International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT217113

A Review on Different Types of Lossless Data Compression Techniques

Anshul Gupta*[1], Prof. Sumit Nigam[2]
*[1]M. Tech Scholar, Department of Information Technology, SVIIT, SVVV, Indore, Madhya Pradesh, India
[2]Assistant Professor, Department of Information Technology, SVIIT, SVVV, Indore, Madhya Pradesh, India

Article Info
Volume 7, Issue 1
Page Number: 50-56
Publication Issue :
January-February-2021

Article History
Accepted : 01 Jan 2021
Published : 10 Jan 2021

ABSTRACT

This paper provides different kinds of techniques for lossless data compression and comparison between them. By eliminating redundant bits, data compression decreases the file size. In order to reduce the capacity needed for that data, it decreases the redundant bits in data representation and thus uses the bandwidth effectively to reduce the communication cost. Compression of data saves file volume, network bandwidth and speeds up the transfer speed as well. Lossless and Lossy are the two techniques for data compression. Lossless compression maintains the data properly.

Keywords : Huffman Coding, Shannon-Fano Coding, Compression, Lossless, Lossy, RLE, LZW, LZ77, LZ78, Lempel-Zev Welch

I. INTRODUCTION

Data compression is a method to compress different types of files like text, audio, video such that the original file can be recovered without any data loss. If someone wants to save lots of storage space, this method is useful. It is very convenient to share the compressed file over the internet with the help of the compression, it reduces the size of file so it can be uploaded or downloaded much faster. Data compression is so much important for file storage and distributed systems. There are two types of Data Compression techniques: Lossy Techniques & Lossless Techniques. The compression process in which certain bits or portions of data are omitted is called lossy data compression. And the compression method in which the data or information size is compressed without any loss of bits is known as lossless data compression.

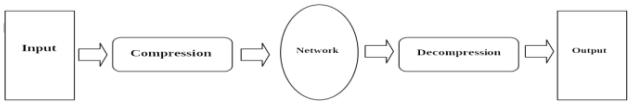

Fig. 1 Compression and Decompression Process

Data compression is the method of compressing data or file to less size than the original file, so it reduces storage space and transmission time for transmitting file over the network. Data compression is practicable since many redundant bits are found in most real-world data. To convert data from a simple-to-use format to one optimised for compactness, data compression is used. An uncompressing algorithm likewise returns the data to its original state. Different types of algorithms are used for data compression some of them are Arithmetic Encoding,

Shannon-Fano, Huffman Coding, and RLE (Run Length Encoding).

II. DIFFERENT TYPES OF DATA COMPRESSION

There are basically two types of techniques available for compression: lossy and lossless. Each technique has same aim to reduce the file size.

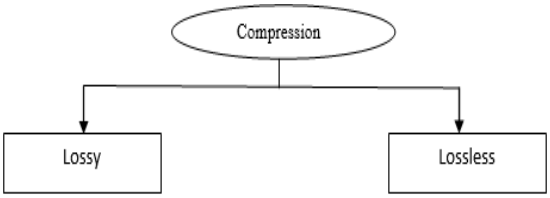

Fig. 2 Types of Compression

A. Lossy Compression

A way of data compression in which some volume of data or data is lost is known as Lossy Compression. For some applications, some loss of information (some bits) is appropriate. In such instances, methods of lossy compression can be used. In order to produce the picture in real time in video conferencing where there is a reasonable amount of frame losses. The loss of information might be in the form of colour depth or graphic information. The process of lossy compression searches for 'redundant' pixel and discard this information permanently. Lossy compression methods are not used for text-based data, because it needs to retain all their data. Lossy data compression is used for MP3, JPEG, MPEG and 3GP, etc.

