Analysis of Lossless Compressors Applied to Integer and Floating-Point Astronomical Data

Òscar Maireles-González, Joan Bartrina-Rapesta,
Miguel Hernández-Cabronero and Joan Serra-Sagristà

Department of Information and Communications Engineering
Universitat Autònoma de Barcelona
Cerdanyola del Vallès, 08193 Bellaterra, Spain
oscar.maireles@uab.cat

Abstract

In this work, lossless compression algorithms are evaluated on a variety of real, current astronomical images. The test dataset comprises raw (integer) and processed (floating-point) images of discrete and extensive astronomical objects, captured by spatial or terrestrial telescopes. Compression techniques herein analyzed are chosen to be representative of the most recent algorithms devised for astronomical data, as well as the most commonly employed compressors employed in real observatories. Experimental results suggest that coding techniques such as RICE and HCOMPRESS, typically employed in world-class observatories such as Roque de los Muchachos, do not produce the best possible lossless compression results. Instead, JPEG-LS, LZMA and NDZIP yield the best compression ratio results for 16-bit data (2.72), floating-point data (2.38) and radio data (1.81), respectively. Therefore, the efficiency with which data are stored and transmitted by these observatories could be significantly improved by selectively employing the aforementioned algorithms.

Introduction

The field of astronomy experienced a radical transformation due to the appearance of Charge-Coupled Devices (CCDs). As telescopes and detectors are improved, larger data quantities of higher quality are retrieved, which results in larger data volumes and acquisition rates [1, 2]. This is reflected in the next generation of satellites and telescopes, led by The James Webb Space Telescope (JWST) [3], launched late 2021; and The European Extremely Large Telescope (E-ELT) [4]. The E-ELT is the biggest land telescope ever built, and is expected to be operative by 2024 at The European Southern Observatory. While JWST will generate 100 TB every year, E-ELT will produce 1 TB every night [5]. In this context, efficient data compression is paramount to control the storage and transmission costs associated to these enormous volumes of data.

In astronomy, compressed data are not always used. Sometimes data obtained after an observation night in an observatory is shared to the astronomer or scientist using an FTP link.[1] When compression is required, the key aspects of astronomical images compression are the modeling and compression of noise, the random space objects distribution [6], and the possible data values and their differences (floating-point values lower than 1 are the most difficult to compress). This is due to the

[1] https://www.ing.iac.es/astronomy/computing/recording.html

long exposure times needed to acquire data and the random noise introduced by the atmosphere, space background and heat noise. The resulting images are typically composed of a dark space background with some discrete astronomical objects, e.g., stars and galaxies, which are the ones of scientific interest. To study high-redshift and faint objects in a precise way, it is also necessary to preserve weak object signals as faithfully as possible.

Astronomical images consist of either floating-point or integer samples. Original images taken from the instrument CCD always retrieve integer samples, each pixel value being roughly proportional to the number of photons received. This primary signal is then calibrated based on known statistics to suppress noise, producing a floating-point image. The use of integer or floating-point data for storage and transmission depends on the observatory and survey.

The main contribution of this work is to disclose what existing compression techniques perform best on different types of astronomical images. Techniques hereafter analyzed include those nowadays employed in observatories, as well as others representative of the state of the art in integer and floating-point lossless data compression. To the best of our knowledge, this is the first time a study with this scope is conducted.

The rest of the article is organized as follows. Next section describes the employed dataset, followed by a state-of-the-art of the different coding techniques used for astronomical data. Then, compression results for all described techniques and different datasets are investigated. Conclusions and future work are presented last.

Materials and methods

Astronomical dataset

The astronomical surveys chosen for this study represent a wide variety of astronomical images, including optical, radio, infra-red and multi-spectral. They are acquired from both terrestrial and spatial telescopes. Table 1 introduces each dataset, reporting width, height and spectral dimension. Sets with a spectral dimension higher than 1 correspond to extensive object images. CALIFA, CHEOPS and SDSS-MaNGA are multi-spectral images, so they have the same two spatial dimensions as the rest of the dataset plus a third spectral dimension. The Type column indicates if the dataset contains integer and/or floating point data (IEEE 754 standard), complemented by the data format: 16, 32 and 64-bits[2]. The last column indicates the acquisition year of the images.

