2023 International Conference on Networking and Communications (ICNWC) | 979-8-3503-3600-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICNWC57852.2023.10127236

COMPRESSION AND DECOMPRESSION OF FILES WITHOUT LOSS OF QUALITY

K. Anand
Department of Computer Science and Engineering
Rajalakshmi Engineering College
Chennai, India
anand.k@rajalakshmi.edu.in

M. Priyadharshini
Department of Computer Science and Engineering
Rajalakshmi Engineering College
Chennai, India
priyadharshini.m.2019.cse@rajalakshmi.edu.in

K. Priyadharshini
Department of Computer Science and Engineering
Rajalakshmi Engineering College
Chennai, India
priyadharshini.k.2019.cse@rajalakshmi.edu.in

***Abstract*— This paper proposes the Gzip algorithm for image and document compression and decompression. Gzip is a hybrid algorithm that combines Lz77 and Huffman. In document management and communication systems, picture and document compression and decompression are crucial. Image and document compression technologies are used to lower the amount of data required to represent the file. Image compression has proven to be the most advantageous and practical method in the field of digital image processing. The goal is to reduce the images' and documents' redundancy so that data may be stored or sent efficiently. In order to reduce data redundancy and conserve more hardware space and transmission bandwidth, the theory of data compression and decompression is therefore becoming more and more important. Compression is beneficial because it makes use of less expensive resources like hard disc space and transmission bandwidth. When we evaluate the image quality, decompression is beneficial. In the proposed system, there is no reduction in data, but there is a decrease in data size without loss of quality.**

Keywords-Lz77;Gzip;Hybrid algorithm

I. INTRODUCTION

The creation of and demand for multimedia products have both accelerated recently. The theory of file compression is consequently becoming more and more significant in order to decrease data redundancy, decrease hardware space, and enhance transmission bandwidth. Data encoding using information-carrying units or fewer bits than the original file is known as file compression or source coding. Transmission bandwidth increased and reduces the need for hard drive storage, compression process is advantageous. Compression is a handy tool for reducing file size without sacrificing the file's quality or data integrity. When the quality of the photographs is taken into account, decompression is advantageous. Compression system is essential for faster and efficient data transfer without loss of quality.

In the fields of databases, storage, and the Internet, data compression technology is frequently used. GZIP, which has a high compression ratio and enough bandwidth for both compression and decompression, is the most popular compression algorithm. For instance, both Apache and IIS use GZIP compression on web pages to enhance access response. For data compression and decompression in conventional applications, software-based techniques are widely used. However, extensive CPU and memory resources are required for data compression and decompression. The platform offers a decompression bandwidth of 200 Mega Bytes per second using a 2.66GHz CPU as an example. Consequently, the deal was successful. There are only a few CPU and memory resources left.

Our suggested system can considerably increase system capacity by offloading the CPU utilisation. The Gzip algorithm is used in this case. Gzip uses the LZ77 and Huffman algorithms for both compression and decompression. There were two operations done.

For compression, first the input file is processed by LZ77, and the output of LZ77 is provided as input to the Huffman encoder, which divides the input into two parts. Then the compressed file is produced.

For decompression, first the input file is processed by the Huffman decoder, and then the output of the Huffman decoder is fed as input to the Lz77 decoder, which produces the decompressed file. It decompresses compressed as well as uncompressed file. Decompression is the reverse of compression process. It maintains the quality of file without loss of data.

The gzip compression algorithm already exists and is designed to work on unix-like systems. It is especially suitable for files that are used on the web. In some cases, we cannot download the file after compression, which creates a lot of problems in today's world. If the file size is larger and the file contains more redundant data, it creates a lot of problems while transferring a file. Compression helps us reduce the size of the file so that the redundancy of the file is reduced while still maintaining the quality of the file without loss of quality or data.

