

A Historical Perspective on Approaches to Data Compression

Virendra Nikam, Sheetal Dhande

Computer Science and Engineering, Sipna College of Engineering and Technology, Amravati, India

Email address:

virendranikamphd@gmail.com (Virendra Nikam), sheetaldhandedandge@gmail.com (Sheetal Dhande)

To cite this article:

Virendra Nikam, Sheetal Dhande. A Historical Perspective on Approaches to Data Compression. *Mathematics and Computer Science*. Vol. 8, No. 3, 2023, pp. 68-72. doi: 10.11648/j.mcs.20230803.11

Received: July 23, 2022; **Accepted:** August 9, 2022; **Published:** July 11, 2023

Abstract: Several data compression algorithms are investigated in this study. Data compression is commonly utilized in the community. Because data compression allows us to conserve storage space, it can also assist to speed up data transport from one point to another. It is vital to have a compression tool on hand when compressing from one person to another. This method can be used to make data smaller. In addition to text data, images and video may be saved. Lossy and non-lossy compressions are the two types of compression techniques. Compression (lossless) and compression (lossy) which is, nevertheless, the most widely used? It is necessary to conduct lossless compression. Huffman, Shannon Fano, and other lossless compression techniques, as well as Tunstall, Lempel, Ziv Welch, and run-length encoding, are all instances of runlength encoding. This article explains how a compression strategy works and which approach is most typically used in data compression. A form of compression is text compression. The consequences of this process may be seen in the compressed file size, which is less than the original file. In this article, many data compression techniques are surveyed, including those developed by Shannon, Fano, and Huffman. Data compression seeks to increase active data density by minimizing redundant information in data that is stored or sent. Storage and distributed systems are two domains where data compression is crucial. Information theory ideas are thoroughly examined in relation to the objectives and assessment of data compression techniques. The algorithms that are presented are subjected to a framework that is created for the evaluation and comparison of approaches.

Keywords: Data Compression, Compression Algorithms, Loss-Less Compression, Shannon Fano, Huffman, Tunstall, RLE, LZW

1. Introduction

With the help of software and hardware and the fast growth of technology, it is getting easier and easier to spread information quickly around the world on the internet. Professionals in information technology can easily send information through the internet as a way to get in touch. Not all information, though, can be given this way. There is a very big size that can slow down the speed at which data moves and save computer storage. When information or data is sent or exchanged too quickly [1, 3, 6], it can be fixed by "compressing" the information or data. This can save storage space and let data move at the same time. Compression is the process of turning a set of data into a code so that less space is needed to store and send the data. By compressing data, you can save both time and space in your memory. There are many effective ways to use compression algorithms, such as

the Huffman [18], Lempel, Ziv Welch, Run Length Encoding, Tunstall, and Shannon Fano methods. Figure 1 shows the way in which data is compressed. Compression technique When the data is not compressed, the uncompressed data is continued and processed using a lossless compression algorithm, and the compressed data has a smaller size than the file before it is compressed. Compression is the process of shrinking a file from its original size to a smaller size. A compression test will be done to facilitate the procedure. Transmission of a large file with several attachments in characters The workings of compression are identified by looking for recurring patterns in data and replacing them with a certain pattern sign. Lossless and lossy compression are the two types of compression [7, 8, 10, 12]. Both lossy and lossless compression are employed. When original data is

replaced with compressed data, the process becomes more concise while keeping the same amount of information. Lossy compression is the process of taking original data and turning it into compressed data with different values. The value of this difference is taken into account to make sure that important information from the original data is not lost.

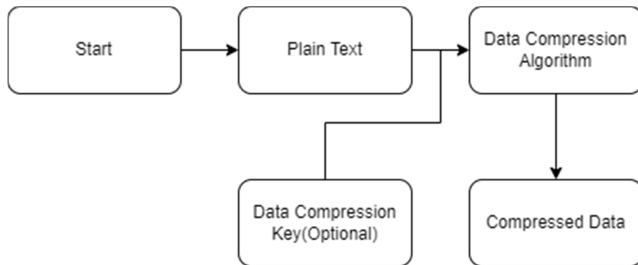


Figure 1. Data Compression Process Flow.

