Volume 5, Issue 7, July 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Approach for Lossless Text data Compression using Advanced Bit Reduction Algorithm

Amarjit Kaur
Student of M.Tech
AIET, Faridkot, Punjab, India

Navdeep Singh Sethi
Assistant Professor
AIET, Faridkot, Punjab, India

Abstract: ***Data Compression is a procedure for encoding chooses that allows impressive diminishment in the total number of bits to store or transmit an archive. Transmission of extensive amount of information cost more cash. Henceforth picking the best information pressure calculation is truly essential. Notwithstanding diverse pressure advancements and approachs, determination of a decent data compression algorithm is generally imperative. There is a complete scope of distinctive data compression methods accessible both online and disconnected from the net working such that it turns out to be truly hard to pick which strategy serves the best. In this paper we present an algorithm that is an advanced version of Bit Reduction Algorithm for data compression.***

Keywords: Text data compression, Advanced Bit Reduction method, Lossless data compression.

I. INTRODUCTION

Data compression is a process by which a file (Text, Audio, Video) may be transformed to another (compressed) file, such that the original file may be fully recovered from the original file without any loss of actual information. This process may be useful if one wants to save the storage space. For example if one wants to store a 4MB file, it may be preferable to first compress it to a smaller size to save the storage space.

Also compressed files are much more easily exchanged over the internet since they upload and download much faster. We require the ability to reconstitute the original file from the compressed version at any time. Data compression is a method of encoding rules that allows substantial reduction in the total number of bits to store or transmit a file. The more information being dealt with, the more it costs in terms of storage and transmission costs. In short, Data Compression is the process of encoding data to fewer bits than the original representation so that it takes less storage space and less transmission time while communicating over a network.

Data compression algorithms are classified in two ways i.e. lossy and lossless data compression algorithm. A compression algorithm is utilized to change over information from a simple to-utilize arrangement to one advanced for smallness. In like manner, an uncompressing system gives back the data to its unique structure.

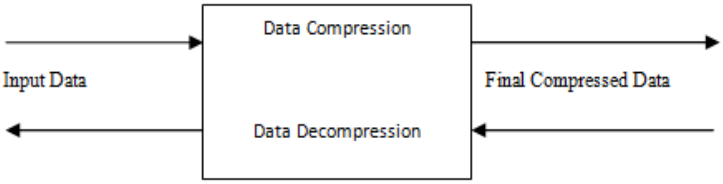


Fig. 1.2 Data Compression and Decompression

1.2 TYPES OF DATA COMPRESSION

Data Compression techniques can be classified in two ways:

- Lossy Data Compression
- Lossless Data Compression

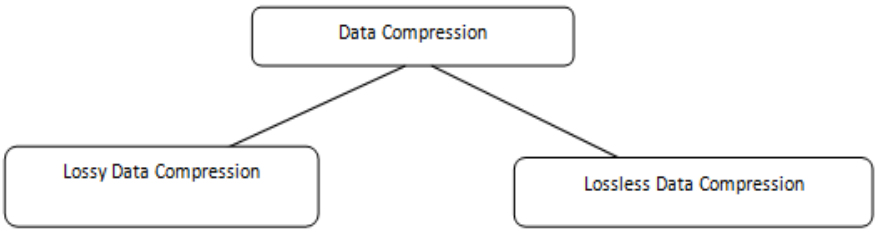


Fig: 1.3 Classification of Data Compression

1.2.1 Lossy data compression

A lossy data compression system is one where the data recovers after decompression may not be precisely same as the first data, but rather is "sufficiently close" to be valuable for particular reason. After one applies lossy data compression

to a message, the message can never be recuperated precisely as it was before it was packed. At the point when the compacted message is decoded it doesn't give back the first message. Data has been lost. Since lossy compression can't be decoded to yield the definite unique message, it is not a decent system for compression for basic data, for example, printed data. It is most valuable for Digitally Sampled Analog Data (DSAD). DSAD comprises for the most part of sound, feature, illustrations, or picture documents. In a sound document, for instance, the high and low frequencies, which the human ear can't listen, may be truncated from the record.

The cases of continuous utilization of Lossy data compression are on the Internet and particularly in the gushing media and telephony applications. A few samples of lossy data compression calculations are JPEG, MPEG, MP3.Most of the lossy data compression methods experience the ill effects of era misfortune which means diminishing the nature of content in light of over and again packing and decompressing the record. Lossy picture compression can be utilized as a part of computerized cameras to build stockpiling limits with negligible debasement of picture quality.