II. RELATED WORK

Platforms for social networking are popular today. As a result, the amount and size of data shared online are growing quickly. A network's bandwidth will never be unlimited. Therefore, it is important to select an efficient algorithm because data sharing via a network will be quicker and more efficient [13]. Sometimes, if we transmit a larger amount of data, it may be more expensive [1]. In the New Approach technique, LZW is a widely used algorithm that is used to increase the compression ratio using the LZW lossless algorithm [2]. Compression of data has become more essential in the data industry. This compression will be efficient, but the decompression process will have some queries repeated repeatedly. When we compress data, it takes a lot of time to decompress. So decompression will be accomplished in various ways, such as assigning one of the individual block of data to a different process, breaking them to fast up data compression [4]. Throughout the network, JPEG image compression technology is frequently employed.JPEG uses the Discrete Cosine Transform (DCT) to compress images by breaking them up into non overlapping 88-pixel chunks [6]. In this case, we know how to efficiently decompress an algorithm using LZ parsing. The only recent work on this problem that we are aware of is by Billeetal, who describes an algorithm for any 0 1 logs that uses O(n log time and O(z log1 -) space [7]. Iteratively applied column folding and row folding are employed in several compression processes until the image size is reduced to a lower pre-defined value. Decompression of text files using dynamic compression is a compression algorithm used to reduce data redundancy and minimise data space requirements. It uses arithmetic encoding,the LZ family, and dynamic Markov compression techniques. In this compression for text files coupled with dynamic compression, instead of taking an entire file to decompress, the user will take only a sectional part to decompress the data [8]. Polynomial Representation is used for compression. With a hybrid lossy and non-lossy coding scheme that used a two-dimensional polynomial with variable block size in accordance with the correlation between neighbouring pixels to represent colour contents. Paper Huffman tables are used for image retrieval and compression in very Fast Image Retrieval based on JPEG. Analysis and comparison of arithmetic and adaptive Huffman coding algorithms concentrated on experiments like the systematic sampling and compression of a set of image files. It was found that a fast arithmetic coding variant consistently produced quick compression times and significant space savings for image files and documents after incrementally and systematically compressing a subset of sample files with the two chosen general compression methods. Instead of providing the data in its original or uncompressed form, compression has the unique ability to show it in a conservative format. Moreover, the size of a particular document can be reduced by using information compression. This is quite useful when producing or transporting a massive document that requires a lot of resources. The number of components conveyed, the time required for the encoder to build the coded message, and the time needed for the decoder to restore the owner ensemble all influence how quickly information is transmitted. The degree of compression is the main issue in an application for storing information. lossless and lossy are two types of compression. The primary data from the compacted record is recreated using lossless compression techniques. There are two types of compression: lossless and lossy. The primary data from the compacted record is recreated using lossless compression techniques. In this way, the decompression and compression processes do not cause any changes to the data. These types of compression calculations are known as reversible compressions so because the primary message is recreated during the decompression cycle. This study looks at the Huffman algorithm, the LZW Algorithm, the Arithmetic Encoding Algorithm, the Adaptive Huffman Encoding Algorithm, the Shannon Fano Algorithm, and the RLE Algorithm. In particular, the effectiveness of these algorithms is evaluated when it comes to compressing text data. The file has been placed on transmitting data and compressing data. Data compression before transmission and data compression before storage can both reduce bandwidth requirements. Random Access Memory will be used to become overflowing with data if the same data is continuously encoded. This study introduces the Varint encoding technique to address this issue. On large amounts of repetitive and continuous data, varint has good compression effects. As a result, Varint LZW encoding uses less RAM bit width than LZW encoding for this kind of data. In existing system Lzw and Huffman algorithm sometimes while compressing the files it produces file size same or more than that of original file size. Without loss of quality file size is reduced is our main satisfied which is not possible in existing system. In data processing, strangulation and bandwidth are the major drawbacks in compression and decompression. GZIP Algorithms are used for data compression as a storage concern, so we improve a GZIP algorithm with adaptive Huffman coding by replacing it with Huffman coding in this paper. Additionally vector quantizaion is used to decrease the BMP images range of bytes which increased the space saved as a result of compaction using the fast AC variation.

.