The dataset consists of 103 images, with a combined uncompressed size of 16 GB.

Coding techniques for integer data

Currently, the most commonly used compression techniques applied on observatories are designed for integer lossless compression. These are HCOMPRESS [9] and RICE [10]. HCOMPRESS uses a wavelet-transform followed by a quadtree coding,

[2]All the data used for this research is available at the following public repository: https://gicilab.uab.cat/omaireles/astronomical-compression in FITS (original) format or in raw (uncoded) format.

Survey/Telescope	Dimensions	Type	Format	Year
DESI	2036x2048x1	Integer	16-bit	2015
INT	2154x4200x1	Integer	16-bit	2004
JKT	2148x2148x1	Integer	16-bit	2007
WHT	1024x1024x1	Integer	16 & 32-bit	2005
CALIFA	78x73x1877	Floating-point	32-bit	2013
GAIA	4000x3000x1	Floating-point	64-bit	2015
HerMES	315x2354x1	Floating-point	32 & 64-bit	2011
LOFAR	7564x7564x1	Floating-point	32 & 64-bit	2012
SDSS-BOSS [7]	2048x1489x1	Floating-point	32-bit	2005-2006
SDSS-MaNGA [8]	54x54x4563	Floating-point	16, 32 & 64-bit	2015-2016
VIMOS	2108x2108x1	Floating-point	32 & 64-bit	2018
WISE	4095x4095x1	Floating-point	32-bit	2009
CHEOPS	50x50x3070	Integer & floating point	16, 32 & 64-bit	2019
TJO	4096x4108x1	Integer & Floating-point	16 & 32-bit	2016

Table 1: List of all the datasets and their main features.

while RICE technique uses a set of variable-length codes. These two techniques are supported by fpack and funpack software [11]. These compressing and decompressing programs are designed specifically to compress Flexible Image Transport System (FITS) format, which is the one universally used in astronomical data. The technique of [11] divides the image in rectangular tiles, to compress and store them in a FITS binary table. Each HDU (Header/Data Units) of a multi-extension FITS file is compressed separately, so it is not necessary to uncompress the entire file to read a single image. This adaptation to the FITS format makes reference [11] the most popular approach to work with astronomical data.

On the other hand, there exist techniques devised for image compression that provide 16-bit integer support, such as JPEG-LS [12]. JPEG-LS uses a modeling step, which assigns probabilities to data, and a Golomb (or an arithmetic) coder. Another JPEG family standard, i.e., JPEG 2000 [13], is also considered. This technique performs compression by applying a discrete wavelet transform, quantization and entropy coding.

Coding techniques for floating-point data

To the best of our knowledge, no specific floating-point compression technique has been devised for astronomical images. The most used and the newest technique for floating-point data type is, respectively, ZFP [14] and NDZIP [15]. ZFP uses vector quantization and orthogonal block transform. NDZIP follows 3 main steps: data

division into fixed blocks, Lorenzo transform [16] and residual value encoding.

Coding techniques for integer and floating-point data

Finally, there are several techniques supporting both integer and floating point data simultaneously. One of the first coding techniques used for floating-point astronomical data was LZ77 [17], a dictionary compression technique. LZF technique [18] is an implementation of the most basic LZ77. The LZ77 technique can also be applied by combining it with another one, such as i) LZMA [19], which after using LZ77 technique, encodes the output with a range encoder; ii) ZSTANDARD [20], which, after applying the LZ77 dictionary, uses Finite State Entropy (FSE) and Huffman coding; iii) GZIP, which has two stages, the first using LZ77 technique and the second one using Huffman coding; and iv) LZ4 [21], performing LZ77 compression by representing data as series of sequences.

One of the most used computer compression techniques is BZIP2 [22]. This technique uses Run-Length Encoding (RLE) and then the Burrows-Wheeler transform and Huffman coding.