Here is an explanation of the data compression application.

- 1) Compression of audio
- 2) Text Compression
- 3) Video Compression
- 4) Image Compression

A. Compression of Audio

Audio compression is a type of data compression that can be used to reduce the size of an audio or video file [11].

- 1) MP3 and Vorbis are both overflow formats.
- 2) FLAC is a format for music that can hold an unlimited number of files.

Compression happens both when audio files are made and when they are sent to people. Some problems with audio compression:

- 1) Sound recording technology changes quickly and in many ways.
- 2) A sound sample's value changes very quickly.

There are a lot of high-quality audio codecs. The following uses may be given more weight:

- 1) Rates of packing and unpacking
- 2) The amount of pressure
- 3) Help with software and hardware.

Audio codecs that aren't there can be found on:

- 1) How the sound is
- 2) compression coefficient
- 3) Compression and decompression rates
- 4) The inherent delay in the algorithm (required for real-time operations).
- 5) Help with software and hardware.

B. Text Compression

The file is reset to the beginning of the text during the decompression process. Decompression outcomes are influenced by the type of compression utilized, particularly if it was loss- less compression. Lossy compression is another option. When a method for compressing text without losing information is discovered, the original text can be restored after an operation is performed on it. Sayood (2001) demonstrates how to extract the correct data from a decompressed file. Arithmetic encoding is a method of compression in which no data is lost. Compression is terrible [19]. When a file is compressed, some

information is lost. the decompression results can't be identical to the original text (Sayood, 2001).

C. Ratio of Compression

The Compression Ratio shows how much of the original file has been compressed compared to how big the file was before. The difference in size is the size of the original file minus the size of the compressed file. The compression ratio goes up the better the compression works. This is because a compression file is smaller when the compression ratio is high.

$$\text{Compression ratio} = \frac{x}{\text{original file}} * 100$$

Figure 2. CR Ratio.

D. Video Compression

Video is a way to take pictures, record them, edit them, send them, and put them back together again. Celluloid film, electric signals, and digital media are the most common types of media. To digitize and store full-motion video clips for 10 minutes on a computer, a lot of data needs to be sent in a short amount of time. To make a single frame of a 24-bit digital video component, the computer needs about 1 MB of data. There's no need for a video. The hard disc will be reached after 30 seconds of changing layers. GB was accused. For full-size and full-motion videos, you need a computer that can send about 30 MB of data per second. Second, big technological problems could be solved. codecs, or digital video conversion techniques (coders and decoders). Codecs are ways to change (encode) a video so it can be sent and then instantly decode it so it can be played back quickly. Different codecs have been made to work best with different delivery methods. (This could be from hard drives, CD-ROMs, or the Internet, for example.) The goal of video compression/conversion is to lower the bit rate of the digital video representation signal while keeping the right amount of signal quality and minimizing the complexity of codecs (encoding and decoding) as well as delay or deterioration. In other words, data is what video compression is. A method for making a video file smaller is called "video compression." Video compression is a way to cut down on the amount of data needed to do a job. Image compression space and image compression time motion depict a digital video picture.

E. Compression of Images

CompuServe made the Graphic Interchange Format (GIF) in 1987 so that it could store a lot of bitmap images in a single file that could be easily changed over a computer network. GIF was the first one. GIF is a file format for pictures that can be used on the Internet. It can handle up to 8-bit pixels. making use of as many colors as possible One way to compress images is to use 256 colors (28 colors times 256), 4-pass interlacing, transparency, and the Lempel-Ziv Welch (LZW) algorithm [5, 6, 9, 10]. There are two kinds of GIFs, and GIF87a is one of them. It can support both interlacing and multiple files. The way of doing things was called "GIF87" because it was found and made standard in 1987. The GIF89a standard adds to what the GIF87a standard said. changes to how transparency, text, and animated text