III. THE PROPOSED FRAMEWORK

The architecture diagram for the suggested technique is displayed in Figure 1. In our system, we compress images and documents based on the intensity of pixels and the frequency of characters, respectively. If there are 100 distinct pixels, there are 100 distinct pieces of data. It creates a compressed file from any input file. It is unable to delete essential info from files. Based on the frequency of characters in texts and the pixel intensity for images, it eliminates unnecessary data. At the same time, it keeps the file's quality. It succeeds in increasing compression ratio. When the file is decompressed, the system creates a new, higher-quality file. The gzip technique is used to compress files on local machines and was designed to run on any operating system. Users of this system can download documents and photos that have been compressed. The user has the option of storing the compressed file in a database and retrieving it at a later time. As an extension, a compression system is created.

Our proposed method makes use of the Gzip compression and decompression technology, which preserves file quality. Gzip is a hybrid algorithm that combines the Huffman and Lz77 algorithms. The Lz77 algorithm is used to process the input file first. The input data are analysed using the LZ77 Compression Algorithm to find how to minimise their size by removing unnecessary information and substituting it with metadata.

The Huffman encoder is then given the Lz77 encoder's output. The traditional method of data compression is huffman coding. It is an efficient prefix code that has been employed in a number of compression applications. These codes use an amount of bits and have various code lengths

In our system, we perform compression as well as decompression. By using compression, the amount of bits needed to represent a file is decreased. The opposite of compression is decompression.

Our key objectives are to decrease data redundancy, conserve more hardware space, boost transmission bandwidth, and retain the file's quality without any data loss.

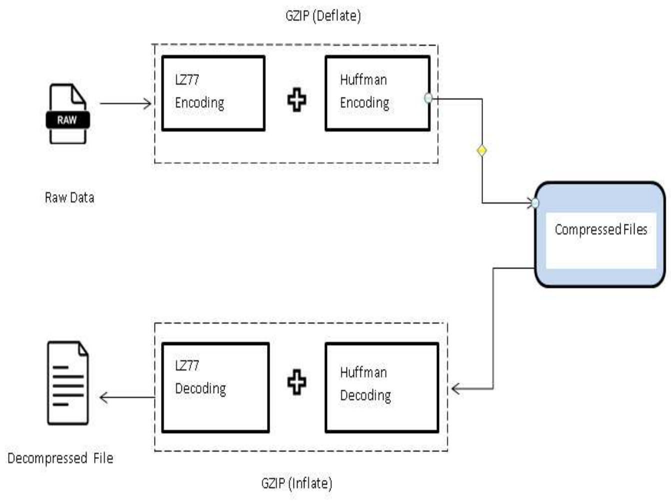

Figure 1. Architecture Diagram

IV. ANALYSIS OF GZIP

The DEFLATE method, which includes the LZ77 and Huffman coding algorithms, is the foundation of the Gzip algorithm.

The Lz77 method reduces the size of the raw data by removing unnecessary elements. The outcomes of Lz77 are encoded using Huffman coding.

The software implementation of the Lz77 algorithm uses a "sliding window" to identify the best match, eliminating duplicate data strings; a typical window size is 32 KB. To locate every instance of the matching, such a window must be searched, which makes processing extremely slow. The speed may be improved by lowering the window (for example, to 16 KB), but the compression ratio would suffer.

The preferred method in GZIP compression is Huffman coding. The output of Lz77 will be split into two sections: equal length. In Figure 2, the process flow is displayed.