Besides these, there are other techniques that have different compression approaches depending on the input data, such as FAPEC [23] or SPDP [24]. FAPEC can use different pre-processing stages depending on the data. SPDP uses its own framework, called CRUSHER, to select the better compression techniques depending on the input data. In CRUSHER, Run Length Encoding (RLE), LZln (LZ77 variant) or zero encoder [25], can be selected to perform compression.

The last technique is SZIP, an implementation of the extended-RICE lossless compression technique, to adapt it to lossless floating-point data compression.

Experimental Results and Analysis

Since not all the techniques work for each image type, compression tests have been organized according to the type-format and the most significant astronomical image types. Thus, experiments and analysis are grouped by Integer (16 bits), Float (32 and 64 bits), Extensive objects, Discrete objects, and Radio data. Compression performance is provided for lossless mode and is measured in terms of compression ratio. The next sections provide a discussion of the coding performance results for each data type. In addition, Table 2 summarizes the results.

Integer (16 bits)

This first dataset is used to assess the best compressors to use on the acquired data (without post-processing). The techniques evaluated are: RICE, HCOMPRESS, GZIP, LZF, SZIP, LZ4, SPDP, BZIP2, FAPEC, LZMA, ZSTANDARD, JPEG-LS and JPEG 2000 (with 0 and 5 wavelet levels).

Fig. 1(a) shows that the most efficient coding techniques are JPEG-LS, BZIP2 and JPEG 2000, providing compression ratios of 2.72, 2.69 and 2.69, respectively. Considering that JPEG-LS and BZIP2 employ Golomb and Huffman as entropy encoder, respectively; results suggest that the predictor of JPEG-LS and the Burrows-Wheeler