look It was made in a format called Portable Network Graphics (PNG). GIF was made legal so that it could work better with the old format. PNG is meant for lossless image storage techniques [17, 21]. PNG images have some of the same qualities as GIF images, which are (many photos) getting better at everything (like interlacing and compression) and adding the newest features. Support for the Web, including the ability to build plug-ins for web browsers Joint Photographic Experts Group, or JPEG, is a file format that is used to reduce the size of a full-color or grayscale image, like the original world scene. JPEGs work well with images that have a continuous range of tones, like photos and art that tries to look real, but they aren't very good at sharpness and color. You can already use JPEGs to do things like make simple cartoons or draw with a lot of lines. JPEG supports 24-bit color depth, which means it can show 16,777,216 different colors. Another benefit of JPEG is that it looks like it's going in the same direction as a terlaced GIF. JPEG 2000 is the most up-to-date way to compress images. JPEG was created in the year 2000. It has a high bit rate. Compared to Jpeg, they have better quality, lower transmission, and are graded. Lossy and lossless compression are both used in JPEG 2000 Compression techniques. Also, ROI coding, which stands for "Region of Interest," is used. interest). JPEG 2000 is made for use on the Internet, for scanning, and for other similar tasks. Remote sensing, medical imaging, and digital photography are all types of digital imaging. The library and online shopping Since the 1980s, the International Organization for Standardiza- tion (ISO) and the International Telecommunication Union (ITU) have worked together to set standards. There should be a standard for grayscale and picture compression. Photographers can use images made by the Joint Photographic Experts Group, or JPEG. In response to the fast growth of multimedia technology, which needs high-performance compression techniques, a new standard was made in March 1997. JPEG is an effort to reduce the size of images that began in 2000. This study made a new way to code different kinds of images (bi-level, grey-level, color, and multi-component) that have different qualities (Natural Images, for example). (Such as science, medicine, remote sensing, text, etc.)

2. Literature Review

2.1. Shannon Fano Algorithm

Data compression is a method in which a compression approach is used to make a very useful compression method that is used in a zip file or. rar format [2]. The Shannon-Fano algorithm can be made to work with VHDL. The ModelSim SE 6.4 simulator is being used to write code and collect data. The algorithm is used to pack things together. If we want to know how these algorithms compress a lot of data, we can look at the equation below: The ratio of data before compression to data after compression is the amount of compression. We know that encryption can make data compression better. Unfortunately, most standard Shannon

Fano codes aren't very long, and this research [1] gave two algorithms. To make the Shannon-Fano code shorter, In some applications, the improved Shannon-Fano coding method is used. extremely helpful [1].

2.2. Run Length Encoding

The Run Length Encoding (RLE) algorithm is one way to compress data so that the size of the output is smaller than the size of the original data [4]. As an example, this time shows the cost and benefit of data from a sentence [12, 13]. RLE (Run Length Encoding) is the easiest type of data encoding that doesn't lose any information. It is a method for compressing a group of files that all have the same format. Each number in the series will be written down as a piece of data. The name of this algorithm is very good when there are a lot of data points with the same value. Sequences are made up of things like icon files, line drawings, and animations. This Normal data doesn't work well with the method. The Run Length Encoding Algorithm is shown in Figure 3 as a flow chart.