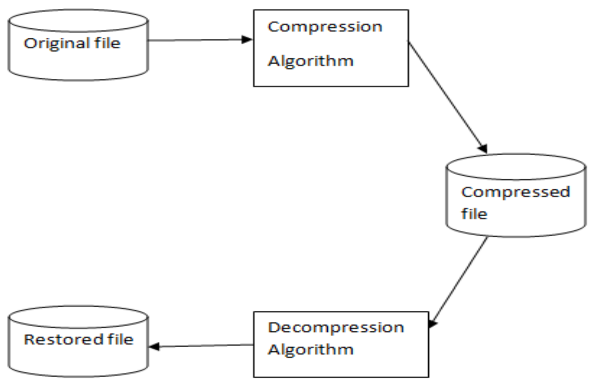

FIGURE 2. Process Flow

V. METHODOLOGY

GZIP algorithm, a hybrid method, is used in the proposed methodology. Its basis is the Deflate technique, which enables file compression to speed up network transfers. Working of GZIP algorithm is shown in Figure 3. In this LZ77 and Huffman algorithms are combined. A lossless data compression algorithm is called LZ77. Using the LZ77 method, it is possible to identify redundant information and determine how to replace it with metadata to minimise the amount of the input data. Although the LZ77 technique can offer a high compression ratio by itself, the file's quality cannot be preserved. In order to maintain quality while compressing, we are combining it with Huffman coding. Each character in a text file that is input is given a variable-length via Huffman algorithm.

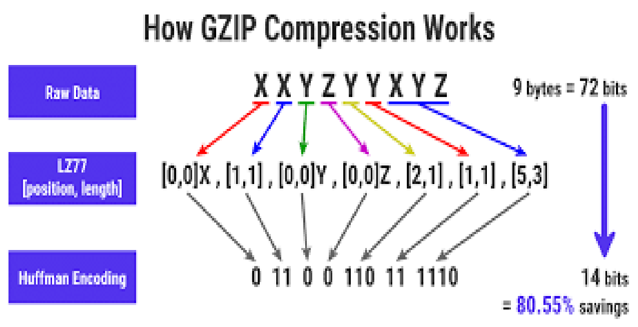

FIGURE 3. WORKING OF GZIP

A. Lz77 Algorithm

The input data are analysed using the LZ77 Compression Algorithm to find how to minimise their size by removing unnecessary information and substituting it with metadata. Sections of the data that are identical to parts of the encoded data are replaced with a small bit of metadata that explains how to expand those parts again. This mix of data and information is serialised into a stream of bytes by the encoding method so that it may be later decoded and decompressed.
Because there is no data loss during file compression and decompression, the LZW algorithm is a lossless algorithm. Although it is only occasionally utilised in PDF and TIFF, this method is frequently used in GIF. No data is lost during compression because it is lossless. The method has a very high throughput potential and is simple to implement in hardware. The idea is dependent on repeating patterns to conserve data space. This algorithm is the most popular method for all data compression because its ease of use and adaptability. The fact that the encoding and decoding tables are generated on the fly without any knowledge of the input is another crucial aspect of the LZW compression approach. LZW compression makes use of a code table. A typical choice is to offer 4096 table entries. The LZW-encoded data in this case is entirely made up of 12 bit codes, where each bit code corresponds to a different item using the code table. Individual bytes from the incoming file are consistently represented in the coding table by the codes 0-255. The block diagram of Lz77 is shown in Figure 4.

Implementing the LZ77 Compression Algorithm

1. Set the input stream's first frame as the coding position.
2. In the lookahead buffer window, locate the longest match.
3. The pointer P is output if a match is discovered. The window and coding location are both done using advanced L bytes.
4. Output a null pointer and the first byte of the lookahead buffer if a match cannot be found. One byte is added to both the window and the coding location.
5. Return to step 2 if the lookahead buffer is not empty.

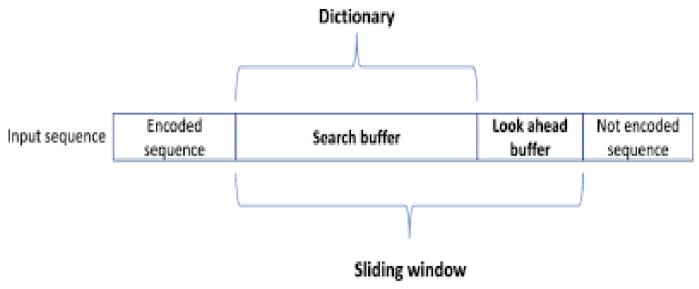

FIGURE 4. BLOCK DIAGRAM OF LZ77

1. Compression Process:

The next step is to output a pointer to the longest match that the compression algorithm could find between the window and the start of the lookahead buffer. Since it's possible that not even a 1-byte match will be found, the output cannot just be pointers. The compression technique fixes this by publishing the first byte from the lookahead buffer after the match and after the pointer. If no match is found, the method sends a null-pointer and the byte at the coding location.