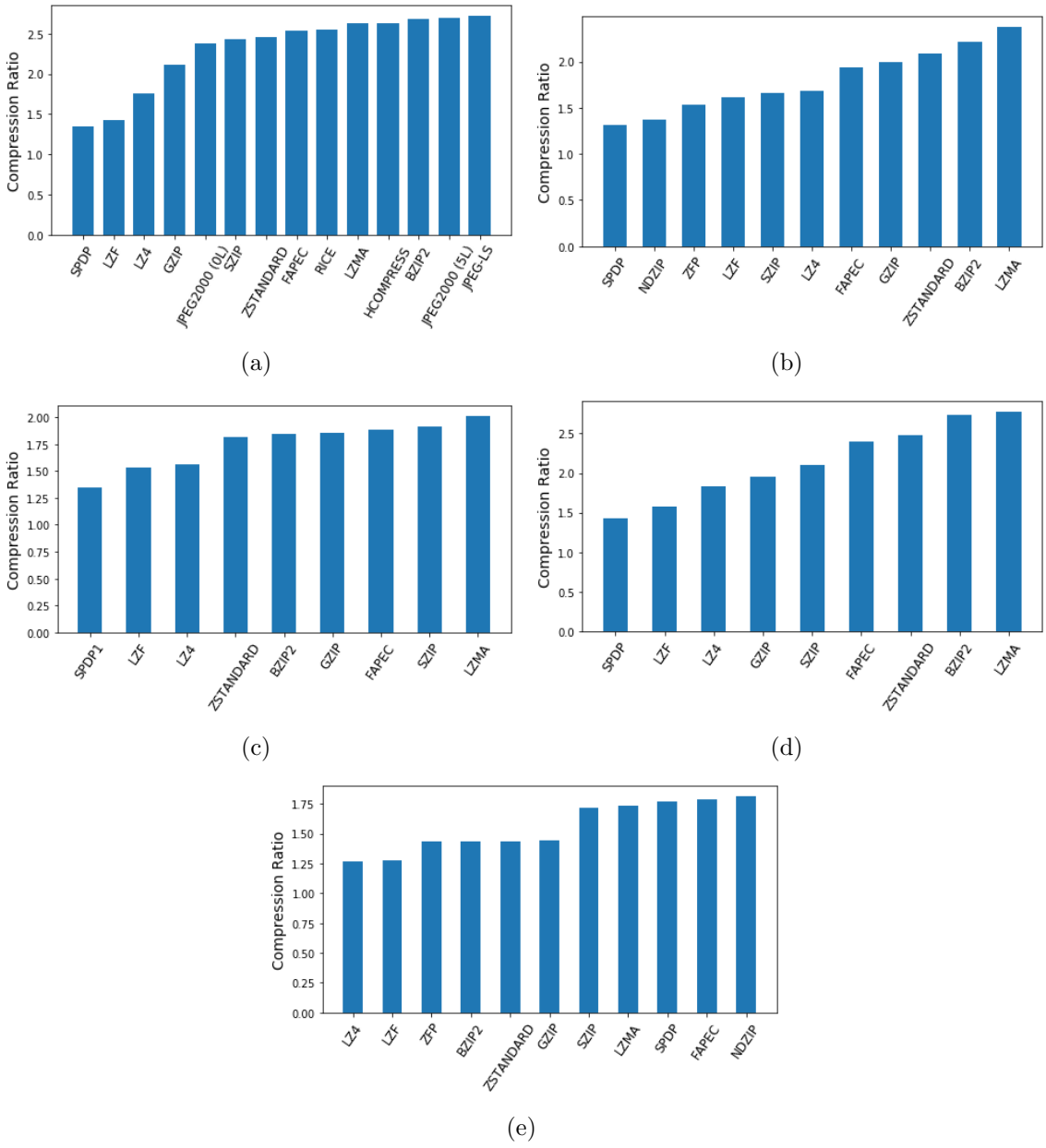

Figure 1: Mean values of the compression ratio of (a) integer 16-bit data (b) floating-point data (32 & 64-bit) (c) extensive objects data (CALIFA, CHEOPS and MaNGA) (d) discrete objects data (DESI, GAIA, INT, JKT, HerMES, LOFAR, BOSS, MaNGA, WHT, VIMOS, WISE, TJO) (e) radio data (LOFAR).

transform of BZIP2 provide similar results decorrelating this data. For JPEG 2000 there does not exist significant gain when 0 and 5 wavelet levels are applied. This suggests that spatial information is not properly exploited.

On the opposite side, the worst performances are those of SPDP, LZF and LZ4, the representative techniques of the LZ77 dictionary that employ very simple dictionary strategies.

Technique	Integer	Float	Extense	Discrete	Radio
HCOMPRESS	2.63	-	-	-	-
JEPG2000 level 5	2.69	-	-	-	-
JPEG-LS	**2.72**	-	-	-	-
RICE	2.55	-	-	-	-
NDZIP	-	1.36	-	-	**1.81**
ZFP	-	1.73	-	-	1.43
BZIP2	2.69	2.22	1.85	2.74	1.43
FAPEC	2.53	1.94	1.89	2.39	1.79
GZIP	2.11	2.00	1.85	1.95	1.44
LZF	1.43	1.61	1.54	1.58	1.27
LZ4	1.75	1.68	1.56	1.83	1.27
LZMA	2.63	**2.38**	**2.01**	**2.77**	1.73
SPDP	1.35	1.31	1.35	1.43	1.77
SZIP	2.43	1.66	1.91	2.10	1.72
ZSTANDARD	2.46	2.09	1.82	2.48	1.44

Table 2: Compression ratio mean values for each technique tested in that particular data type. "-" indicates the technique is not able to perform lossless compression for that dataset type (integer or floating-point). Best performing technique is highlighted in green background and bold type.

Float (32 and 64 bits)

This dataset is made of floating-point images, both 32 and 64-bit data. The investigated techniques are: LZF, SZIP, GZIP, LZ4, ZFP, SPDP, BZIP2, NDZIP FAPEC, LZMA and ZSTANDARD.