1.2.2 Lossless data compression

Lossless data compression is a procedure that permits the utilization of data compression calculations to pack the content data furthermore permits the precise unique data to be remade from the compacted data. This is in as opposed to the lossy data compression in which the careful unique data can't be recreated from the compacted data. The prevalent ZIP record organize that is being utilized for the compression of data documents is likewise a use of lossless data compression approach. Lossless compression is utilized when it is vital that the first data and the decompressed data be indistinguishable. Lossless content data compression calculations typically abuse factual excess in such a path in order to speak to the sender's data all the more briefly with no blunder or any kind of loss of vital data contained inside of the content information data. Since the majority of this present reality data has factual excess, thusly lossless data compression is conceivable. Case in point, In English content, the letter "an" is a great deal more basic than the letter 'z', and the likelihood that the letter "t" will be trailed by the letter "z" is little. So this sort of repetition can be evacuated utilizing lossless compression. Lossless compression techniques may be classified by kind of data they are intended to pack. Compression calculations are essentially utilized for the compression of content, pictures and sound. Most lossless compression projects utilize two various types of calculations: one which creates a factual model for the info data and another which maps the information data to bit strings utilizing this model as a part of such a route, to the point that as often as possible experienced data will deliver shorter yield than improbable(less continuous) data.

The upside of lossless techniques over lossy systems is that Lossless compression results are in a closer representation of the first info data. The execution of calculations can be thought about utilizing the parameters, for example, Compression Ratio and Saving Percentage. In a lossless data compression document the first message can be precisely decoded. Lossless data compression lives up to expectations by discovering rehashed examples in a message and encoding those examples in an effective way. Thus, lossless data compression is likewise alluded to as repetition decrease. Since repetition decrease is reliant on examples in the message, it doesn't function admirably on arbitrary messages. Lossless data compression is perfect for content.

II. LITERATURE REVIEW

This section involves the Literature survey of various techniques available for Data compression and analyzing their results and conclusions.

H. Altarawneh and M. Altarawneh, "Data Compression Techniques on Text Files: A Comparison Study" : This paper presents various methods of data compression such as LZW, Huffman, Fixed-length code (FLC) and Huffman after using Fixed-length code (HFLC) on text files. The authors have evaluated and test the algorithms on various sizes of text files and compared their performance on various parameters such as compression size, compression ratio, compression time and entropy.

U. Khurana and A.Koul, "Text Compression And Superfast Searching": A new compression technique that uses referencing through two-byte numbers (indices) for the purpose of encoding has been presented. The technique is efficient in providing high compression ratios and faster search through the text. It leaves a good scope for further research for actually incorporating phase 3 of the given algorithm. The same should need extensive study of general sentence formats and scope for maximum compression. Another area of research would be to modify the compression scheme so that searching is even faster. Incorporating indexing so as to achieve the same is yet another challenge.

A. Singh and Y. Bhatnagar, "Enhancement of data compression using Incremental Encoding" : This paper describes the two phase encoding technique which compresses the sorted data more efficiently. This research paper provides a way to enhance the compression technique by merging RLE compression algorithm and incremental compression algorithm. In first phase the data is compressed by applying RLE algorithm that compresses the frequent occur data bits by short bits. In the second phase incremental compression algorithm stores the prefix of previous symbol from the current symbol and replaces with integer value. This technique can reduce the size of sorted data by 50% using two phase encoding technique.

A.J Mann, "Analysis and Comparison of Algorithms for Lossless Data Compression" : In this paper the author discusses and compares selected set of lossless data compression algorithms such as RLE, Huffman and Arithmetic coding. The author compares the performance of these algorithms on the basis of various parameters such as Compression Ratio, Compression speed, Decompression speed, Memory space etc. The author has concluded that the compression speed of Huffman is better than the Arithmetic coding, but the compression ratio of Arithmetic coding is better as compared to the Huffman coding. The author has also concluded that Arithmetic coding is the efficient compression algorithm among the selected ones.

III. RESEARCH AND DESIGN METHODOLOGY

Modified Huffman Data Compression algorithm works in three phases to compress the text data. In the first phase data is compressed with the help of dynamic bit reduction technique and in second phase unique words are to be found to compress the data further and in third and final phase Huffman coding is used to compress the data further to produce the final output. Following are the main steps of algorithm for compression and decompression :