2. Decompression Process:

The compressed stream is handled beginning to end by the decompression method. For each null pointer, it immediately inserts the matching byte to the end of the output stream. It reads back from the current end of the output stream to the provided offset for every non-null pointer and adds the required amount of bytes to the output stream's end.

B. Huffman Algorithm

By assigning codes of various lengths to each of its data chunks in accordance with their frequencies in the data, one of the lossless compression techniques, Huffman Coding, aims to lower the total code length of the data. Low-frequency chunks receive significantly longer code than high-frequency chunks when the compression value is less than 1. The Huffman algorithm's block diagram is displayed. in Figure 4.

This Algorithm processes the data to generate a frequency table of the symbols. The table's two least frequently occurring symbols are removed. They are combined and their frequencies are added to form a single node. This node is then returned to the frequency table, and step 2 is repeated until only one node remains in the table. A tree-like structure emerges called the Huffman tree. The Huffman codes for each symbol are generated by traversing the tree.

Huffman's approach is based on the idea that an ASCII character is normally represented by 8 bits. As a result, the amount of count of bits, with the letters "OPQR" in a row is 4 x 8 = 32 bits. We only need a file that is 10 bits in size if individual character is assigned a code, such as O = 0, P = 10, Q = 111, and R = 110. (0010111110). To prevent a code from being formed from other codes, the codes must be distinct.

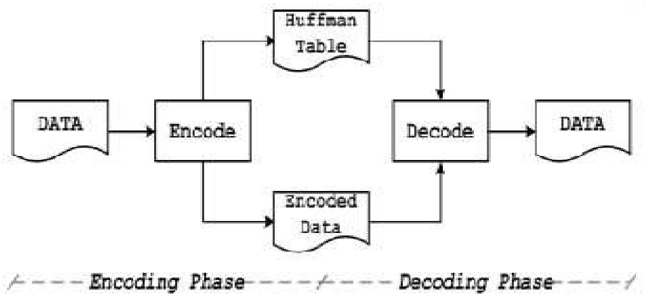

FIGURE 5. Huffman block diagram

Algorithm	Time Complexity
Huffman algorithm	O(nlogn)
Lzw algorithm	O(n)

VI. RESULTS

In file management and communication systems, image and document compression and decompression are essential. It is essential to cope with the enormous volumes of data generated in many forms in the digital world as a compression and decompression approach. The amount of data needed to represent the file is minimized using an image and document compression approach. As a result, while using this strategy, the size of the data is reduced without sacrificing its quality. The processing of files takes less time overall.

VII CONCLUSION AND DISCUSSIONS

Compression and decompression play crucial roles in the world, as we all know. The most beneficial and effective technology in the area of digital image processing has been image compression. In file management and communication systems, image and document compression and decompression are crucial. A technique for compressing images and documents is used to reduce the amount of data required to represent the file. The suggested method shrinks the size of the data without sacrificing its quality, without reducing the total amount of data. In earlier research, only zip files were used as input for compression, and only zip files were produced as output. However when we compress a file, the file size grows, and when we decompress a file, an exact copy of the compressed file is produced, also leading to a loss in quality. There will be more problems with earlier work as a result. We accept any file as an input for our project and generate output in any format depending on our specifications. Without sacrificing quality, a compressed file will be created. It won't delete crucial data from files. It reduces file size while maintaining file quality.

REFERENCES

[1] Y.Suresh(2022),Lossless Image Compression based on Data Folding,2022 International Conference on Recent trends in Information Technology(ICRTIT).

[2] Amande ep Singh Sidhu, Er.Meenakshi Garg(2021) Text Data Compression algorithm using Hybrid Approach.

[3] R. Karthik, V. Ramesh, M. Siva International Conference on Emerging Trends in Applications of Computing (ICETAC 2K17) “Text Data Compression and Decompression Using Modified Deflate Algorithm” 2020.