Fig. 1(b) shows that the most efficient coding techniques are LZMA, BZIP2 and ZSTANDARD. In this case, there is a higher compression ratio difference than for the integer (16 bits) case between the best technique and the following ones. Note that LZMA provides the best performance with a compression ratio of 2.38 followed by BZIP2 and ZSTANDARD with 2.22 and 2.09, respectively. LZMA and ZSTANDARD are based on LZ77, but LZMA uses a huge dictionary (up to 4GB) followed by a range encoder, whereas ZSTANDARD employs a dictionary with a large search window. The large dictionary of LZMA can manage more efficiently all the codewords provided by images with dynamic ranges of 32 and 64 bits than ZSTANDARD or BZIP2.

For this data type, the less efficient techniques are NDZIP and, again, SPDP and LZF. This fact hints that LZ77 alone (in its most basic form) is not an efficient compression technique for this kind of data. This also shows the efficiency of the range encoder, given that just adding a range encoder after LZ77 –with a large dictionary, like LZMA – makes the whole approach to become the best one.

Extensive objects

This dataset assesses the most efficient techniques on extensive objects, which are stored in multi-component images. These images have two short spatial dimensions and a wide third spectral dimension. The astronomical object covers almost all of the image surface, so the standard structure in astronomical images with a dominant background and some discrete small object is not met. Given that integer and floating-point data are used, the compressors able to compress such different data types are: BZIP2, FAPEC, LZF, SZIP, GZIP, LZ4, LZMA, SPDP and ZSTANDARD.

Fig. 1(c) shows that the best coding technique again is LZMA followed by SZIP and FAPEC. Now that the image represents an extensive object instead of discrete ones, LZMA is still way better than the other techniques. LZMA provides a compression ratio of 2.01 and SZIP 1.91. In this case, LZ77 becomes more useful, given that the shape of the astronomical data objects has a major presence on the image and has an approximate radial distribution. In this case, there are no dominant background values, but a wide variety of similar probability values. The complexity of the LZMA dictionary becomes more useful than in the previous section to efficiently encode these values. In the previous case, the third best technique was ZSTANDARD, using LZ77 plus FSE plus Huffman coding; now GZIP has a higher compression ratio, which follows the same structure without FSE, noticing that there is a gradient between the center and the image edges that does not benefit FSE technique (based on value probability).

Discrete objects

This dataset assesses the performance of the same techniques as in the extensive objects test, but applied to discrete astronomical objects (stars and far galaxies).

Fig. 1(d) shows that the most efficient coding techniques are LZMA, BZIP2 and ZSTANDARD. The same result as in the floating-point section. The better compression ratio is mostly caused by the addition of integer data to the dataset.

Radio data

This dataset considers the radio data from LOFAR survey (floating-point), so the same techniques as in floating-point section are assessed. Radio data values are the lowest in the dataset and some of them are even negative, due to the treatment process of the data, which requires extra steps than optical or IR acquisition.

Fig. 1(e) reports the results. NDZIP is now the most efficient compressor, followed by FAPEC, SPDP and LZMA. The competitive behaviour of NDZIP is attributed to performing residual value encoding, which benefits from very low values; FAPEC pre-processing stage allows it to adapt properly to really low values and SPDP uses a zero encoder useful on very low values. After these three coding techniques LZMA is the one that provides competitive coding performance, possibly due to the large dictionary and the poor spatial redundancy exploitation by the pre-processing strategies of FAPEC and NDZIP.

Conclusions and Future Work

Astronomical data compression is a compelling topic attracting the interest of practitioners in both data compression and astronomical research communities. This work establishes a starting point to disclose the most interesting techniques to be employed for astronomical data in general terms.

Our contribution investigates, for the first time, the performance of several coding techniques in more than one type of astronomical data. In particular, astronomical images in this study contain a dominant sky background representation and some dispersed objects with a number of pixel values much higher than the background. These images are not one-shot pictures, but long exposure data acquisition files.

Experimental results reveal that LZMA has almost always the best coding performance, except for 16-bit integer data, where the best compressor is JPEG-LS, and for radio data, where NDZIP is the most efficient one. Currently, RICE and HCOMPRESS compression techniques supported by fpack are the most widespread techniques, because they are designed to work with the FITS astronomical data format. However, our research indicates that these techniques are not the most competitive for lossless data compression. The average compression ratio obtained with HCOMPRESS for 16-bit integer data is 2.63, whereas the compression ratio obtained with JPEG-LS is 2.72. LZMA achives 2.38, 2.01 and 2.77 on floating-point data, extensive objects and discrete objectes, respectively. On radio data, NDZID compression technique achieves a compression ratio of 1.81. In light of our results, LZMA, JPEG-LS or NDZIP (for radio data) could be adapted to work directly on FITS data format, thus increasing the coding performance.

