International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 14
October2016

Novel Cyclic Redundancy Check software for compressed & decoded Data

Pragya Rathore[1]&Vandana Kapoor[2]

[1]M-TECH, Dept. CSE Delhi Institute of Technology, Management & Research, Faridabad, Haryana, INDIA,Mail Id:- er.pragyarathore@gmail.com

[2]ASSISTANT PROFESSOR, Dept. CSE Delhi Institute of Technology, Management & Research, Faridabad, Haryana, INDIA,Mail Id:-kapoor.vandana337@gmail.com

Abstract

The aim of this Paper is to describe a software using which, size of any file can be reduced. Files will be compressed using the CRC (Cyclic Redundancy Check) algorithm. The efficiency will be determined by comparing the compression ratio of the original file to the compressed file. The compressed file will be decoded accordingly.

Keywords:CRC (Cyclic Redundancy Check),software,compressed file,decoded.

1. Introduction

As the world evolves to rely more and more on computers and data storage, data compressionbecomes an increasingly useful and important tool. Most data used by humans contains a high amount ofredundancy. Compression algorithms aim to eliminate this redundancy, while still preserving all theinformation contained in the original data. The end result is a reduction in storage requirements, inexchange for increased computation.

This software help in reducing the size of a file by reducing number of bits originally needed to storethat file

1.1 Product perspective

- The reduction in the size of the file will result in better memory management.
- While transferring a file via network it would consume lesser network resources.
- The time required to transmit the file in a network would be less as compared to theuncompressed file.

1.2 Product Functions

The software developed will be used for reducing number of bits needed for storing a file.

1.3 User Characteristics

Any computer user who wants to reduce the number of bits needed to represent his/ her data.

Also any use who deals with various file formats can use this software to compress all the files.

1.4 General Constraints

This software must be installed at the users workstations.

Assumption

- The user must have some basic knowledge to deal with the software.
- The user must be able to distinguish between compressed file and uncompressed file.

Dependencies

- **Algorithm structure.**
- **Compression ratio.**

2. Related Work

The software must take a file with any extension as input and the output must be thereduced form of the input file. Here reduction means that number of bits needed to represent theinput file is reduced.

2.1 Functions

This software will take as input a file and will reduce the number of bits needed to storethat file. CRC (Cyclic Redundancy Check) algorithm for compression will be implemented.

2.2 Performance Requirements

Every programmatic implementation of compression encoding has a differentcompression ration and performance. Choosing the right encoder from a list of publiclyavailable encoders is not a simple task because performance and compression ration alsodepend on the type of data, particularly on the size of the alphabet (number of differentsymbols) and words (in case of LZ encoding). One of two particular encoders may have betterperformance for small alphabets while the other may show better performance forlarge files. Most encoders have limitations on the size of the alphabet.

So an algorithm must be selected such that it could compress most of the fileformats giving high compression ratio and efficiency. The software must give thecompressed file instantly not taking much time. The time complexity and spacecomplexity must be optimized.

2.3 Compression- Compression is the process of encoding information using fewer bits (orother information-bearing units) than an un-encoded representation would use, through use ofspecific encoding schemes.Compression is useful because it helps reduce the consumption of expensive resources,such as hard disk space or transmission bandwidth. On the downside, compressed data must bedecompressed to be used, and this extra processing may be detrimental to some applications. Forinstance, a compression scheme for video me require expensive hardware for the video to bedecompressed fast enough to be viewed as it

is being decompressed (the option ofdecompressing the video in full before watching it may be inconvenient, and requires storagespace for the decompressed video). The design of data compression schemes therefore involvestrade-offs among various factors, including the degree of compression, the amount of distortionintroduced (if using a lossy compression scheme), and the computational resources required tocompress and uncompress data.

2.4 Symmetry and asymmetry, in the context of data compression, refer to the time relationbetween compression and decompression for a given compression algorithm.If an algorithm takes the same time and space to compress a data archive as it does todecompress it, it is considered symmetrical.

The point to be noted here is that, compression and decompression, even for a symmetricalgorithm, may not be perfectly symmetric in practice, depending on the devices the data it beingcopied to and from, and other factors.

While if the compression and decompression times of an algorithm are vastly different, itis considered asymmetrical.Symmetric algorithms are typically used for media streaming protocols, as either theserver taking too long to compress the data, or the client taking too long to decompress, wouldlead to delays in the viewing of the data.

Asymmetrical algorithms wherein the compression is faster than the decompression canbe useful for backing up or archiving data, as in these cases data is typically much more oftenstored than retrieved.

2.5 Lossless data compression is a class of data compression algorithms that allows theexact original data to be reconstructed from the compressed data. The term lossless is in contrastto lossy data compression, which only allows an approximation of the original data to bereconstructed, in exchange for better compression rates.

Lossless data compression is used in many applications. For example, it is used in theZIP file format and in the UNIX tool GZIP. It is also often used as a component within lossy datacompression technologies.