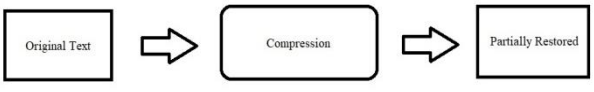

Fig. 3 Process of lossy Compression

B. Lossless Compression

Lossless compression reduces the size of a file without any damage or loss to the file or reduction in quality. In this data is compressed without any loss of data (bits). When it is decompressed the original data or bits are retrieved. It is applied where original bits are necessary. Mostly, it is applicable on text document files and spreadsheet, etc where the data or information is very important. The advantage of lossless compression is that it maintains quality of the data and on contrary it does not reduce too much size of the file.

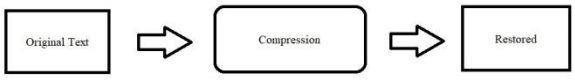

Fig. 4 Process of Lossless Compression

The Lossless data compression techniques can be categorised as :

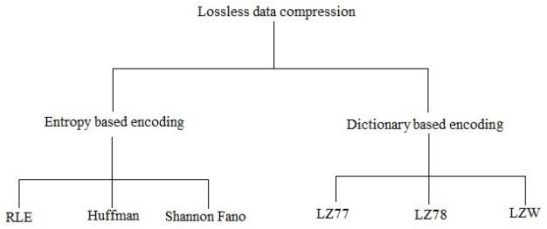

Fig. 5 Types of Lossless data Compression

1. Entropy Based Encoding

Entropy encoding is a type of lossless coding that represents frequently occurring patterns with few bits and rarely occurring patterns with many bits to compress digital content. Huffman coding is a type of entropy coding in which an entropy coder is a general method of lossless data compression that encodes symbols by using several bits that are inversely proportional to the symbols' likelihood. A term used to describe certain compression algorithms or techniques that operate based on the distribution of probability of Huffman coding source symbols. Entropy encoding is the backbone of Huffman coding, Arithmetic coding and Shannon-Fano coding.

2. Dictionary based encoding

The dictionary-based algorithms encode symbol strings of random length as single tokens. The encoder finds the exact match pair of strings from the original text in the dictionary and if this match is found, it replaces it with the dictionary pair.

III. DATA COMPRESSION ALGORITHM

A. Huffman Coding

In 1952, David Huffman discovered the Huffman coding algorithm. Huffman encoding is a binary code tree generation method. This ensures that each symbol's probability of occurrence results in the length of its code. Several data formats, such as ZIP, GZIP and JPEG, include Huffman codes. It is a sophisticated and effective technique for lossless data compression. In this encoding system, the shortest binary code is the symbol with the highest probability, and the longest binary code is the symbol with the lowest probability.

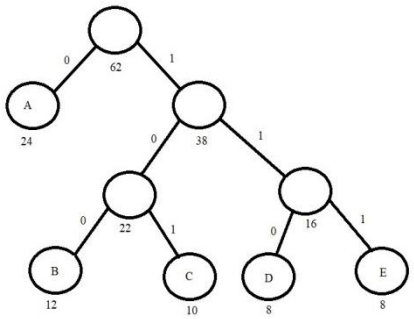

Fig. 5 Binary Tree

B. Run Length Encoding

Run length RLE (Run Length Encoding) is an algorithm for lossless data compression. Sequences of similar bytes are often found in data or information. A reduction in data can be accomplished by replacing these repetitive byte sequences with the number of occurrences. By reducing the physical size of a repeated string of characters, RLE compresses the size of data. The repeating string is known as a run that is usually encoded in two bytes where the first byte shows the total number of characters in the run, which is the run count, and replaces runs of two or more of the same character with a number representing the run length that will be followed by the original characters, and single characters are coded as runs of 1. The text containing the space character is a symbol; it ignores single blanks or pairs of blanks. RLE is useful where data redundancy is high, or it can also be used in conjunction with other methods of compression.

Example of RLE:

- Input string: RRRRRRRRRYYYBBCCCCDEEEEEE
- Output String: 10R3Y2B4C1D6E

The disadvantage of the Run Length Encoding Algorithm is that, it does not achieve high compression ratios as compare to other algorithm, but the benefit of Run Length Encoding is very easy to implement and fast to execute, making it a good alternative for a complex compression algorithm.