Acknowledgements

This research was partially funded by Beatriu de Pinós, reference 2018-BP-00008, funded by the Secretary of Universities and Research (Government of Catalonia) and by the Horizon 2020 programme of research and innovation of the European Union (EU) under the Marie Skłodowska-Curie grant agreement #801370, by the Secretary of Universities and Research (Government of Catalonia) under grant 2017SGR-463, and by the Spanish Ministry of Economy and Competitiveness and the European Regional Development Fund under grant RTI2018-095287-B-I00 (MINECO/FEDER, UE).

This study uses data provided by the Calar Alto Legacy Integral Field Area (CALIFA) survey (http://califa.caha.es/). Based on observations collected at the Centro Astronómico Hispano Alemán (CAHA) at Calar Alto, operated jointly by the Max-Planck-Institut für Astronomie and the Instituto de Astrofísica de Andalucía (CSIC).

Part of this work is based on archival data, software or online services provided by the Space Science Data Center - ASI.

The Legacy Surveys imaging of the DESI footprint is supported by the Director, Office of Science, Office of High Energy Physics of the U.S. Department of Energy under Contract No. DE-AC02-05CH1123, by the National Energy Research Scientific

Computing Center, a DOE Office of Science User Facility under the same contract; and by the U.S. National Science Foundation, Division of Astronomical Sciences under Contract No. AST-0950945 to NOAO.

Based on observations made with the Isaac Newton Telescope, William Herschel Telescope and Jacobus Kapteyn Telescope operated on the island of La Palma by the Isaac Newton Group of Telescopes in the Spanish Observatorio del Roque de los Muchachos of the Instituto de Astrofísica de Canarias.

This research has made use of data from HerMES project (http://hermes.sussex.ac.uk/). HerMES is a Herschel Key Programme utilising Guaranteed Time from the SPIRE instrument team, ESAC scientists and a mission scientist. The HerMES data was accessed through the Herschel Database in Marseille (HeDaM- http://hedam.lam.fr) operated by CeSAM and hosted by the Laboratoire d'Astrophysiquede Marseille.

This paper is based (in part) on data obtained with the International LOFARTelescope (ILT). LOFAR (van Haarlem et al. 2013) is the Low Frequency Array designed and constructed by ASTRON. It has observing, data processing, and data storage facilities in several countries, that are owned by various parties (each with their own funding sources), and that are collectively operated by the ILT foundation under a joint scientific policy.

The Joan Oró Telescope (TJO) of the Montsec Astronomical Observatory (OAdM) is owned by the Catalan Government and operated by the Institute for Space Studies of Catalonia (IEEC). Data author: IEEC - Kike Herrero.

This research uses data from the VIMOS VLT Deep Survey, obtained from the VVDS database operated by Cesam, Laboratoire d'Astrophysique de Marseille, France

This publication makes use of data products from the Wide-field Infrared Survey Explorer, which is a joint project of the University of California, Los Angeles, and the Jet Propulsion Laboratory/California Institute of Technology, funded by the National Aeronautics and Space Administration.

References

[1] J.-L. Stark and F. Murtagh, "Handbook of Astronomical Data Analysis," *Springer-Verlag, Heidelberg-Berlin*, 2002.

[2] K. Borne, "Data Science Challenges from Distributed Petascale Astronomical Sky Surveys," *DOE Conference on Mathematical Analysis of Petascale Data*, 2008.

[3] M. Bourque et al., "The James Webb Space Telescope Quicklook Application (JWQL)," *Astronomical Data Analysis Software and Systems XXIX. ASP Conference Series*, vol. 527, no. 10, pp. 539, Oct. 2020.