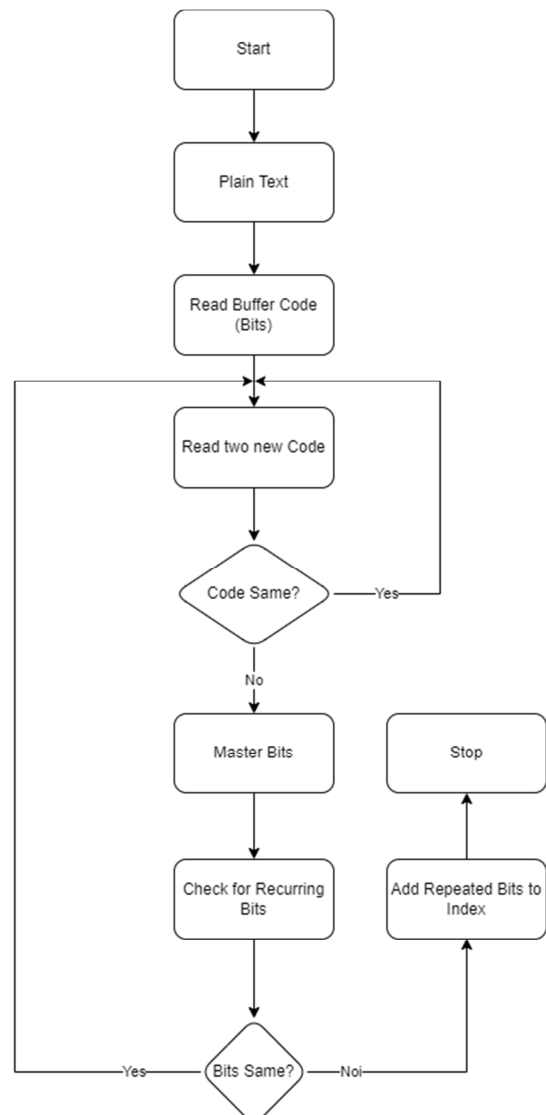


Figure 3. RLE Algorithm.

2.3. Lempel Ziv Welch

In general, the LZW algorithm is a way to compress data without losing any of it. It does this by using a dictionary. LZW compression makes a dictionary as part of the compression process. There are a lot of ways to compress data applications. In this experiment, [5] has made a similar point. Between the usual LZW coding and the MLZW coding that was proposed [5], Coding The result of compression is a phrase made up of words. Output the amount of bit or codeword compression that the LZW algorithm can achieve Before compression, the output must be smaller than the original file. The formula is a It has been updated to the Unicode standard and can be used very easily for any Bangla text compression [5]. The idea behind this method is to start with a new character to make a new dictionary. In the LZW method, a variable word width dictionary is used to balance the compression and decompression of a file [20]. The Lempel Ziv Welch Algorithm is shown in Figure 4 as a flowchart.

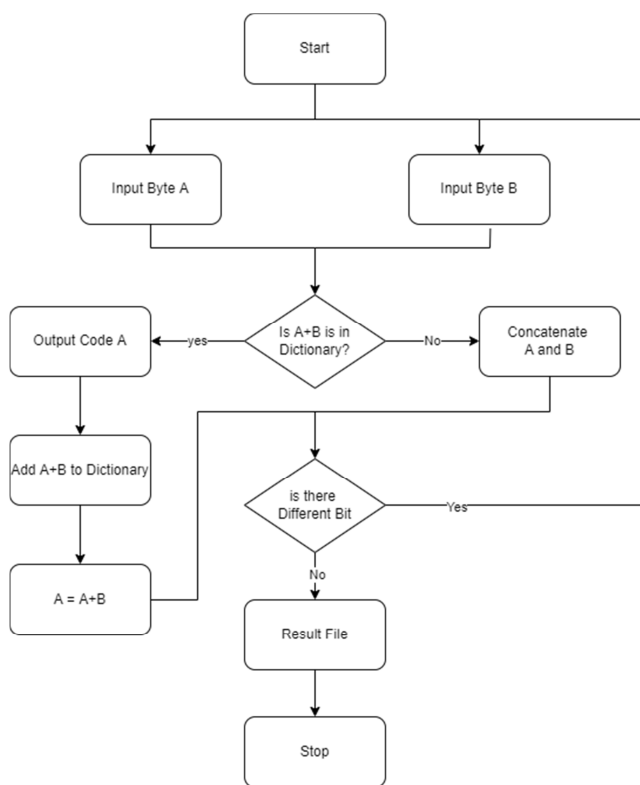


Figure 4. LZW Algorithm.

