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

Lossless Text Data Compression Algorithm Using Modified Huffman Algorithm

Manjeet Kaur
M.Tech Student
CSE Department, Gurukashi University
Talwandi Sabo, Punjab, India

Er. Upasna Garg
Assistant Professor
CSE Department, Gurukashi University
Talwandi Sabo, Punjab, India

Abstract: ***Data Compression is a strategy for encoding decides that permits considerable diminishment in the aggregate number of bits to store or transmit a document. Transmission of large quantity of data cost more money. Hence choosing the best data compression algorithm is really important. In addition to different compression technologies and methodologies, selection of a good data compression tool is most important. There is a complete range of different data compression techniques available both online and offline working such that it becomes really difficult to choose which technique serves the best. In this paper we represent an algorithm based on modified Huffman coding to compress and decompress the text data.***

Keywords: ***Text data compression, Decompression of data, Dynamic Bit Reduction method, Huffman coding, lossless data compression.***

I. INTRODUCTION

An information pressure calculation makes an interpretation of a data article to a compacted grouping of yield images, from which the first data can be recouped with a coordinating decompression calculation. Compressors (and their coordinating decompressors) are planned with the objective that the compacted yield is, by and large, less expensive to store or transmit than the first information. For example if one wants to store a large data 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 capacity to reconstitute the first record from the compacted rendition whenever. Information pressure is a system for encoding decides that permits considerable decrease in the aggregate number of bits to store or transmit a document. The more data being managed, the more it expenses regarding stockpiling and transmission costs. To put it plainly, Data Compression is the procedure of encoding information to less bits than the first representation so it consumes less storage space and less transmission time while conveying more than a system.

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.

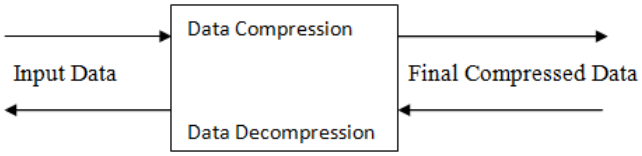

Fig. 1.2 Data Compression and Decompression

1.2 TYPES OF DATA COMPRESSION

As of now, two essential classes of Data Compression are connected in diverse areas. One of these is lossy Data Compression, which is generally used to pack picture information documents for correspondence or files purposes. The other is lossless data compression that is regularly used to transmit or file content or parallel records needed to keep their data in place whenever. Data Compression algorithm can be classified in two ways:

- Lossy Data Compression
- Lossless Data Compression

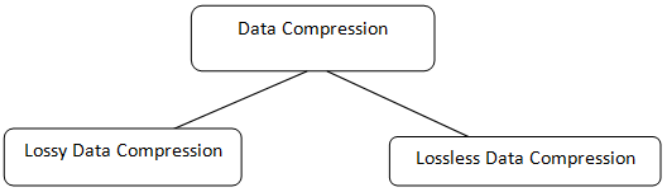

Fig: 1.3 Classification of Data Compression

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.

R.S. Brar and B.Singh, "A survey on different compression techniques and bit reduction algorithm for compression of text data" : This paper provides a survey of different basic lossless and lossy data compression techniques. On the basis of these techniques a bit reduction algorithm for compression of text data has been proposed by the authors based on number theory system and file differential technique which is a simple compression and decompression technique free from time complexity. Future work can be done on coding of special characters which are not specified on key-board to revise better results [1].

S. Porwal, Y. Chaudhary, J. Joshi, M. Jain, "Data Compression Methodologies for Lossless Data and Comparison between Algorithms" : This research paper provides lossless data compression methodologies and compares their performance. Huffman and arithmetic coding are compared according to their performances. In this paper the author has found that arithmetic encoding methodology is powerful as compared to Huffman encoding methodology. By comparing the two techniques the author has concluded that the compression ratio of arithmetic encoding is better and furthermore arithmetic encoding reduces channel bandwidth and transmission time also [2].

S. Shanmugasundaram and R. Lourdusamy, "A Comparative Study of Text Compression Algorithms" : There are lot of data compression algorithms which are available to compress files of different formats. This paper provides a survey of different basic lossless data compression algorithms. Experimental results and comparisons of the lossless compression algorithms using Statistical compression techniques and Dictionary based compression techniques were performed on text data. Among the statistical coding techniques the algorithms such as Shannon-Fano Coding, Huffman coding, Adaptive Huffman coding, Run Length Encoding and Arithmetic coding are considered. A set of interesting conclusions are derived on their basis. Lossy algorithms achieve better compression effectiveness than lossless algorithms, but lossy compression is limited to audio, images, and video, where some loss is acceptable. The question of the better technique of the two, "lossless" or "lossy" is pointless as each has its own uses with lossless techniques better in some cases and lossy technique better in others [3].

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 : Apply Dynamic bit Reduction method to compress the data.
Step III : Find the unique symbol to compress the data further.
Step IV : Create the binary tree with nodes representing the unique symbols
Step V : Apply Huffman coding to Finally compress the data.
Step VI : Display the final result obtained in step V.

DECOMPRESSION ALGORITHM

Step I : Enter the compressed data as input.
Step II : Apply Inverse Huffman code to decompress the data.
Step III : Construct the binary tree representing nodes as data.
Step IV :Replace the unique symbols with substitutes.
Step V : Apply reverse Dynamic Bit Reduction method to finally decompress the data.
Step VI : 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
174	52	59.60	70.1
234	72	61.14	69.2
176	87	24.7	50.5
179	96	17.76	46.6

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 %)
174	75.8	42.5	29.8
234	75.2	42.7	30.7
176	75	57.9	49.4
179	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.

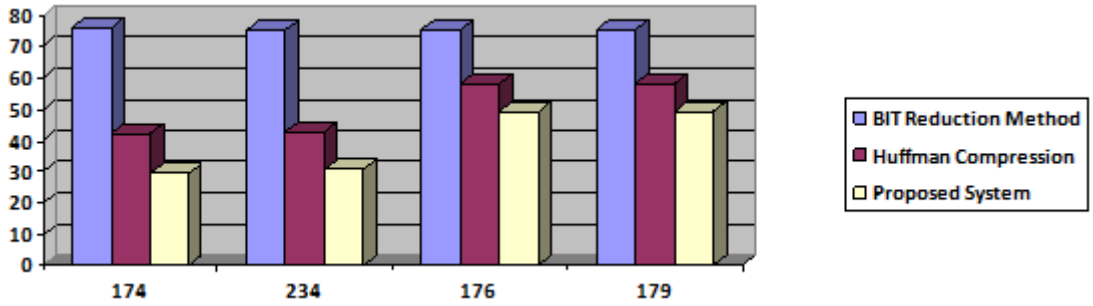

Fig. 5.3 Compression ratio comparison graph for random dataset

The graph above shows the comparison among three systems.

V. CONCLUSION AND FUTURE WORK

In this paper Modified Huffman algorithm for text data compression is presented. Proposed system is tested on various inputs of different size of text. System shows very improved results than the existing systems. Proposed system works only with text data written in single language which can also be tested to compress the multi lingual data i.e. text data written in multiple languages in a single file. In other words, the system works only on ASCII dataset which can be extended to work with Unicode data for future work.

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 for Lossless 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 Compression Algorithms" 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 Computer 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 2012, pp.1-6