[4] Praveen Kumar Muvva, P A V Krishna Rao, Prasad G, Srinivasa Rao Namburi The topic of the February 2020 issue of the International Journal of Engineering Applied Sciences and Technology is "A new way to boost LZW algorithm compression ratio."

[5] H. PeterHofstee ,Jian Fang, Jianyu Chen , Jinho Lee, Zaid Al-Ars · (2020) Journal of Signal Processing Systems in 2020.”An Efficient High-Throughput LZ77-Based Decompressor in Reconfigurable Logic”.

[6] G. Lohman and K. Ross,E. Sitaridi, R. Mueller, T. Kaldewey,"Massively-Parallel Lossless Data Decompression," 2020 45th International Conference on Parallel Processing (ICPP).

[7] Anitha Murugan Master of Philosophy(2020) International Symposium on Information Theory and Its Application (2020)“ Lossless Image Compression and Decompression using Huffman Coding”.

[8] B. Rajesh, M. Javed and P. Nagabhushan,(2019) "Automatic Text Line Segmentation Directly in JPEG Compressed Document Images," 2019 IEEE 8th Global Conference on Consumer Electronics (GCCE), 2019.

[9] S. J. Puglisi and M. Rossi, "On Lempel-Ziv Decompression in Small Space," 2019 Data Compression Conference (DCC), 2019.

[10] Amit Jain Sir Padampat Singhania University, Udaipur, Rajasthan, India “Dynamic Decompression of Text File 2019, IEEE 8th Global Conference on Consumer Electronics (GCCE), 2019.

[11] Ahmad Shah, Muhammad Athar, Javed Sethi (2019)EAI Endorsed Transactions on Context-aware Systems and Applications “The Improvised GZIP, A Technique for Real Time Lossless Data Compression”.

[12] Houchen Li; Zhijie Qiu; Lei Luo(2019) 10th IEEE International Conference on Computer and Information Technology “Decompression Method for Massive Compressed Files in Mobile Rich Media Applications”.

[13] Mael Kerbiriou ¨ INRIA Lille , Rayan Chikhi , 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (2019) “Parallel decompression of gzip-compressed files and random access to DNA sequences”.

[14] Roopa P. Huilgol,Vishwanath P. baligar,TanujaR. Patil (2018) International Conference on Circuits and Systems inDigital Enterprise Technology in 2018 “Lossless Image Compression using Proposed Equations and JPEG-LS Prediction”.

[15] J. Uthayakumar,T. Vengattaraman,P. Dhavachelvan (2018), Journal of King Saud University - Computer and Information Sciences“A Survey on Data Compression Techniques: From the Perspective of Data Quality, Coding Schemes, Data Type and Applications”.

[16] Niko Rebenich,T. Aaron Gulliver, Ulrich Speidel,(2018) , Conference: 2018 International Symposium on Information Theory and Its Applications (ISITA)A Note on Lempel-Ziv Parser Tails and Substring Lengths,”A Note on Lempel-Ziv Parser Tails and Substring Lengths”

[17] R. P. Huilgol, V. P. baligar and T. Patil, (2018) International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET), ”Lossless Image Compression using Proposed Equations and JPEG-LS Prediction”.

[18] Apoorv Gupta, Vidhi Khanduja Netaji Subhas Institute of Technology ,Computer Engineering Conference: 2nd IEEE International Conference on Electrical, Computer and Communication Technologies (2017); “Modern Lossless Compression Techniques: Review, Comparison and Analysis”.

[19] F. Saeed H. LuG.E.Hedrick(2017) “Data compression with Huffman coding; an efficient dynamic implementation using file”.

[20] Luluk Anjar Fitriya College Student,Tito Waluyo Purboy Lecturer, Faculty of Electrical Engineering, Telkom University, Bandung, Indonesia. Anggunmeka Luhur Prasasti Lecturer, Faculty of Electrical Engineering, Telkom University, Bandung, Indone (2017) ”A review ofdata compression techniques”.