C. Shannon Fano Coding

This technique for data compression is discovered by Claude Shannon and Robert Fano in 1949. In this process, a binary tree is generated that shows the probabilities of each symbol that occurs. The symbols are ordered as per their occurrences means the symbols at the top of the tree appear most and the symbols at the bottom are the least likely.

In general, Shannon-Fano coding does not provide any guarantee that an optimal code will be generated. Shannon-The Fano algorithm is more successful when the probabilities are much closer to 2 power inverses.

D. Arithmetic Coding

An optimal entropy coding technique or scheme is an arithmetic coding algorithm as it provides the best compression ratio and produces better results than Huffman coding. When a string is converted to arithmetic encoding, the characters with the highest occurrence are stored with less bits and the characters that do not occur so much are stored with more bits, resulting in less bits being used in overall. Arithmetic encoding transforms the stream of input symbols to the output as a single floating-point number. Arithmetic coding does not separately encode each symbol like Huffman Coding. Each symbol is coded by considering all the prior information. Thus, from the outset, a data stream encoded in this fashion must always be read. Consequently, it is difficult to provide random access.

E. LZ77

The dictionary-based scheme was invented in 1977 for lossless data compression by Jacob Ziv and Abraham Lempel, which is called LZ77. LZ77 is the most likely compression algorithm. It is the compression process, on which applications such as PkZip and a few other algorithms are based. It exploits the fact that it is likely to repeat words and phrases inside a text file. It can be encoded as a reference to an earlier case, if the text includes a repeat, accompanied by a pointer with the number of characters to match. It is very effective and simple approach that does not require any prior knowledge of the source, does not seem to require any assumptions about the source's characteristics. In this approach, the dictionary acts as a portion of the previously encoded sequence. By pressing into the sliding window service that consists of two parts, the encoder analyses the input sequence: first is the search buffer, and then Look-ahead buffer. A portion of the newly encoded sequence holds by search buffer, while the next portion of the sequence to be encoded is holds by look-ahead buffer. With the start of the look-ahead buffer, the algorithm finds the sliding window for the longest match and provides a pointer to that match. There could be no match at all, so the output cannot only contain pointers. The series is encoded as a triple <o, l, c> in the LZ77 algorithm, where 'o' stands for an offset to the match, 'l' represents the match length, and 'c' represents the next symbol to be encoded. In the absence of a match, the null pointer is created as the pointer (offset and match length equal to 0) and the first symbol in the look-ahead buffer, i.e. (0, 0"character'). The offset values must be limited to certain maximum constants for a match and length. In addition, the compression efficiency of the LZ77 depends on these values.

F. LZ78

The LZ78 is LZW is dictionary based compression technique instead of a statistical model. The dictionary is a set of possible words of a language, and is stored in a table like structure and used the indexes of entries to represent larger and repeating dictionary words. The Lempel-Zev Welch algorithm or simply LZW algorithm is one of such algorithms in which a dictionary is used to store and index the previously seen string patterns. In the compression process, those index values are used instead of repeating the string patterns. The dictionary is generated dynamically in the process of compression and no need to transfer it with the encoded message for decompressing. In the decompression process, the same dictionary is generated dynamically. Therefore, this algorithm is known as an adaptive compression algorithm.

G. LZW

LZW is dictionary based compression technique instead of a statistical model. The dictionary is a set of possible words of a language, and is stored in a table like structure and used the indexes of entries to represent larger and repeating dictionary words. The Lempel-Zev Welch algorithm or simply LZW

algorithm is one of such algorithms in which a dictionary is used to store and index the previously seen string patterns. In the compression process, those index values are used instead of repeating the string patterns. The dictionary is generated dynamically in the process of compression and no need to transfer it with the encoded message for decompressing. In the decompression process, the same dictionary is generated dynamically. Therefore, this algorithm is known as an adaptive compression algorithm.