COMPRESSION ALGORITHM

Step I : Input the text data to be compressed.
Step II : Find the number of unique words in the input text data and assign the symbols that are not in the input.
Step III : Assign the numeric code to the unique symbols found in the step II.
Step IV : Starting from first symbol in the input find the binary code corresponding to that symbols from assigned numerical codes and concatenate them to obtain binary output.
Step V : Add number of 0's in MSB of Binary output until it is divisible by 8.
Step VI : Generate the ASCII code for every 8 bits for the binary output obtained in step V and concatenate them to create input for second phase.
[Step VI is the result of dynamic bit Reduction Method in ASCII format]
Step VII: Display the final result obtained in step VII.
[Output from step VII is final compressed output]

DECOMPRESSION ALGORITHM

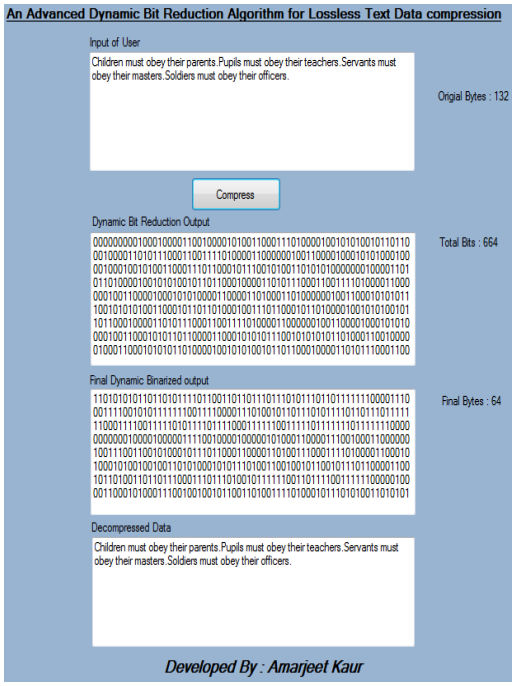
Step I : Input the Final output from compressed phase.
Step II : Calculate the binary code corresponding to the ASCII values of input obtained in Step I.
Step III : Remove the extra bits from the binary output added in the compression phase.
Step IV : Calculate the numeric code for every 8 bits obtained in the Step IV.
Step V : For every numeric value obtained in the step V, find the corresponding symbol to get the final decompressed data.
Step VI : Concatenate the data symbols obtained in the step VI and obtain the final output.
Step VII: Display the final result to the user.

IV. RESULTS AND DISCUSSION

We have tested the proposed system on various types of inputs containing verities of text inputs. Following are the

Performance Parameters: Performance evaluation of the proposed algorithm is done using two parameters- Compression Ratio and Saving Percentage.

- **Compression ratio:** Compression ratio is defined as the ratio of size of the compressed file to the size of the source file.
  Compression ratio=C2/C1 *100%
- **Saving Percentage:** Saving Percentage calculates the shrinkage of the source file as a percentage.
  Saving percentage = (C1-C2/C1)*100%
  C1= Size before compression
  C2= Size after compression



Input text size (in bytes)	Output of proposed System (in bytes)	Compression Ratio of proposed system (In %)	Saving Percentage
132	64	48.4	83.5
117	36	30.7	69.2
176	87	49.4	50.5
352	174	49.4	50.5

Table Above shows the various experiments conducted by the authors to determine the compression ratio and space saving percentage achieved by the final proposed system for random data set.

Comparison Table and Graph on Compression Ratio for Random Dataset: The following tables and graphs represent the comparison of Compression ratios of the existing techniques and the proposed system.

TABLE 5.2 COMPRESSION RATIO COMPARISON FOR RANDOM DATASET

Input text size (in bytes)	Bit Reduction Compression ratio (In %)	Huffman Compression ratio (In %)	Proposed System Compression ratio (in %)
132	62.8	48.4	47.4
117	75.2	42.7	30.7
176	75	57.9	49.4
352	75	57.9	49.4

From
Table 5.2, it is clear that the compression ratio achieved by the proposed system is lesser as compared to the existing techniques which means it results in more savings of the storage space.

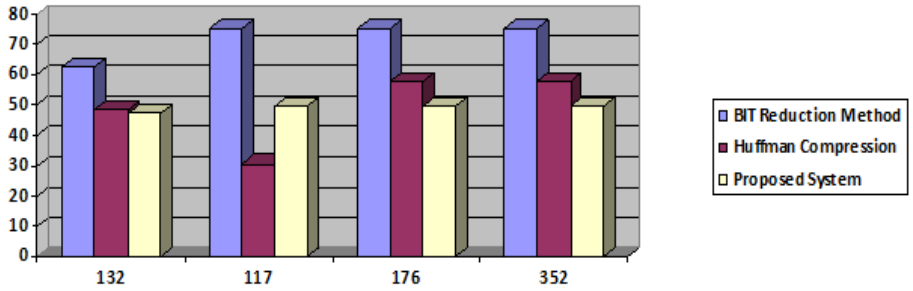


Fig. 5.3 Compression ratio comparison graph for random dataset

The graph above shows the comparison among three systems.

