	6.44
	3	97.67	6.15
	4	92.82	10.56
	5	69.31	12.17
	6	76.56	17.86
	7	88.78	5.68
	8	67.42	13.01
	9	60.08	9.09
Gzip	1	279.63	13.23
	2	269.41	16.94
	3	281.21	19.54
	4	284.03	18.47
	5	253.21	26.06
	6	303.50	25.14
	7	297.71	24.76
	8	302.87	21.76
	9	290.21	22.52
lzf	-	320.00	23.33
LZ4	-	292.23	23.33

ing that different compression levels does not affect the decompression speed of gzip algorithm. The results for decompression of HSIs with suppressed background are presented in Table 5, showing a noticeable boost in the decompression speed. *Bzip2* gained the most boost, with an average 285.7 % increase. For *Gzip*, *lzf* and *LZ4* an average boost of about 30 % can be concluded.

4 Conclusion

The dense 3D array structure of near-infrared hyperspectral images makes their storage particularly challenging. To tackle this, we have benchmarked four of the most common lossless compression algorithms on a dataset of HSIs of lightweight plastic packaging and shown that if the background pixels are suppressed before compression the achieved compression ratios of all four algorithms increases. The best algorithm was *Bzip2* with a CR as high a 96.7 % while maintaining

Table 5: Decompression results for compressed HSIs with background suppression.

		Speed [MB/s]		
		mean	std	
Bzip2	Algorithm	Compression level		
		1	257.41 60.06	
		2	254.68 56.12	
		3	247.25 53.40	
		4	247.33 55.24	
		5	239.33 52.74	
		6	237.85 56.33	
		7	229.05 53.70	
		8	222.04 52.44	
		9	221.86 54.25	
		1	440.73 51.74	
		2	429.08 53.49	
		3	423.07 48.67	
		4	407.50 45.86	
Gzip		5	366.34 67.48	
		6	338.38 45.36	
		7	411.94 47.61	
		8	403.80 50.92	
		9	404.44 42.64	
		lzf	-	492.63 59.37
		LZ4	-	395.68 21.77

an acceptable compression speed, *Gzip* also achieved CRs higher than 90 % with much faster compression speed, showing its feasibility for a real-time implementation. However, a real-time implementation depends highly on the equipment, in our case for recording with 300 FPS we would need a compression speed of 39.28 MB/s, which is still lower than the slowest settings of *Gzip*. In conclusion, two pipelines can be introduced depending on use case, for non-real-time applications a pipeline of first suppressing background and then using *Bzip2* algorithm for compression is suggested but if real-time performance is deemed then *Gzip* should be utilized for compression at cost of higher storage requirement. For future work our results could be used for comparison with lossy algorithms and the trade-off between compression performance and classification results.

Acknowledgements

This work was funded the German Federal Ministry of Education and Research (BMBF) within the program "Resource-efficient circular economy - plastic recycling technologies (KuRT)" under the project ReVise-UP (grant no. 033R390A). We thank Peter Bardenheuer for his assistance in recording the dataset used in this work. His contributions were essential in ensuring the data was collected effectively and accurately.

References

1. E. R. K. Neo, Z. Yeo, J. S. C. Low, V. Goodship, and K. Debattista, "A review on chemometric techniques with infrared, raman and laser-induced breakdown spectroscopy for sorting plastic waste in the recycling industry," *Resources, Conservation and Recycling*, vol. 180, p. 106217, 2022.
2. G. ElMasry and D.-W. Sun, "Chapter 1 - principles of hyperspectral imaging technology," in *Hyperspectral imaging for food quality analysis and control*, D.-W. Sun, Ed. London and Burlington MA: Academic Press an imprint of Elsevier, 2010, pp. 3–43. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780123747532100012>
3. G. dDeore, S. Rajaraman, R. Awate, and S. Bakare, "Spectral-spatial hyperspectral image compression based on measures of central tendency," in *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2015, pp. 2226–2232.
4. M. Grossberg, I. Gladkova, S. Gottipati, M. Rabinowitz, P. Alabi, T. George, and A. Pacheco, "A comparative study of lossless compression algorithms on multi-spectral imager data," in *DCC 2009*, J. A. Storer and M. W. Marcellin, Eds. Los Alamitos Calif.: IEEE Computer Society, 2009, p. 447.
5. J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337–343, 1977.
6. J. Seward, "Bzip2 algorithm." [Online]. Available: <http://www.bzip.org>.
7. M. Lehmann, "Liblzf," 2008. [Online]. Available: <http://oldhome.schmorp.de/marc/liblzf.html>
8. "Lz4 - extremely fast compression," 2024. [Online]. Available: <https://lz4.org/>

9. S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, "scikit-image: image processing in python," *PeerJ*, vol. 2, p. e453, 2014.