Lossless compression is used in cases where it is important that the original and thedecompressed data be identical, or where deviations from the original data could bedecompressed data is identical. Or where deviations from the original data could be deleterious.Typical examples are executable programs, text documents and source code. Some image fileformats, like PNG or GIF, use only lossless compression, while others like TIFF and MNG mayuse either lossless or lossy methods. Lossless audio formats are most often

used for archiving orproduction purpose, with smaller lossy audio files being typically used on portable players and inother cases where storage space is limited and/ or exact replication of the audio is unnecessary.

3. Implementation

3.1 Proposed System:

This software is used for compression. The software is designed in such a way that it cancompress any type of file like text, image, audio and video. So this software is a general compressor thatcan compress any kind of file.

The basic functionality of this software is to compress a file which is supplied by the user. Alsotis job is to decompress the compressed file

The working of the software can be better understood by the following steps:-

Compression process

- Take input file from the user

- Check whether the file has .zip extension or not

- If the file has an extension .zip ten give an error message to the user that the file is already compressed and end the process.

-If file does not have .zip as its extension then

- Ask the location where the user wants to save the compressed file
- Browse to the location
- Compress the file using the algorithm

Decompression process

-Take input file from the user

- Check whether the file haz .zip extension or not.

- If the file does not have the extension .zip then give an error message to the user that the file isalready decompressed and end the process

- If file has .zip extension then

- Ask the location where the user wants to save the decompressed file
- Browse to the location
- Decompress the file using the algorithm

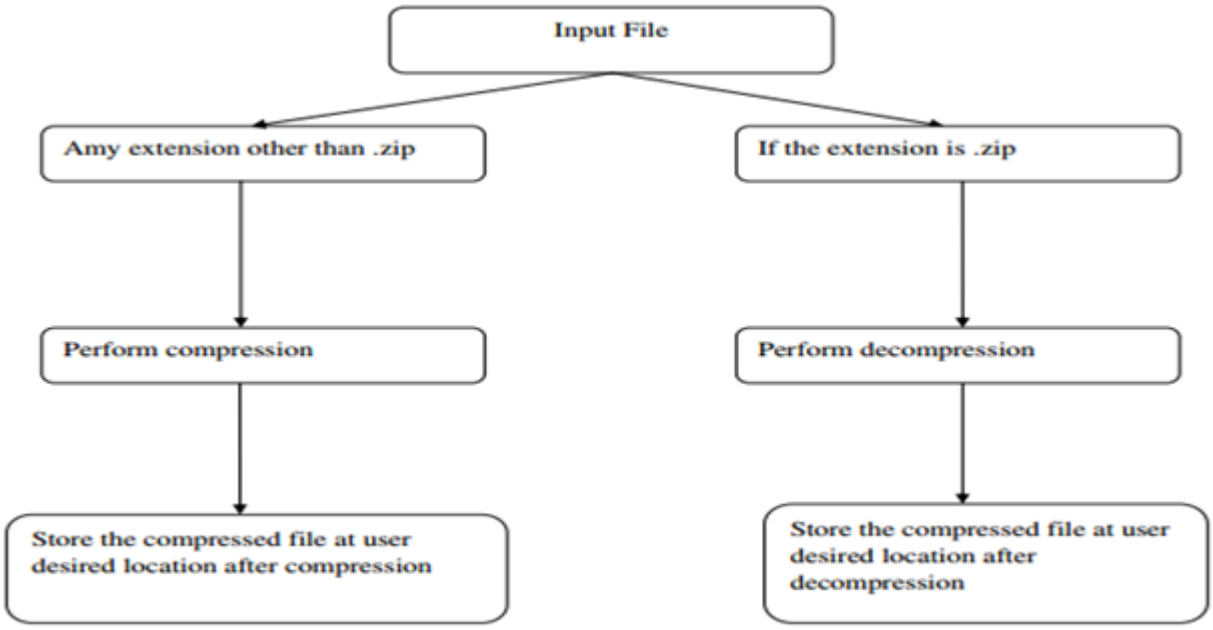

Fig 1: Functional block diagram.

Brief description of the various blocks used in the project

Input file – The first step of the process of the software is to ask the user to give a file as the

1.input. This input can be single file, multiple files or a folder

2. Compression- This process compresses the given file by reducing the number of bitsrequired to store the file

3. Decompression- This process decompresses the compressed file such that original file isobtained

4. .zip- It is the extension given for all the compressed files. Whenever a file is compressed itwould be stored with an extension .zip

5. Other Extension- Other extension refer to various file extension that are used for differentfile formats like .txt and .doc for text files, jpg and .gif for images, .mp3 and .wav for audiofiles, .mp4 and .3gp for video files. In other words, the file with any extension other than .zipwill be used for compression while files with extension .zip would be used fordecompression.