2.4. Huffman Technique

The number of characters that show up often tells you how to make a Huffman code [14-16]. The Huffman code frequency goes up as the number of bits created goes down. On the other hand, the less character appearances there are, the more bits are made. When being pressed, This Huffman compression technique has ways to do things. It is compression without loss. Lossless compression is a type of data compression that doesn't change the original data while

making it smaller. Huffman's method is based on the fact that 8 bits are usually used to represent each ASCII character. So, if there is a row of the letters "ABCD" in a file, there are 40 bits in the file, or 5 bytes. If every character is given a name, If we have a code like "A = 0," "B = 10," "C = 111," and "D = 110," all we need is a 10 bit file (0010111110). Pay attention to the fact that codes must be unique; otherwise, they can't be copied. built on top of other codes. We can use the Huffman method to compress and make codes that describe codes that must be unique. The number of bits is used to shorten the characters that show up most often. The Huffman Algorithm is shown in Figure 4, which is a flowchart.

3. Conclusion

The compression method can be used to make the file sizes as small as possible. In order to save space on a computer's hard drive, large amounts of data are shrunk down. It is possible to utilize data compression. using different compression algorithms on text, image, and video data. When it comes to compression, different techniques have pros and cons.

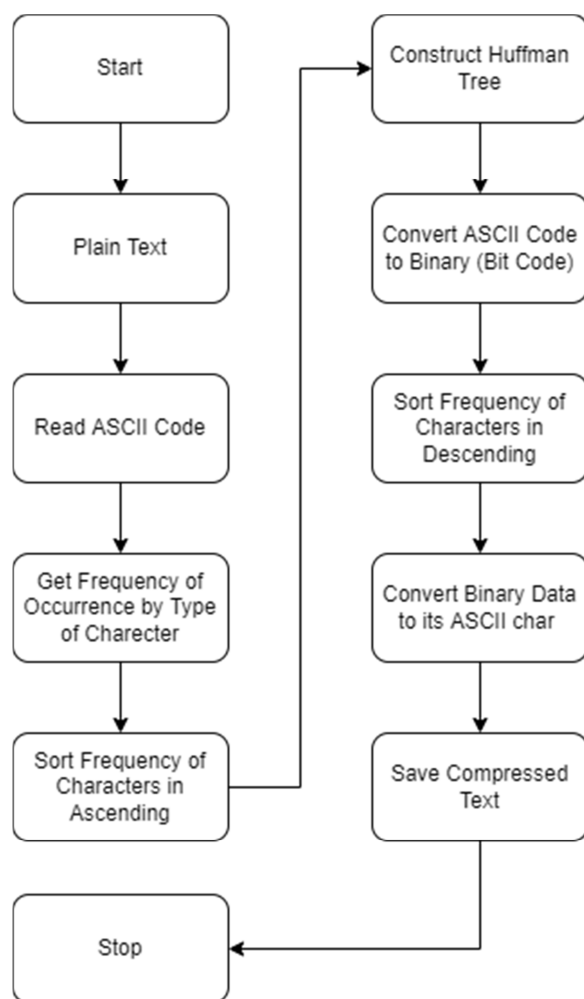


Figure 5. Huffman Technique.