IV. LITERATURE REVIEW

In [1] reviews of different basic lossless data compression methods are given. Lossless compression algorithm is described in brief. This paper concentrates on different methods of data compression, such as dictionary-based and entropy-based. The Shannon Fano and Huffman encoding algorithms are the two approaches that are better than the RLE algorithm in the entropy-based technique. But both the Shannon Fano and Huffman compression algorithms are almost the same, but the Huffman coding algorithm is stronger than the Shanon-Fano algorithm, and the LZW approach provides the best and most reliable result in the dictionary-based compression algorithm.

In [2], Run Length Encoding is addressed as a successful compression technique in the case of symbols or data being repeated consecutively. This compression technique does not work properly if there is no repetition of the data. Huffman coding is a better method for compression than Shannon Fano coding. The most efficient coding technique is arithmetic coding, and instead of replacing a stream of input data with a floating number as output, it does not replace every bit with a codeword as other approaches. A message or information is represented by a half-open interval [a, b] in the arithmetic coding algorithm, where a and b are real numbers between 0 and 1. The arithmetic coding compression ratio is higher than coding with Huffman compression and Shannon Fano. It also reduces the bandwidth of the channel and transmission time.

In [5], the paper addresses and analyses the various forms of lossless compression methods by analyzing their parameters of measurement. The Lempel ziv method demonstrates better output than the RLE and Huffman encoding from the study. A strong compression ratio of about 4:1 compression with 76.9 percent space saving has been achieved by the Lemphel ziv compression process. The analysis shows that Lempel Ziv, in terms of compression ratio and space saving, is appropriate to accelerometer data. So it won't take extensive bandwidth to send the transmitter to the receiver with the humongous data.
In [6], three lossless data compression algorithms are discussed in the paper, such as Huffman coding, Arithmetic encoding, and Lempel Ziv Welch coding. It's quick to implement Huffman encoding. It has some disadvantages, such as that the Huffman algorithm is relatively slow and relies on a statistical data model. Because of the various code lengths, decoding is difficult. It's overhead, due to the Huffman tree. It uses a fraction to represent the whole source message in Arithmetic encoding. The ease of adaptation, in which adaptation is the adjustment of the frequency (or probability) tables when processing the data, has one advantage over other similar data compression techniques. Some of the drawbacks of Arithmetic encoding are that in order to start decoding the symbols, the entire codeword must be obtained, and if codeword contains some corrupt bit, then the whole message could become corrupt. The Lempel Ziv Welch algorithm is easy to implement and achieves a high degree of hardware implementation consistency. By referencing a dictionary instead of tabulating character counts and constructing trees, LZW encodes data (as for Huffman encoding). The paper concluded that the technique of arithmetic coding

works more effectively compared to Huffman coding in statistical compression techniques. Arithmetic encoding encodes the entire message or data into a single number, while Huffman encoding divides the input into component symbols and replaces each with a code. Arithmetic encoding usually has a st ronger compression ratio than Huffman encoding. Dictionary-based algorithms such as LZW work faster than those based on entropy. Input is processed in the dictionary-based algorithm as a sequence of characters rather than as bit streams. In the compression process, the dictionary is dynamically generated. There is no need to move the dictionary with the encoded message when decompressing. Algorithms that dynamically generate the same dictionary are adaptive compression algorithms.

V. MEASURING COMPRESSION PERFOMANCE

A. **Compression Ratio**: It is ratio of the compressed file size to the actual file size.

$$\text{Compression Ratio} = \frac{\text{Uncompressed Size}}{\text{Compressed Size}}$$

B. **Space Saving**: Space saving is known as the size reduction compared to the uncompressed size.

$$\text{Space Saving} = 1 - \frac{\text{Compressed Size}}{\text{Uncompressed Size}}$$

VI. PERFOMANCE ANALYSIS

TABLE I. Comparison between different entropy based data compression techniques

Parameters	Shanno-Fano Coding	Huffman Coding	Run Length Encoding	Arithmetic Coding
Original File Size (in KB)	2253	2253	2253	2253
Compressed File Size (in KB)	680	616	947	1126
Compression Ratio	30.18	27.34	42.03	50
Space Saving (%)	70	72.65	57.96	50