4. Experimental Work

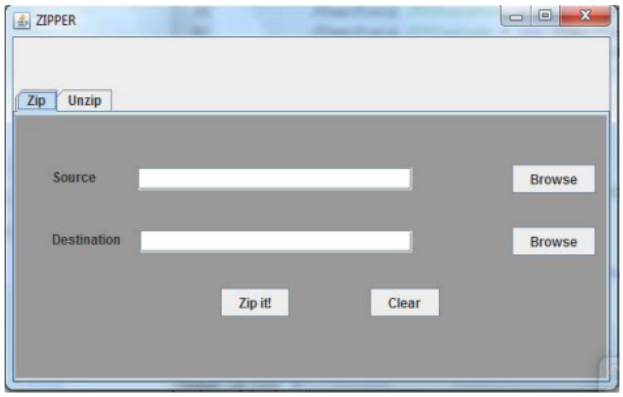

Fig 2: Before zip file browse file from system.

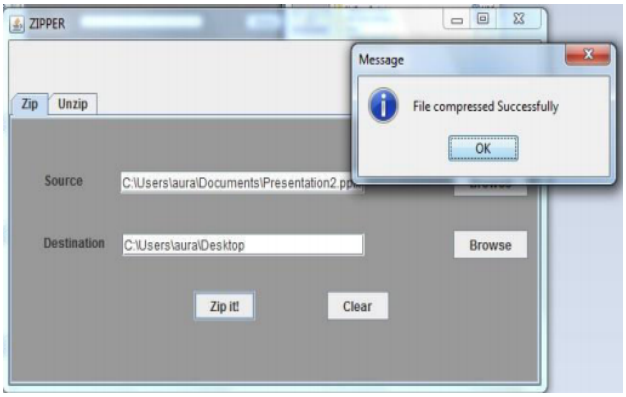

Fig 3: File After zipping Success Page.

5. Conclusion

As the world evolves to rely more and more on computers and data storage, data compression becomesan increasingly useful and important tool. Most data used by humans contains a high amount ofredundancy. Compression algorithms aim to eliminate this redundancy, while still preserving all theinformation contained in the original data. The end result is a reduction in storage requirements, inexchange for increased computation.

Nonetheless, but algorithms form important bases for more sophisticated work. In particular, thealgorithm as presented are known as 0-order…they don't base any of their symbol can be a very goodpredictor of the next symbol. For instance, in C language source code a right curly brace "}" is almostalways followed by carriage return.

6.References

[1] Wade, Graham (1994). Signal coding and processing (2 ed.). Cambridge University Press.p. 34. ISBN 978-0-521-42336-6. Retrieved 2011-12-22. The broad objective of source coding is to exploitor remove 'inefficient' redundancy in the PCM source and thereby achieve a reduction in the overall sourcerate R.

[2] Mahdi, O.A.; Mohammed, M.A.; Mohamed, A.J. (November 2012). "Implementing a Novel Approach aConvert Audio Compression to Text Coding via Hybrid Technique" (PDF). International Journal of ComputerScience Issues 9 (6, No. 3): 53–59. Retrieved 6 March 2013.

[3] Pujar, J.H.; Kadlaskar, L.M. (May 2010). "A New Lossless Method of Image Compression andecompression Using Huffman Coding Techniques" (PDF). Journal of Theoretical and Applied InformationTechnology 15 (1): 18–23.

[4] Salomon, David (2008). A Concise Introduction to Data Compression.

Berlin:Springer. ISBN 9781848000728.e. S. Mittal and J. Vetter, "A Survey of Architectural Approaches for Data Compression in Cache and MainMemory Systems", IEEE Transactions on Parallel and Distributed Systems, 2015.

[5] Tank, M.K. (2011). Implementation of Limpel-Ziv algorithm for lossless compression using VHDL.Think quest 2010: Proceedings of the First International Conference on Contours of ComputingTechnology. Berlin: Springer. pp. 275–283.

[6] Navqi, Saud; Naqvi, R.; Riaz, R.A.; Siddiqui, F. (April 2011). "Optimized RTL design and implementationof LZW algorithm for high bandwidth applications" (PDF).Electrical Review 2011 (4): 279–285.

[7] Mahmud, Salauddin (March 2012). "An Improved Data Compression Method for GeneralData" (PDF).International Journal of Scientific & Engineering Research 3(3): 2. Retrieved 6 March 2013.i. Arcangel, Cory. "On Compression" (PDF). Retrieved6 March 2013.

[8] Marak, Laszlo. "On image compression" (PDF). University of Marne la Vallee. Retrieved 6 March 2013.

[9] Mahoney, Matt. "Rationale for a Large Text Compression Benchmark". Florida Institute of Technology.Retrieved5 March 2013.

[10] Shmilovici A.; Kahiri Y.; Ben-Gal I.; Hauser S. "Measuring the Efficiency of the Intraday Forex Market witha Universal Data Compression Algorithm" (PDF). Computational Economics, Vol. 33 (2), 131-154., 2009.

[11] Ben-Gal. "On the Use of Data Compression Measures to Analyze Robust Designs" (PDF). IEEE Trans. OnReliability, Vol. 54, no. 3, 381-388, 2008.

[12] Korn, D.; et al. "RFC 3284: The VCDIFF Generic Differencing and Compression Data Format". InternetEngineering Task Force. Retrieved 5 March 2013.