V. CONCLUSION AND FUTURE WORK

In this paper an Advanced Bit Reduction Algorithm for lossless text data compression is presented. Test data for the proposed system consist of various inputs from different type of sources such as books, newspapers etc. Algorithm so developed is tested on these inputs of different size of text. Accuracy shown by the system is considerably high and accurate. In future a system is required to be made that can work on text, audio and video files simultaneously for data compression.

REFERENCES

[1] R.S. Brar and B.Singh, "A survey on different compression techniques and bit reduction algorithm for compression of text data" International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) Volume 3, Issue 3, March 2013

[2] S. Porwal, Y. Chaudhary, J. Joshi and M. Jain , " Data Compression Methodologies forLossless Data and Comparison between Algorithms" International Journal of Engineering Science and Innovative Technology (IJESIT) Volume 2, Issue 2, March 2013

[3] S. Shanmugasundaram and R. Lourdusamy, "A Comparative Study of Text CompressionAlgorithms" International Journal of Wisdom Based Computing, Vol. 1 (3), December 2011

[4] S. Kapoor and A. Chopra, "A Review of Lempel Ziv Compression Techniques" IJCST Vol. 4, Issue 2, April - June 2013

[5] I. M.A.D. Suarjaya, "A New Algorithm for Data Compression Optimization", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No. 8, 2012, pp.14-17

[6] S.R. Kodituwakku and U. S. Amarasinghe , "Comparison Of Lossless Data Compression Algorithms For Text Data" Indian Journal of Computer Science and Engineering Vol1No 4 416-425

[7] R. Kaur and M. Goyal, "An Algorithm for Lossless Text Data Compression" International Journal of Engineering Research & Technology (IJERT), Vol. 2 Issue 7, July - 2013

[8] H. Altarawneh and M. Altarawneh, " Data Compression Techniques on Text Files: A Comparison Study " International Journal of Computer Applications, Volume 26– No.5, July 2011

[9] U. Khurana and A. Koul, "Text Compression And Superfast Searching" Thapar Institute Of Engineering and Technology, Patiala, Punjab, India-147004

[10] A. Singh and Y. Bhatnagar, " Enhancement of data compression using Incremental Encoding" International Journal of Scientific & Engineering Research, Volume 3, Issue 5, May-2012

[11] A.J Mann, " Analysis and Comparison of Algorithms for Lossless Data Compression" International Journal of Information and Computation Technology, ISSN 0974-2239 Volume 3, Number 3 (2013), pp. 139-146

[12] K. Rastogi, K. Sengar, "Analysis and Performance Comparison of Lossless Compression Techniques for Text Data" International Journal of Engineering Technology and Comp- uter Research (IJETCR) 2 (1) 2014, 16-19

[13] M. Sharma, "Compression using Huffman Coding " IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.5, May 2010

[14] S. Shanmugasundaram and R. Lourdusamy, " IIDBE: A Lossless Text Transform for Better Compression " International Journal of Wisdom Based Computing, Vol. 1 (2), August 2011

[15] P. Kumar and A.K Varshney, " Double Huffman Coding " International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) Volume 2, Issue 8, August 2012

[16] R. Gupta, A. Gupta, S. Agarwal, "A Novel Data Compression Algorithm For Dynamic Data" IEEE REGION 8 SIBIRCON

[17] A. Kattan, "Universal Intelligent Data Compression Systems: A Review" 2010 IEEE

[18] M. H Btoush, J. Siddiqi and B. Akhgar, "Observations on Compressing Text Files of Varying Length " Fifth International Conference on Information Technology: New Generations, 2008 IEEE

[19] A. Jain, R. Patel, " An Efficient Compression Algorithm (ECA) for Text Data " International Conference on Signal Processing Systems, 2009 IEEE

[20] Md. R. Hasan, "Data Compression using Huffman based LZW Encoding Technique" International Journal of Scientific & Engineering Research, Volume 2, Issue 11, November-2011, pp.1-7

[21] M. Gupta, B. Kumar, "Web Page Compression using Huffman Coding Technique" International Conference on Recent Advances and Future Trends in Information Technology (iRAFIT2012) Proceedings published in IJCA, International Journal of Computing Applications

[22] P. Yellamma, N. Challa, " Performance Analysis Of Different Data Compression Techniques On Text File" International Journal of Engineering Research & Techno- logy (IJERT), Vol. 1 Issue 8, October – 2012, pp.1-6