TABLE III. Comparison between different dictionaries based data compression techniques

Parameters	LZ77	LZ78	LZW
Original File Size (in KB)	2253	2253	2253
Compressed File Size (in KB)	669	1023	421
Compression Ratio	29.69	45.40	18.68
Space Saving (%)	70	54.59	81.31

VII. CONCLUSION

To decrease the file size, compression techniques are used. There is no loss of original bits after decompression in Lossless Data Compression. This paper introduces different kinds of techniques for compressing data. This paper gave an overview of the various algorithms for lossless data compression. In this paper, the main emphasis is on explaining different methods of data compression, such as dictionary-based and entropy-based. The Lempel-Ziv-Welch (LZW) method shows better efficiency (81.31 percent saving spaces) than the other lossless data compression techniques after the performance review of various different lossless data compression techniques.

VIII.REFERENCES

[1] A. B. Author Himali Patel, Unnati Itwala, Roshni Rana, Kruti Dangarwala "Survey of Lossless Data Compression Algorithms" International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Vol. 4 Issue 04, April-2015

[2] Apoorv Vikram Singh, Garima Singh "A Survey on Different text Data Compression Techniques" International Journal of Science and Research (IJSR) ISSN (Online) No. 2319-7064, Vol. 3 Issue 7, July 2014

[3] Prof. Dipti Mathpal, Prof. Mittal Darji, Prof. Sarangi Mehta "A Research Paper on Lossless Data Compression Techniques" International Journal for Innovative Research in Science & Technology (IJITST) ISSN (online) No. 2349-6010 Vol 4 Issue 1, June-2017

[4] Wenjun Huang, Weimin Wang and Hui Xu, "A Lossless Data Compression Algorithm for Real-time Database," *2006 6th World Congress on Intelligent Control and Automation,* Dalian, 2006, pp. 6645-6648, doi: 10.1109/WCICA.2006.1714368

[5] A. Gopinath and M. Ravisankar, "Comparison of Lossless Data Compression Techniques," 2020 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 2020, pp. 628-633, doi: 10.1109/ICICT48043.2020.9112516.

[6] Yamini Arora, Harshit Gupta, Parul Yadav, Shikhar Mittal "Literature Survey on Image and Text Compression Techniques" International Journal of Science Technology and Engineering (IJSTE) ISSN (online): 2349-784X Volume 3 Issue 09, March 2017

[7] Prof. Dipti Mathpal, Prof. Mittal Darji, Prof. Sarangi Mehta "A Research Paper on Lossless Data Compression Techniques" International Journal for Innovative Research in Science & Technology (IJIRST) ISSN (online): 2349-6010 Volume 4 Issue 1, June 2017

[8] Pooja Singh "Lossless Data Compression Techniques and Comparison Between The Algorithms" International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 p-ISSN: 2395-0072 Volume: 02 Issue: 02, May-2015

[9] Er. Mangi Lal, Er. Sammah Rasheed "A Review on Data Compression Techniques" IJARIIE-ISSN(O)-2395-4396 Vol-6 Issue-1 2020

[10] Suman M. Choudhary, Anjali S. Patel, Sonal J. Parmar "Study of LZ77 and LZ78 Data Compression Techniques" International Journal of Engineering Science and Innovative Technology (IJESIT) ISSN: 2319-5967, Volume 4, Issue 3, May 2015

[11] https://en.wikipedia.org/wiki/Data_compression_ratio

[12] Dalvir Kaur, Kamaljeet Kaur "Analysis of Lossless Data Compression Techniques" International Journal of Computational Engineering Research, Volume 03, Issue 4, April 2013

Cite this article as :

Anshul Gupta, Prof. Sumit Nigam, "A Review on Different Types of Lossless Data Compression Techniques", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 7 Issue 1, pp. 50-56, January-February 2021. Available at doi : https://doi.org/10.32628/CSEIT217113
Journal URL : http://ijsrcseit.com/CSEIT217113