References

- [1] "Using Improved Shannon-Fano-Elias Codes Data Encryption," Xiaoyu Ruan and Rajendra Katti, Proceedings of the ISIT Conference, North Dakota State University Fargo, North Dakota, July 9–14, 2006.
- [2] "Data Compression Using ShannonFano Algorithm Implemented By VHDL," by Mahesh Vaidya, Ekjot Singh Walia, and Aditya Gupta, IEEE International Conference on Advances in Engineering Technology Research, August 1–02, 2014.
- [3] "Pattern Run-Length for Test Data Compression," Lung-Jen Lee, Wang- Dauh Tseng, Rung-Bin Lin, and Cheng-Ho Chang, IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems, Vol. 31, No. 4, April 2012.
- [4] "Run Length Encoding for Speech Data Compression," Mohammad Arif and R. S. Anand, IEEE International Conference on Computable Intelligence and Computing Research, 2012.
- [5] "Bangla Text Compression Based on Modified Lempel-Ziv-Welch Algo-rithm," Linkon Barua, Pranab Kumar Dhar, Lamia Alam, and Isao Echizen, International Conference on Electrical, Computer, and Communication En- gineering (ECCE), Bangladesh, February 16–8, 2017.
- [6] "Compound IET "Image Compression Using Parallel Lempel Ziv-Welch Algorithm", IET Chennati Fourth International Conference on Sustain- able Energy and Intelligent Systems, Cincinnati, December 12–14, 2013. G. R. Gnana King, C. Seldev Christoper, and N. Albert Singh.
- [7] "A Data Comprssion Technique Based on Resersed Leading Bits Coding and Huffman Coding," Haoqi Ren, 2015 International Conference on Com- munication and Networking, China.
- [8] "Adaptive Image Compression Using Adaptive Huffman and LZW," Djuned Fernando Djusdek, Hudan Studiawan, and Tohari Ahmad, International Con-ference on Information, Communication Technology, and Systems, 2016.
- [9] "Enumerative Implementation of the Lempel-Ziv-77 Algorithm," Tsutomu Kawabata, ISIT, Toronto, Canada, July 6–11, 2008.
- [10] "A Comparison of Algorithms for Lossless Data Compression Using the Lempel-Ziv-Welch Type Methods," Adrian Traian Murgan and Radu Rade- scu. Bucharest.
- [11] "Implementasi Algoritma Kompresi Data Huffamn Untuk Memperkecil Ukuran File MP3 Player," Victor Amrizal, February 14, 2010.
- [12] "Implementasi Algoritma Run Length Encoding Untuk Perancangan Ap- likasi Kompresi dan Dekompresi File Citra," Jurnal TIMES, Vol. V No. 2, 24–31, 2016. CutTry Utari, "Implementasi Algoritma Run Length Encoding Untuk Perancangan Aplikasi Kompresi dan Dekompresi File.
- [13] "Modified Run Length Encoding Scheme for High Data Compression Rate," M. VidyaSagar, J. S, and Rose Victor, International Journal of Advanced Research in Computer Engineering Technology (IJARCET), Vijayawada, December 2013.
- [14] "Implementation of Data Compression Using Huffman Coding," K. Ashok Babu and V. Satish Kumar, International Conference on Methods and Models in Computer Science, India, 2010.
- [15] "Data compression with Huffman and other algorithms," Harry Fernando, ITB, Bandung.
- [16] "Comparative Study of Different Data Compression Techniques," Mo-hammed Al-Laham and Ibrahim M. M. El Emary, IJCSNS International Journal of Computer Science and Network Security, Jordan, April 2007.
- [17] "Comparison of Lossless Data Compression Algorithms for Text," S. R. Kodituwakku and U. S. Amarasinghe, Indian Journal of Computer Sci- ence and Engineering, Sri Lanka.
- [18] Rhen Anjerome Bedruz and Ana Riza F. Quiros, "Comparison of Huffman Algorithm and Lempel-Ziv Algorithm for Audio, Image, and Text Com-pression," IEEE International Conference on Humanoid, Nanotechnology, Information Technology Communication and Control, Environment, and Management (HNICEM), Philippines, December 9–12, 2015.
- [19] "Knowledge Engineering Perspective of Text Compression," by C. Oswald, Anirban I. Ghosh, and B. Sivaselvan, was published in 2015 by IEEE INDICON in India.
- [20] "Development of Word-Based Text Compression Algorithm for Indonesian Language Documents," by Ardiles Sinaga, Adiwijaya, and Hertog Nugroho. International Conference on Information and Communication Technology (ICoICT), Indonesia, 2015.
- [21] International Journal of Advanced Research in Computer Science and Software Engineering, India, July 2015, Manjeet Kaur, "Lossless Text Data Compression Algorithm Using Modified Huffman Algorithm."