[4] T. de Zeeuw; R. Tamai and J. Liske, "Constructing the E-ELT," *The Messenger*, vol. 158, pp. 3–6, Dec. 2014.

[5] M. Romaniello at al., "From Chile to Europe in Minutes: Handling the Data Stream from ESO's Paranal Observatory," *Proc. SPIE 7737, Observatory Operations: Strategies, Processes, and Systems III*, Jul. 2010.

[6] W. D. Pence; R. Seaman and R. L. White, "Lossless Astronomical Image Compression and the Effects of Noise," *Publications of the Astronomical Society of the Pacific*, vol. 121, no. 878, May 2009.

[7] K. S. Dawson et al., "The Baryon Oscillation Spectroscopic Survey of SDSS-III," *The Astrophysical Journal*, vol. 145, no. 1, Jan. 2013.

[8] K. Bundy et al., "Overview of the SDSS-IV MaNGA Survey: Mapping nearby Galaxies at Apache Point Observatory," *The Astrophysical Journal*, vol. 798, no. 1, Jan. 2015.

[9] R. L White; M. Postman and M. G. Lattanzi, "Compression of the Guide Star Digitised Schmidt Plates," *Digitised Optical Sky Surveys*, p. 167, 1992.

[10] S. W. Golomb, "Run-Length Encodings," *IEEE Transactions on Information Theory*, vol. 12, pp. 399 – 401, Dec. 1966.

[11] W. Pence; R. Seaman and R. White, "Fpack and Funpack User's Guide: FITS Image Compression Utilities," *https://arxiv.org/abs/1112.2671*, Dec. 2011.

[12] M.J. Weinberger; G. Seroussi and G. Sapiro, "The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS," *IEEE Transactions on Image Processing*, vol. 9, pp. 1309 – 1324, Aug. 2000.

[13] M. Rabbani, "JPEG2000: Image Compression Fundamentals, Standards and Practice," *Journal of Electronic Imaging*, vol. 11, Apr. 2002.

[14] P. Lindstrom, "Fixed-Rate Compressed Floating-Point Arrays," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2674 – 2683, Dec. 2014.

[15] F. Knorr; P. Thoman and T. Fahringer, "ndzip: A High-Throughput Parallel Lossless Compressor for Scientific Data," *Data Compression Conference*, 2021.

[16] G. N. N. Martin, "Range Encoding: an Algorithm for Removing Redundancy from a Digitized Message," *Video and Data Recording Conference*, 1979.

[17] J. Ziv and A. Lempel, "A Universal Algorithm for Sequential Data Compression," *IEEE Transactions on Information Theory*, vol. 23, pp. 337 – 343, May 1977.

[18] "LZF Compression Filter for HDF5," http://www.h5py.org/lzf/.

[19] N. Ranganathan and S. Henriques, "High-speed VLSI designs for Lempel-Ziv-based data compression," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 40, pp. 96–106, Feb. 1993.

[20] Y. Collet, "Zstandard," *Retrieved from https://github.com/facebook/zstd.*

[21] Y. Collet., "Real Time Data Compression: LZ4 Explained.," *Retrieved from http://fastcompression.blogspot.ru/2011/05/lz4- explained.html*, 2011.

[22] J. Seward, "BZIP2 ," *http://www.bzip.org/.*

[23] A. G. Villafranca J. Portell de Mora and E. García-Berro, "Quick Outlier-Resilient Entropy Coder for Space Missions," *IEEE Transactions on Computers*, vol. 18, no. 10, pp. 2230–2242, Jul 2010.

[24] S. Claggett; S. Azimi and M. Burtscher, "SPDP: An Automatically Synthesized Lossless Compression Algorithm for Floating-Point Data," *Data Compression Conference*, vol. 58, no. 1, pp. 18 – 31, Jan. 2018.

[25] A. Yang; H. Mukka; F. Hesaaraki and M. Burtscher, "MPC: A Massively Parallel Compression Algorithm for Scientific Data," *IEEE International Conference on Cluster Computing*, p. 381–389, Sep. 2015.